Е.К. Беликова, А.Ю. Илютина, М.А.Круглова, С.И. Плиева,
Л.Б. Саратовская

COMPUTER SCIENCE:
THE ENGLISH LANGUAGE PERSPECTIVE

Москва
ИНФРА-М
2014

УДК
ББК

Е.К. Беликова, А.Ю. Илютина, М.А.Круглова, С.И. Плиева, Л.Б.Саратовская

Computer Science: the English Language Perspective. — М.: ИНФРА-М, 2014. — 299 с.

ISBN 978-5-16-

Текст аннотации

ББК

ISBN
2014

Подписано в печать. Формат. Гарнитура. Бумага офсетная. Усл. печ. л.. Уч.изд. л..
Тираж экз. Заказ №
Цена свободная.

Издательский Дом «ИНФРА-М»
127282, Москва, ул. Полярная, д. 31в.
Тел.: (495) 3800540, 3800543. Факс: (495) 3639212
E-mail: books@infra-m.ru http://www.infra-m.ru

Отпечатано по технологии «печать по требованию»
Тел.: (495) 363-92-15; e-mail: info@rior.ru
www.rior.ru
$

Предисловие

Данное учебно-методическое пособие предназначено для студентов первого и второго курсов факультета ВМК МГУ им. М.В. Ломоносова и является продолжением первых выпусков “English Reader for Computer Science” (2008 – 2009гг.).

Пособие составлено в соответствии с требованиями государственного стандарта профессионального образования в языковой подготовке для неязыковых специальностей высших учебных заведений и опирается на опыт учебно-методической работы кафедры английского языка факультета ВМК. Целью пособия является формирование и совершенствование навыков овладения английским языком как средством письменного и устного общения в сфере профессиональной деятельности. Пособие состоит из 11 разделов, включающих тексты и задания к ним. Структура всех разделов учебного пособия однотипна: текст со списком активной профессиональной лексики и комментариями «Notes» и с последующими заданиями.

Тексты пособия представляют собой оригинальные аутентичные материалы, посвященные основным проблемам современной компьютерной науки и информационным технологиям. При выборе текстов ставилась задача отобрать не только интересные и свежие материалы, но и, прежде всего, соответствующие лекциям, читаемым студентам на первых двух курсах факультетах ВМК.

Задания к текстам направлены на закрепление лексики и терминологии и умения использовать их не только при чтении, но и при письменном переводе тематических текстов как с английского на русский язык, так и с русского на английский язык. Каждый раздел заканчивается темами устных бесед и дискуссий по пройденному материалу.

Авторы пособия надеются, что вошедшие в него свежие познавательные тексты, в основном написанные, понятным и красивым английским языком, помогут вызвать и поддержать интерес к нему не только у студентов данной специальности, но также у широкого круга научных работников и аспирантов, работающих в области вычислительной техники.

Материалы пособия рассчитаны на средний и продвинутый уровни владения английским языком, поэтому авторы оставляют за преподавателями право свободного выбора организации учебного процесса по изучению этих материалов в зависимости от разного уровня знаний учащихся.

Л.Б. Саратовская

HISTORY OF COMPUTING

PRE-COMPUTER ERA

Read the following words and word combinations and use them for understanding and translation of the text:

to emerge – возникать, появляться
prehistory - совокупность сведений о доисторической эпохе
notch - насечка
aid – помощь
whole-number - целочисленный
gears and levers - шестеренки и рычаги
reliable – надежный
loom – ткацкий станок
to weave - ткать
to specify – точно определять, задавать
therefore – поэтому, следовательно
to implement – реализовывать
vision - замысел
to enable – давать возможность, позволять
a wide variety (of) – широкий выбор
to credit sb. with smth. – вменять что-л. в заслугу кому-л.
concept – понятие, идея
loop – цикл
to attribute – приписывать (авторство)
to propose – предлагать
proposition – предложение, утверждение
assertion – утверждение
value – (числовое) значение
to harness – использовать
census – перепись населения
vigorous – сильный, интенсивный
general-purpose - универсальный
principal means – основные способы

The idea of mechanical computation emerged in prehistory when early humans discovered that they could use physical objects such as piles of stones, notches, or marks as a counting aid. The ability to perform computation beyond simple counting extends back to the ancient world: for example, the abacus developed in ancient China could still beat the best mechanical calculators as late as the 1940s.

The mechanical calculator began in the West in the 17th century, most notably with the devices created by philosopher-scientist Blaise Pascal. He built and sold gear-driven mechanical machines, which performed whole-number addition and subtraction. Later in the seventeenth century, a German mathematician, Gottfried Wilhelm von Leibniz, built the first mechanical device designed to do all four whole-number operations: addition, subtraction, multiplication, and division. Unfortunately, the state of mechanical gears and levers at that time was such that the Leibniz machine was not very reliable.

In the late 18th century, Joseph Jacquard developed what became known as Jacquard's Loom, used for weaving cloth. The loom used a series of cards with holes punched in them to specify the use of specific colored thread and therefore dictate the design that was woven into the cloth. Though not a computing device, Jacquard's Loom was the first to make use of an important form of input: the punched card.

It wasn't until the 19th century that the next major step was taken, this time by a British mathematician. **Charles Babbage** designed what he called his *analytical engine*. His design was too complex for him to build with the technology of his day, so it was never implemented. His vision, however, included many of the important components of today's computers. It would have incorporated punched cards for data input, a central calculating mechanism (the "mill"), a memory ("store"), and an output device (printer). The ability to input both program instructions and data would enable such a device to solve a wide variety of problems

Then **Ada Augusta**, Countess of Lovelace, the daughter of Lord Byron, a most romantic figure in the history of computing and a skilled mathematician became interested in Babbage's work on the analytical engine and extended his ideas (as well as correcting some of his errors). Ada is credited with being the first programmer. The concept of the loop - a series of instructions that repeat - is attributed to her. The programming language Ada, used largely by the United States Department of Defense, is named for her.

In 1847, British mathematician **George Boole** proposed a system of algebra that could be used to manipulate propositions, that is, assertions that could be either true or false. In his system, called propositional calculus or Boolean Algebra, propositions can be combined using the "and" and "or" operators (called Boolean operators), resulting in a new proposition that is also either true or false.

Note the correspondence between the two values of Boolean logic and the binary number system in which each digit can have only the values of 1 or 0. Electronic digital computers are possible because circuits can be designed to follow the rules of Boolean logic, and logical operations can be harnessed to perform arithmetic calculations.

Besides being essential to computer design, Boolean operations are also used to manipulate individual bits in memory, storing and extracting information needed for device control and other purposes.

Herman Hollerith invented the automatic tabulating machine, a device that could read the data on punched cards and display running totals. His invention would become the basis for the data tabulating and processing industry.

Aided by Hollerith's machines, a census unit was able to process 7,000 records a day for the 1890 census, about ten times the rate in the 1880 count.

Facing vigorous competition and in declining health, Hollerith sold his patent rights to the company that eventually evolved into IBM, the company that would come to dominate the market for tabulators, calculators, and other office machines. The punched card, often called the Hollerith card, would become a natural choice for computer designers and would remain the principal means of data and program input for mainframe computers until the 1970s.

Notes:

Jacquard's loom (ткацкий станок, машина Жаккарда) was developed in1804-05 by Joseph-Marie Jacquard of France, but it soon spread elsewhere. His system improved on the punched-card technology of Jacques de Vaucanson's loom (1745). Jacquard's loom utilized interchangeable punched cards that controlled the weaving of the cloth so that any desired pattern could be obtained.

Propositional calculus - логическое исчисление

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The ability to perform computation beyond simple counting extends back to the ancient world…
2. …the abacus developed in ancient China could still beat the best mechanical calculators as late as the 1940s.
3. The mechanical calculator began in the West in the 17th century, most notably with the devices created by philosopher-scientist Blaise Pascal.
4. It wasn't until the 19th century that the next major step was taken, this time by a British mathematician.
5. Then Ada Augusta, Countess of Lovelace, the daughter of Lord Byron, a most romantic figure in the history of

computing and a skilled mathematician became interested in Babbage's work on the analytical engine

6. In 1847, British mathematician George Boole proposed a system of algebra that could be used to manipulate propositions, that is, assertions that could be either true or false.
7. Besides being essential to computer design, Boolean operations are also used to manipulate individual bits in memory…
8. Aided by Hollerith's machines, a census unit was able to process 7,000 records a day for the 1890 census, about ten times the rate in the 1880 count.

2. Answer the following questions:

1. What are the four whole-number operations?
2. Who was the first to introduce punched card input?
3. What did Babbage's analytical engine conceptually include?
4. Why is Ada Byron described as "a most romantic figure in the history of computing"?
5. What is the essence of Boolean logic? Why does it perfectly fit electronic components?
6. Why is the punched card often referred to as the Hollerith card?

3. Translate into English:

Компьютер на паровой тяге

Викторианский компьютер, вернее, его точная копия, будет построена британским программистом Джоном Грехемом-Каммингом. На создание прототипа современного ПК Джону понадобится 400 тыс. фунтов.

Как оказалось, создателем первого ПК был отнюдь не Тьюринг, а Чарльз Бэббидж – британский математик. А сделал он такой прорыв в вычислительных технологиях как раз через год после смерти Пушкина. Кстати, главным источником питания для первого компьютера было не электричество, а пар. Устройство было размером с небольшой грузовик, состоящий из множества разнообразных валов, шатунов, стержней и шестеренок. Место процессора в этом удивительном аппарате занимал барабан со вставными стержнями, очень сильно напоминающий барабан механического пианино.

Материалы, из которых был построен первый ПК, не содержали кремния. В основном это была медь и железо. Хотя расширяемая память была предусмотрена и в этой, самой первой модели ПК. Правда, размер ее был совсем небольшим – 1 килобайт. Прототип современных ПК имел ЦПУ, плоттер, принтер и даже микропрограмму, напоминающую современный биос. А вот для программирования такого устройства использовались пластинки, очень сильно напоминающие перфокарты. Кстати, это устройство изначально было предназначено для широкого круга вычислений.

В устройстве Бэббиджа нет недочетов. Оно совершенно даже по современным рамкам. К тому же математик до самой смерти трудился над усовершенствованием своего изобретения, называя его «аналитической машиной» (Analitical Engine). Впрочем, ни сам Чарльз, ни его сын Генри так и не смогли собрать свою разработку “в железе”. Собственных сбережений в семье не хватило, а правительство не видело смысла вкладывать сумасшедшие деньги в монтаж этой массивной, сложной и весьма дорогостоящей установки.

Чтобы воспроизвести устройство Бэббиджа, Грехем-Камминг будет расшифровывать и изучать записи, которые в данный момент находятся в лондонском музее. Затем будет воссоздана модель аналитической машины. Пока идет массовая кампания по сбору средств на сборку аналитической машины. Уже было получено около 1600 пожертвований.

Напомним, что ранее воссоздать аналитическую машину Чарльза Бэббиджа пытался его сын. То, что у него получилось, хранится ныне в Лондонском музее науки.

4. Give the summary of the text using the key terms.

EARLY COMPUTERS

Read the following words and word combinations and use them for understanding and translation of the text:

warfare	**военные действия**
accurate calculations	**точные вычисления**
application	**применение**
vacuum tube	**электронная лампа**
transmission	**(радио)передача**
impregnable	**неприступный**
to investigate	**исследовать**
to arouse interest	**возбуждать интерес**
underwrite	**гарантировать, подписывать(ся)**
rather than	**скорее чем, а не**
to solve a problem	**решить задачу (проблему)**
viable	**конкурентный, жизнеспособный**
mainstream	**господствующая тенденция**
feature	**черта, свойство**
to maintain	**поддерживать, обслуживать**
routine	**стандартная программа**
fundamental	**краеугольный, основной**
unit	**блок, модуль**
to extract square roots	**извлекать квадратные корни**

The highly industrialized warfare of World War II required the rapid production of a large volume of accurate calculations for such applications as aircraft design, gunnery control, and cryptography. Fortunately, the field was now ripe for the development of programmable digital computers. Many reliable components were available to the computer designer including switches and relays from the telephone industry and card readers and punches (manufactured by Hollerith's descendant, IBM), and vacuum tubes used in radio and other electronics.

Early computing machines included the Mark I, a huge calculator driven by electrical relays and controlled by punched paper tape. Another machine, the prewar Atanasoff-Berry Computer was never completed, but demonstrated the use of electronic (vacuum tube) components, which were much faster than electromechanical relays. Meanwhile, Zuse, a German inventor, built a programmable binary computer that combined a mechanical number storage mechanism with telephone relays. He also proposed building an electronic (vacuum tube) computer, but the German government decided not to support the project.

During the war, British and American code breakers built a specialized electronic computer called Colossus, which read encoded transmissions from tape and broke the code of the supposedly impregnable German Enigma machines. The most viable general-purpose computers were developed by J. Presper Eckert and John Mauchly starting in 1943. The first, ENIAC, was completed in 1946 and had been intended to perform ballistic calculations.

Mark I (and IBM Again)

As a doctoral student at Harvard, Aiken began to investigate the possibility of building a large-scale, programmable, automatic computing device and managed to arouse interest in his project, particularly from Thomas Watson, Sr., head of International Business Machines (IBM). In 1939, IBM agreed to underwrite the building of Aiken's first calculator, the Automatic Sequence Controlled Calculator, which became known as the Harvard Mark I.

Like Babbage, Aiken aimed for a general-purpose programmable machine rather than an assembly of special-purpose arithmetic units. Unlike Babbage, Aiken had access to a variety of tested, reliable components, including card punches, readers, and electric typewriters from IBM and the mechanical electromagnetic relays used for automatic switching in the telephone industry. His machine used decimal numbers rather than the binary numbers of the majority of later computers. Sixty registers held whatever constant data numbers were needed to solve a particular problem. The operator turned a rotary dial to enter each digit of each number. Variable data and program instructions were entered via punched paper tape. Calculations had to be broken down into specific instructions similar to those in later low-level programming languages. The results (usually tables of mathematical function values) could be printed by an electric typewriter or output on punched cards. Huge (about 8 feet [2.4 m] high by 51 feet [15.5 m] long), slow, but reliable, the Mark I worked on a variety of problems during World War II, ranging from equations used in lens design and radar to the designing of the implosive core of an atomic bomb.

Compared to later computers such as the ENIAC and UNIVAC, the sequential calculator, as its name suggests, could only perform operations in the order specified.

ENIAC

The Electronic Numerical Integrator and Computer (ENIAC) was developed by John W. Mauchly and John Presper Eckert, Jr., at the University of Pennsylvania. The machine had been financed by the U.S. army during the Second World War as a calculator for ballistic tables. With Mauchly providing theoretical design work and J. Presper Eckert heading the engineering effort, ENIAC was completed too late to influence the outcome of the war.

By today's standards for electronic computers ENIAC was a grotesque monster. Its thirty separate units, plus power supply and forced-air cooling, weighed over thirty tons. Its 19,000 vacuum tubes, 1,500 relays, and hundreds of thousands of resistors, capacitors, and inductors consumed almost 200 kilowatts of electrical power.

But ENIAC was the prototype from which most other modern computers evolved. It embodied almost all the components and concepts of today's high-speed, electronic digital computers. Its designers conceived what has now become standard circuitry such as the gate, buffer and used a modified Eccles-Jordan flip-flop as a logical, high-speed storage-and-control device.

ENIAC could discriminate the sign of a number, compare quantities for equality, add, subtract, multiply, divide, and extract square roots. ENIAC stored a maximum of twenty 10-digit decimal numbers. Its accumulators combined the functions of an adding machine and storage unit. No central memory unit existed, per se. Storage was localized within the functioning units of the computer.

The primary aim of the designers was to achieve speed by making ENIAC as all-electronic as possible. The only mechanical elements in the final product were actually external to the calculator itself. These were an IBM card reader for input, a card punch for output, and the 1,500 associated relays.

ENIAC, however, did not have any programming interface. For each new program, cables had to be plugged in the right devices, adaptors used on the right connections, dials and switches set for the right values etc. Thus, it need not surprise that the planning of a computation, not only the translation of the mathematics into a general scheme but also the realisation of the scheme as a combination of cables and switch settings, could take weeks and had to be done with utmost care.

ENIAC was frustrating to use because it wouldn't run for more than a few minutes without blowing a tube, which caused the system to stop working. Every time a new problem had to be solved, the staff had to enter the new instructions by rewiring the entire machine. The solution was **the stored program concept**, an idea that occurred to just about everyone working with electronic computers after World War II.

Rather than the program being set up by wiring or simply read sequentially from tape or cards, the program instructions would be stored in memory just like any other data. Besides allowing a computer to fetch instructions at electronic rather than mechanical speeds, storing programs in memory meant that one part of a program could refer to another part during operation, allowing for such mechanisms as branching, looping, the running of subroutines, and even the ability of a program to modify its own instructions.

Notes:

Enigma machine was an electro-mechanical rotor cipher machine used in the 20-th century for enciphering and deciphering secret messages. It was invented by the German engineer Arthur Scherbius at the end of World War I.

J. Presper Eckert and **John Mauchly** designed ENIAC, the first general-purpose electronic digital computer, as well as EDVAC, BINAC and UNIVAC I.

Eccles-Jordan flip-flop - триггер (триггерная система) - класс электронных устройств, обладающих способностью длительно находиться в одном из двух устойчивых состояний и чередовать их под воздействием внешних сигналов. Используются, в основном, в вычислительной технике для организации регистров, счетчиков, процессоров, ОЗУ.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. Meanwhile, Zuse, a German inventor, built a programmable binary computer that combined a

mechanical number storage mechanism with telephone relays.

2. During the war, British and American code breakers built a specialized electronic computer called Colossus, which read encoded transmissions from tape and broke the code of the supposedly impregnable German Enigma machines.
3. Sixty registers held whatever constant data numbers were needed to solve a particular problem.
4. ENIAC could discriminate the sign of a number, compare quantities for equality, add, subtract, multiply, divide, and extract square roots.
5. ENIAC was frustrating to use because it wouldn't run for more than a few minutes without blowing a tube, which caused the system to stop working.
6. Besides allowing a computer to fetch instructions at electronic rather than mechanical speeds, storing programs in memory meant that one part of a program could refer to another part during operation…

2. Answer the following questions:

1. What paved the way for the emergence of the first digital computer?
2. Who backed the Mark I project?
3. What was the fundamental difference between the Mark I and modern computers?
4. Where did the military use the Mark I?
5. What aspects of creating ENIAC did Mauchly and Eckert (respectively) work on? What do you think each aspect encompassed?
6. Which components of the modern computer did ENIAC include?
7. What was ENIAC capable of?
8. What was the major drawback of the ENIAC? What came as a solution?

3. Translate into English:

Первый советский компьютер был создан под руководством Сергея Алексеевича Лебедева (1902—1974). Необходимость создания собственного ЭВМ в СССР была осознана несколько позже, чем в США, так что соответствующие работы начались только с осени 1948 года. Инициаторами проекта выступили ученые-ядерщики — в те годы буквально вся страна работала над атомным проектом.

Для разработки отечественной ЭВМ Лебедеву и его сотрудникам отвели целое крыло двухэтажного здания тайной лаборатории в местечке Феофания под Киевом. По воспоминаниям участников тех событий, работали все члены коллектива без сна и отдыха. Только к концу 1949 определилась принципиальная схема блоков машины. Далее начались чисто технические сложности — те самые, с которыми за несколько лет до этого столкнулись американцы. Но к концу 1950 года вычислительная машина была все-таки построена. С 1952 года на запущенных в масштабное производство МЭСМ-ах решались важнейшие научно-технические задачи из области термоядерных процессов, космических полетов и ракетной техники, дальних линий электропередачи, механики, статистического контроля качества, сверхзвуковой авиации.

В 1952—1953 годах МЭСМ была самой быстродействующей и практически единственной регулярно эксплуатируемой ЭВМ в Европе.

В это время Лебедев и его команда буквально наступала на пятки своим американским и британским коллегам.

Советские ученые, разумеется, знали о разработках западных коллег в области вычислительной техники. Знали и о компьютере ENIAC, который был построен в 1946 году в университете штата Пенсильвания в рамках оборонного проекта Project PX (создание водородной бомбы). Однако разработки советских ученых велись совершенно независимо от западных коллег.

Еще продумывая проект своей машины, Лебедев обосновывает принципы построения ЭВМ с хранимой в памяти программой совершенно независимо от Джона фон Неймана, разработавшего концепцию запоминаемой программы, которая предполагала совместное хранение кодов и данных. Именем Неймана до сих пор называется архитектура, применяемая в современных компьютерах. Разработанные Лебедевым принципы были успешно реализованы в МЭСМ. На основе же концепции Неймана в 1952 году был построен EDVAC.

4. Give the summary of the text using the key terms.

1950s - PRESENT DAY: DECADE BY DECADE GUIDE

Read the following words and word combinations and use them for understanding and translation of the text:

establishment - создание, установление, учреждение, основание
viable - жизнеспособный. конкурентный
rotating magnetic drum - вращающийся магнитный барабан
advent – появление
ceramic substrate - керамическая подложка
assembly process - процесс сборки
versatile - разносторонний, подвижный, изменчивый
compatibility - совместимость
promulgation - распространение
mainstream – господствующая тенденция
developer's kits - комплекты (наборы) разработчика
plug-in chips - чипы программного расширения

expansion bus - шина расширения
kernel – ядро
shell - оболочка
utility program – обслуживающая программа, утилита
competitor – конкурент, участник рынка
to mature – созреть, возмужать
to supplant – вытеснять
belatedly - с запозданием
to plunge - погружать
to vow – клясться, давать обет
seamlessly - легко, беспрепятственно, без проблем
freeware – свободно распространяемое ПО
clock rate- тактовая частота

The 1950s saw the establishment of a small but viable commercial computer industry in the United States and parts of Europe. Eckert and Mauchly formed a company to design and market the UNIVAC. This new generation of computers would incorporate the key concept of the *stored program.*

The UNIVAC became a hit with the public when it was used to correctly predict the outcome of the 1952 presidential election. Forty UNIVACs were eventually built and sold to such customers as the U.S. Census Bureau, the U.S. Army and Air Force, and insurance companies. Several companies had some success in selling computers, but it was IBM that eventually captured the broad business market for mainframe computers.

The 1960s saw the advent of a "solid state" computer design featuring transistors in place of vacuum tubes and the use of ferrite magnetic core memory. These innovations made computers both more compact, more reliable, and less expensive to operate (due to lower power consumption.) There was a natural tendency to increase the capacity of computers by adding more transistors. As the decade progressed, however, the concept of the integrated circuit began to be implemented in computing. The first step in that direction was to attach a number of transistors and other components to a ceramic substrate, creating modules that could be handled and wired more easily during the assembly process.

IBM applied this technology to create what would become one of the most versatile and successful lines in the history of computing, the IBM System/360 computer. This was actually a series of 14 models that offered successively greater memory capacity and processing speed while maintaining compatibility so that programs developed on a smaller, cheaper model would also run on the more expensive machines.

By the mid-1960s, however, a new market segment had come into being: the minicomputer, pioneered by Digital Equipment Corporation (DEC) Architecturally, the mini usually had a shorter data word length than the mainframe, and used indirect addressing for flexibility in accessing memory. Minis were practical for uses in offices and research labs that could not afford a mainframe.

In programming, the main innovation of the 1960s was the promulgation of the first widely-used, high-level programming languages, COBOL (for business) and FORTRAN (for scientific and engineering calculations.) The new higher-level languages made it easier for professionals outside the computer field to learn to program and made the programs themselves more readable, and thus easier to maintain. The invention of the compiler was yet another fruit of the stored program concept.

The 1970s saw minis becoming more powerful and versatile. Meanwhile, at the high end, Seymour Cray left CDC to form Cray Research, a company that would produce the world's fastest supercomputer, the compact, freon-cooled Cray-1. In the mainframe mainstream, IBM's 370 series maintained that company's dominant market share in business computing.

The most striking innovation of the decade, however, was the microcomputer. The microcomputer combined three basic ideas: an integrated circuit so compact that it could be laid on a single silicon chip, the design of that circuit to perform the essential addressing and arithmetic functions required for a computer, and the use of microcode to embody the fundamental instructions. Intel's 4004 introduced in late 1971 was originally designed to sell to a calculator company. When that deal fell through, Intel started distributing the microprocessors in developer's kits to encourage innovators to design computers around them.

Word of the microprocessor spread through the electronic hobbyist community, being given a boost by the January 1975 issue of *Popular Electronics* that featured the Altair computer kit, available from an Albuquerque company called MITS for about $400.

The Altair was hard to build and had very limited memory, but it was soon joined by companies that designed and marketed ready-to-use microcomputer systems, which soon became known as personal computers (PCs). By 1980, entries in the field included Apple (Apple II), Commodore (Pet), and Radio Shack (TRS-80). These computers shared certain common features: a microprocessor, memory in the form of plug-in chips, read-only memory chips containing a rudimentary operating system and a version of the BASIC language, and an expansion bus to which users could connect peripherals such as disk drives or printers.

Meanwhile, programming and the art of software development did not stand still. Innovations of the 1970s included the philosophy of structured programming. New languages such as Pascal and C supported structured programming design to varying degrees. Programmers on college campuses also had access to UNIX, a powerful operating system containing a relatively simple kernel, a shell for interaction with users, and a growing variety of utility programs that could be connected together to solve data processing problems. It was in this environment that the government-funded ARPANET developed protocols for communicating between computers and allowing remote operation of programs. Along with this came e-mail, the sharing of information in newsgroups (Usenet), and a growing web of links between networks that would eventually become the Internet.

In the 1980s, the personal computer came of age. IBM broke from its methodical corporate culture and allowed a design team to come up with a PC that featured an open, expandable architecture. Other companies such as Compaq legally created compatible systems (called "clones"), and "PC-compatible" machines became the industry standard. Under the leadership of Bill Gates, Microsoft gained control of the operating system market and also became the dominant competitor in applications software (particularly office software suites).

Although unable to gain market share comparable to the PC and its clones, Apple's innovative Macintosh, introduced in 1984, adapted research from the Xerox PARC laboratory in user interface design. At a time when PC compatibles were still using Microsoft's text-based MS-DOS, the Mac sported a graphical user interface featuring icons, menus, and buttons, controlled by a mouse. Microsoft responded by developing the broadly similar Windows operating environment, which started out slowly but had become competitive with Apple's by the end of the decade.

The 1980s also saw great growth in networking. University computers running UNIX were increasingly linked through what was becoming the Internet, while office computers increasingly used local area networks (LANs) such as those based on Novell's Netware system. Meanwhile, PCs were also being equipped with modems, enabling users to dial up a growing number of on-line services.

In the programming field a new paradigm, object-oriented programming (OOP) was offered by languages such as Smalltalk and C++, a variant of the popular C language. The federal government adopted the Ada language with its ability to precisely manage program structure and data operations.

By the 1990s, the PC was a mature technology dominated by Microsoft's Windows operating system. UNIX, too, had matured and become the system of choice for university computing and the worldwide Internet, which, at the beginning of the decade was far from friendly for the average consumer user.

This changed when Tim Berners-Lee, a researcher at Geneva's CERN physics lab, adapted hypertext (a way to link documents together) with the Internet protocol to implement the World Wide Web. By 1994, Web browsing software that could display graphics and play sounds was available for Windows-based and other computers.

In the office, the Intranet (a LAN based on the Internet TCP/IP protocol) began to supplant earlier networking schemes. Belatedly recognizing the threat and potential posed by the Internet, Bill Gates plunged Microsoft into the Web server market, included the free Internet Explorer browser with Windows, and vowed that all Microsoft programs would work seamlessly with the Internet.

Moore's Law, the dictum that computer power roughly doubles every 18 months, continued to hold true as PCs went from clock rates of a few tens of mHz to more than 1 gHz.

Beyond 2000

The new millenium began with great hopes, particularly for the Web and multimedia "dot-coms". By 2005 the computing industry in many ways was stronger than ever. On the Web, new software approaches are changing the way services and even applications are delivered. The integration of search engines, mapping, local content, and user participation is changing the relationship between companies and their customers.

Moore's law is now expressed not through faster single processors, but using processors with two, four, or more processing "cores," challenging software designers. Mobile computing is one of the strongest areas of growth, with devices combining voice phone, text messaging, e-mail, and Web browsing.

Computer hardware has evolved from big and bulky to compact and efficient. The monitor, for example, progressed from the large beige cathode ray tube, or CRT, monitors of the 1990s to slimmer, widescreen liquid crystal display, or LCD, monitors that surpassed the sales of CRT monitors starting in 2007. Toward the end of the first decade of the 21st century, developments in computer technology supported dual monitors and 3D LCD monitors. The first optical mouse also began replacing the trackball mouse at the beginning of the century. Essentially, developments in hardware design have increased the accessibility and ease of use of computers.

On the first day of 2001, Microsoft announced that Windows 95, its first operating system that was a commercial success, became a legacy item and the company would no longer sell it. This propelled the production of the later Windows operating systems that control almost 90 percent of the entire market share. Apple, however, has made its way back into the market, not as much because of its easy-to-use operating systems as because of its line of multimedia devices, starting with the iPod in 2000, the iPhone in 2007 and the iPad in 2010. However, both software giants face an emerging trend toward the use of freeware, as free software such as Adobe Reader and OpenOffice becomes increasingly available.

Although the start of the decade saw Internet users keying numbers into a dial-up modem to gain access, by the end of the decade most households had high-speed, broadband Internet, and the rise of Wi-Fi has made wireless Internet access possible through desktop computers, laptops and mobile phones. Social networking became prevalent on the Internet by storm in the first decade.

The industry continues to face formidable challenges ranging from mitigating environmental impact to the shifting of manufacturing and even software development to rapidly growing countries such as India and China.

The Top Ten Computer Trends for the 21st Century

1. Computers will become powerful extensions of human beings designed to augment intelligence, learning, communications, and productivity.
2. Computers will become intuitive - they will “learn,” “recognize,” and “know” what we want, who we are, and even what we desire.
3. Computer chips will be everywhere, and they will become invisible - embedded in everything from brains and hearts, to clothes and toys.
4. Computers will manage essential global systems, such as transportation and food production, better than humans will.
5. Online computer resources will enable us to download applications on-demand via wireless access anywhere and anytime.
6. Computers will become voice-activated, networked, video-enabled, and connected together over the Net, linked with each other and humans.
7. Computers will have digital senses-speech, sight, smell, hearing-enabling them to communicate with humans and other machines.
8. Neural networks and other forms of artificial intelligence will make computers both as smart as humans, and smarter for certain jobs.
9. Human and computer evolution will converge. Synthetic intelligence will greatly enhance the next generations of humans.
10. As computers surpass humans in intelligence, a new digital species and a new culture will evolve that is parallel to ours.

Notes:

UNIVAC - американская компания, подразделение корпорации Remington Rand. Название происходит от первого коммерческого серийного компьютера UNIVAC I

(UNIVersal Automatic Computer I), который был выпущен в 1951 году.

U.S. Census Bureau - Бюро переписи населения США

(Ferrite) magnetic core memory - was the predominant form of random-access computer memory (circa 1955-75)

Commodore International - находившаяся в Западном Честере (штат Пенсильвания) компания производила и продавала персональные компьютеры Commodore и Amiga. Обанкротилась в 1994 году.

RadioShack Corporation is an American franchise of electronics retail stores in the United States, as well as parts of Europe, South America and Africa.

UNIX - семейство переносимых, многозадачных и многопользовательских операционных систем

DEC - Digital Equipment Corporation- американская компьютерная компания, была основана в 1957г.

CDC - Control Data Corporation- американская компания-производитель вычислительной техники (1960-е – 1980-е).

Popular Electronics - an American magazine started by Ziff-Davis Publishing in October,1954 for electronics hobbyists and experimenters.

ARPANET - компьютерная сеть, созданная в 1969 году в США Агенством Министерства обороны США по перспективным исследованиям (DARPA) и явившаяся прототипом сети Интернет.

Xerox PARC laboratory - научно-исследовательский центр, основанный в Калифорнии, в 1970году. В 2002 году PARC стал отдельной компанией (в собственности Xerox).

MS-DOS - Microsoft Disk Operating System- дисковая операционная система для компьютеров на базе архитектуры x86 (80-е годы- сер. 90-х годов)

CERN - the European Organization for Nuclear Research (ЦЕРН- Европейский совет по ядерным исследованиям)

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. These innovations made computers both more compact, more reliable, and less expensive to operate (due to lower power consumption.)
2. As the decade progressed, however, the concept of the integrated circuit began to be implemented in computing.
3. This was actually a series of 14 models that offered successively greater memory capacity and processing speed while maintaining compatibility so that programs developed on a smaller, cheaper model would also run on the more expensive machines.
4. In the 1980s, the personal computer came of age.
5. IBM broke from its methodical corporate culture and allowed a design team to come up with a PC that featured an open, expandable architecture.
6. University computers running UNIX were increasingly linked through what was becoming the Internet…
7. On the first day of 2001, Microsoft announced that Windows 95, its first operating system that was a commercial success, became a legacy item and the company would no longer sell it.
8. The industry continues to face formidable challenges ranging from mitigating environmental impact to the shifting of manufacturing and even software development to rapidly growing countries such as India and China.

2. Answer the following questions:

1. Which part of the computer market did IBM initially dominate?
2. What were the innovations of the 1960s?
3. What was the mainstream of the 1970s?
4. When did the major programming languages appear (list in order of appearance)?
5. Which company was the first to introduce GUI?
6. Who is credited with the creation of the World Wide Web?
7. What does Moore's law state?
8. What are the major trends for the 21st century? Do you agree?

3. Translate into English:

"...В далёкие времена, когда деревья были ниже, а космос ещё так далёк, где-то в конце 50-х прошлого столетия, зарождалась эра вычислительных машин.

Инженеры в белых халатах творили историю.

Транзисторы, диоды, реле, ферритовые кубы… создавались первые ЭВМ.

В стенах МГУ появилась легенда. И имя ей — Сетунь."

В начале 60-х годов МГУ им М.В. Ломоносова была разработана троичная ЭВМ под руководством Н.П. Бруснецова. Новому троичному компьютеру было дано название Сетунь. Машину назвали по имени речки, протекавшей, не далеко от университета, где разрабатывали ЭВМ. Данная машина, по своей элементной базе, относится ко второму поколению компьютеров. Но по своей архитектуре абсолютно отличается от своих современников т.к. основывается на троичной логике. Серийный выпуск Сетунь был не продолжительным, с

1962-1965 год. Но это была первая троичная ЭВМ, выпускаемая серийно.

Ее конструктивные особенности были таковы, что она могла адресовать, одновременно, только один трайт оперативной памяти. Использовалась троичная система счисления: 0, 1, -1. И только для чисел с фиксированной точкой. Оперативная память на ферритовых сердечниках емкостью в 162 трайта. В качестве внешней памяти, использовался магнитный барабан, предшественник современных жестких дисков. На нем вмещалось до 4000 трайт. Пропускная способность шины памяти составляла 54 трайта. Что давало высокую производительность и не слишком частое обращение к медленной внешней памяти. Троичная машина выполняла порядка четырех тысяч операций в секунду. Ввод и вывод происходил через телетайп и перфоленту. Сетунь имела 37 электронных ламп, 300 транзиторов, 4500 полупроводниковых диодов, 7000 ферритовых колец.

Идеи, заложенные в архитектуру первого троичного компьютера и реализованные в "Сетуни", оказались настолько удачными, что в 1967 году было принято решение выпустить её модифицированную версию. Выпущенный в 1970 году вариант обновлённого троичного компьютера получил название "Сетунь-70".

К сожалению, как следует обкатать идеи, реализованные в "Сетуни-70", не получилось, и "Сетунь-70" переселилась на чердак студенческого общежития в главном корпусе МГУ.

Остаётся надеяться, что души "Сетуни" и "Сетуни-70" обретут троичное бессмертие не только в программных эмуляторах, но и в будущих поколениях компьютеров, которые не будут знать, что "третьего не дано".

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- Women in Computer Science
- Bletchley Park and its Codebreakers
- Soviet Cybersavvy
- Computer Generations
- The Life and Death of Moore's Law

COMPUTING COMPONENTS. THE VON NEUMANN ARCHITECTURE.

ARCHITECTURE AND ORGANIZATION

Read the following words and word combinations and use them for understanding and translation of the text:

to adapt to- приспособиться к
instruction set - набор команд
to span - измерять, (перен.) охватывать
to encompass - заключать в себе, охватывать
enhancement - расширение
without regard to - независимо от, безотносительно к
in a sequential fashion - последовательно
explicitly - ясно, подробно
intrinsically - в действительности, по сути
to emerge - появляться
to evince- показывать, доказывать
compatibility - совместимость
interplay (between) - взаимодействие

Computer science often distinguishes between abstraction and implementation - i.e. between the general and the particular. We may examine any computer system at two major levels: its *architecture* and its *organization.*

The **architecture** of a computer system is the abstraction equivalent to the user-visible interface: the structure and the operation of the system as viewed by the assembly language programmer and the compiler-writer. If an architecture is well-designed, well-engineered to adapt to future technologies, it may persist for a decade or longer.

The **organization** of a computer is the realization and construction of that interface and structure out of specific hardware (and possibly software) components. Because of technological advances, any particular implementation (i.e., one model) may only be actively marketed for a relatively short period of time.

Examples of architectural attributes include the instruction set, the number of bits used to represent various data types (e.g., numbers, characters), I/O mechanisms, and techniques for addressing memory. Organizational attributes include those hardware details transparent to the programmer, such as control signals; interfaces between the computer and peripherals; and the memory technology used.

For example, it is an architectural design issue whether a computer will have a multiply instruction. It is an organizational issue whether that instruction will be implemented by a special multiply unit or by a mechanism that makes repeated use of the add unit of the system.

Historically, and still today, the distinction between architecture and organization has been an important one. Many computer manufacturers offer a family of computer models, all with the same architecture but with differences in organization. Consequently, the different models in the family have different price and performance characteristics. Furthermore, a particular architecture may span many years and encompass a number of different computer models, its organization changing with changing technology. A prominent example of both these phenomena is the IBM System 370 architecture. This architecture was first introduced in 1970 and included a number of models. Over the years, IBM has introduced many new models with improved technology to replace older models, offering the customer greater speed, lower cost, or both. These newer models retained the same architecture so that the customer's software investment was protected. Remarkably, the System 370 architecture, with a few enhancements, has survived to this day IBM's mainframe product line.

The modern computer is a remarkably complex and sophisticated device, but its fundamental principles of operation have really changed remarkably little since the earliest mechanical computers, and in particular, since IAS machine designed by John von Neumann at the Institute for Advanced Studies, Princeton, in the 1940s. The so-called **von Neumann architecture** is at the heart of almost every modern information processing system. It is based on three key concepts:

- data and instructions are stored in a single read-write memory,
- the contents of this memory are addressable by location, without regard to the type of data contained there,
- execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.

Contemporary architectures fall into three classes. Complex Instruction Set Computers **(CISC)** typically include large numbers of machine instructions of many different styles. That complexity poses difficulties of implementation, because each style of instruction may require substantial real estate on the computer chip. Reduced Instruction Set Computers **(RISC)** are defined by smaller numbers of machine instructions of very few styles. The savings in space on a computer chip can, in favorable situations, make possible intrinsically faster circuitry. RISC programs can thus potentially execute faster than CISC programs, even though they usually contain more machine instructions. The third and newest class of contemporary architecture, Explicitly Parallel Instruction Computers **(EPIC)**, includes the Itanium architecture.

Computer architectures may also be classified according to the width of the datapath, the internal components through which information flows, e.g., 64 bits for Itanium architecture.

When new architectures emerge, they may appear to be evolutionary because they evince strong family resemblances to earlier architectures from the same vendor. On the other hand, they appear revolutionary because they offer a clean break with the past.

In a class of computers called microcomputers, the relationship between architecture and organization is very close. Changes in technology not only influence organization but also result in the introduction of more powerful and more complex architectures. Generally, there is less of a requirement for generation-to-generation compatibility for these smaller machines. Thus, there is more interplay between organizational and architectural design decisions.

Notes:

IAS - Institute for Advanced Studies, Princeton

CISC (Complex Instruction Set Computer) - процессор со сложным набором команд - традиционная архитектура процессоров

RISC (Reduced Instruction Set Computer) - компьютер с сокращенным набором команд

EPIC (Explicitly Parallel Instruction Computer) – компьютер с заданным параллелизмом команд. В этой технологии компилятор говорит процессору, какие команды можно исполнять параллельно, а какие зависят от других команд

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Furthermore, a particular architecture may span many years and encompass a number of different computer models, its organization changing with changing technology.
2. Remarkably, the System 370 architecture, with a few enhancements, has survived to this day IBM's mainframe product line.

3. That complexity poses difficulties of implementation, because each style of instruction may require substantial real estate on the computer chip.
4. When new architectures emerge, they may appear to be evolutionary because they evince strong family resemblances to earlier architectures from the same vendor.
5. On the other hand, they appear revolutionary because they offer a clean break with the past.
6. Generally, there is less of a requirement for generation-to- generation compatibility for these smaller machines.

2. Answer the following questions:

1. Define the distinction between architecture and organization of a computer.
2. What is a family of computer models?
3. Which spans longer: a computer architecture or a computer model?
4. What are the key concepts of von Neumann architecture?
5. Name the classes of contemporary architectures.
6. What is the relationship between architecture and organization in microcomputers?

3. Translate into English:

Принципы фон Неймана

1. Использование двоичной системы счисления в вычислительных машинах. Преимущество перед десятичной системой заключается в том, что устройства можно делать достаточно простыми.

2. Программное управление ЭВМ. Работа ЭВМ контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом.

3. Память компьютера используется не только для хранения данных, но и программ. При этом и команды программы, и данные кодируются в двоичной системе счисления.

4. Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы. В любой момент можно обратиться к любой ячейке памяти по ее адресу. Этот принцип открыл возможность использовать переменные в программировании.

5. Возможность условного перехода в процессе выполнения программы. Несмотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода.

4. Give the summary of the text using the key terms.

STRUCTURE AND FUNCTIONS

Read the following words and word combinations and use them for understanding and translation of the text:

CPU (Central Processing Unit) – ЦПУ (центральный процессор)
bus - шина
to be referred to as - упоминаться как, называться
control unit - блок управления
arithmetic logic unit (ALU) - арифметическое логическое устройство (АЛУ)
register - регистр. счетчик
binary - двоичный
to be in charge of - быть ответственным за
fetch-execute cycle - цикл выполнения команды
instruction register (IR) - регистр команд
program counter (PC) - счетчик команд
main memory - оперативная память

to accomplish - выполнять
volatile - энергозависимый
cache memory - кэш-память
bus controller - контроллер шины
motherboard - материнская плата

One of the major characteristics of the von Neumann architecture is that the units that process information are separate from the units that store information. This characteristic leads to the following components of the computer:

- **Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as **processor.**
- **Main memory:** Stores data.
- **I/O:** Moves data between the computer and its external environment
- **System interconnection:** The mechanism that provides for communication among CPU, main memory, and I/O. A common example of interconnection is by means of a **system bus,** consisting of a number of conducting wires to which all the other components attach.

These components are interconnected in some fashion to achieve the basic function of the computer, which is to execute programs. There may be one or more of each aforementioned components. Traditionally, there has been just a single processor. In recent years, there has been increasing use of multiple processors in a single computer.

CPU

The most complex component is the CPU, with its major structural components being as follows:

- **Control unit**, which controls the operation of the CPU and hence the computer
- **Arithmetic and logic unit (ALU),** which performs the computer's data processing functions

- Registers, which provide storage internal to the CPU

- CPU interconnection: the mechanism that provides for communication among the control unit, ALU, and registers.

The **ALU** is that part of the computer that actually performs arithmetic and logical operations on data. All of the other elements of the computer system **-** control unit, registers, memory, I/O - are there mainly to bring data into the ALU for it to process and then to take the results back out. In a sense, the ALU is the core or essence of a computer.

An ALU and, indeed, all electronic components in the computer are based on the use of simple digital logic devices that can store binary digits and perform simple Boolean logic operations.

Most modern ALUs have a small amount of special storage units called **registers.** Registers are used to store intermediate values or special data, i.e. information that is needed again immediately. Access to registers is much faster than access to memory locations.

The **control unit** is the organizing force in the computer, for it is in charge of the *fetch- execute cycle*. The steps in the processing cycle are:

- Fetch the next instruction.
- Decode the instruction.
- Get data if needed.
- Execute the instruction.

There are two registers in the control unit. The *instruction register (IR)* contains the instruction that is being executed, and the *program counter (PC)* contains the address of the next instruction to be executed.

There are several approaches to the implementation of the control unit; one common approach is a *microprogrammed* implementation. In essence, a microprogrammed control unit operates by executing microinstructions that define the functionality of the control unit.

Main memory

Generally speaking, memory is a facility for temporarily storing program instructions or data during the course of processing. Within the *main memory* (sometimes called *primary storage)*, instructions and data are stored in distinct locations so that they can be distinguished easily. The main memory is often referred to as **RAM (random access memory).** It acts as a staging post between the hard disk and processor. The more data it is possible to have available in the RAM, the faster the PC will run.

Main memory is attached to the processor via its address and data buses. Each transaction between the CPU and memory is called a bus cycle. The number of data bits a CPU is able to transfer during a single bus cycle affects a computer's performance.

One distinguishing characteristic of RAM is that it is possible both to read data from the memory and to write new data into the memory easily and rapidly. Both the reading and writing are accomplished through the use of electrical signals.

The other distinguishing characteristic of RAM is that it is volatile. A RAM must be provided with a constant power supply. If the power is interrupted, then the data is lost.

The two traditional forms of RAM used in computers are **DRAM** (dynamic RAM) and **SRAM** (static RAM). A dynamic memory cell is simpler and smaller than a static memory cell. Thus, a DRAM is more dense (smaller cells = more cells per unit area) and less expensive than a corresponding SRAM. DRAMs tend to be favored for large memory requirements. SRAMs are generally somewhat faster than DRAMs. Because of these relative characteristics, a static RAM is used for *cache memory* (an intermediate layer of memory that is smaller but faster, and can hold portions of programs that are likely to be used again shortly), and a dynamic RAM is used for main memory.

Input/Output Units

All of the computing power in the world wouldn't be useful if we couldn't input values into the calculations from outside or report to the outside the results of the calculations. Input and output units are the channels through which the computer communicates with the outside world.

An *input unit* is a device through which data and programs from the outside world are entered into the computer. The first input units interpreted holes punched on paper tape or cards. Modern input devices include: the mouse, keyboard, touch screen monitor, scanner, track pad, microphone, joystick, and web camera.

An *output unit* is a device through which results stored in the computer memory are made available to the outside world. The most common output devices are printers, video display terminals, speakers, and plotters.

System interconnection

The memory, the peripherals and the CPU communicate with each other via the *bus*, which carries data around and allows, for example, data to be transferred from disc into main memory. The bus consists of a set of physical wires plus a protocol (of which there are many, for example, PCI, ISA, IDE etc) that is implemented by the *bus controller* that determines which subsystem can communicate (depending on priorities, previous access history, and other factors). In all cases, only one piece of data can be on the bus at any time.

A bus that connects major computer components (processor, memory, I/O) is called a *system bus.* The most common computer interconnection structures are based on the use of one or more system buses.

In a personal computer, the components in a von Neumann machine reside physically in a printed circuit board called *motherboard*. The motherboard also has connections for attaching other devices to the bus such as a mouse, a keyboard, or additional storage device.

Notes:

RAM (Random Access Memory) **-** оперативная память, оперативное запоминающее устройство, ОЗУ

PCI (Peripheral Connect [Component] Interconnect) **-** межсоединение периферийных компонентов, шина PCI

ISA (Industry Standard Architecture) - архитектура шины промышленного стандарта, шина ISA

IDE (Integrated Drive Electronics) **-** встроенный интерфейс накопителей, интерфейс IDE

Assignments

1. Translate the sentences from the text into Russian paying attention to the underlined words and phrases:

1. The more data it is possible to have available in the RAM, the faster the PC will run.
2. A static RAM is used for cache memory (an intermediate layer of memory that is smaller but faster, and can hold portions of programs that are likely to be used again shortly).
3. All of the computing power in the world wouldn't be useful if we couldn't input values into the calculations from the outside or report to the outside the results of the calculations.
4. The first input units interpreted holes punched on paper tape or cards.
5. In all cases, only one piece of data can be on the bus at any time**.**

2. Answer the following questions:

1. Name the components of the computer.
2. What are the major structural components of the CPU?
3. Why is the ALU considered to be the essence of a computer?
4. What kind of information do registers store?
5. What is the control unit responsible for?
6. What kind of information is stored in the main memory?
7. How is the main memory connected to the CPU?
8. Name the distinguishing characteristics of RAM.
9. How do the memory, the CPU, and the peripherals communicate with each other?

3. Translate into English:

а.) Шины, как известно, используются для передачи данных от центрального процессора к другим устройствам персонального компьютера. Для того, чтобы согласовать передачу данных к другим компонентам, работающих на своей частоте, используется чипсет – набор контроллеров, конструктивно объединенных в Северный и Южный мосты. Северный мост отвечает за обмен информацией с оперативной памятью и видеосистемой. Южный – за функционирование других устройств, подключаемых через соответствующие разъемы – жесткие диски, оптические накопители, а также устройств, находящихся на материнской плате (встроенная аудиосистема, сетевое устройство и др.), и для внешних устройств (клавиатура, мышь и т.д.).

б.) Кэш-память – это высокоскоростная память произвольного доступа, используемая процессором компьютера для временного хранения информации. Она увеличивает производительность, поскольку хранит наиболее часто используемые данные « ближе» к процессору, откуда их можно быстрее получить. Хотя оперативная память намного быстрее диска, тем не менее и она не успевает за потребностями процессора. Поэтому данные, которые требуются часто, переносятся на следующий уровень быстрой памяти, называемой кэш-памятью второго уровня. Она может располагаться на отдельной высокоскоростной микросхеме статической памяти (SRAM), установленной в непосредственной близости от процессора (в новых процессорах кэш-память второго уровня интегрирована непосредственно в микросхему процессора).

На более высоком уровне информация, используемая чаще всего, хранится в специальной секции процессора, называемой кэш-памятью первого уровня. Это самая быстрая память.

4. Give the summary of the text using the key terms.

SECONDARY STORAGE DEVICES

Read the following words and word combinations and use them for understanding and translation of the text:

to install- устанавливать
storage device- запоминающее устройство, накопитель
to back up – создавать резервную копию
to discard - отбрасывать, отвергать
time-consuming - трудоемкий, отнимающий много времени
to spin - вращаться, крутиться
to retrieve - находить, извлекать
to vary - меняться. варьировать
capable of - способный к
consistent – совместимый, последовательный
to time information - распределять (синхронизировать) информацию

hence - отсюда, следовательно
high-definition - высокая четкость, с высоким разрешением
to retain - сохранять, удерживать
non-volatile- энергонезависимый
network switch - сетевое переключение
parallel processing system - система параллельной обработки
pipelining - конвейерная обработка
to hand over - передавать
to update - обновлять
to range from…to - варьировать, изменяться

Since most of main memory is volatile and limited, it is essential that there be other types of storage devices where programs and data can be stored when they are no longer being processed or the machine is not turned on. These other types of storage devices (other than main memory) are called secondary or auxiliary storage devices.

Secondary storage devices can be installed within the computer box at the factory or added later as needed. Because these storage devices can store large quantities of data they are also known as mass storage devices.

Magnetic Tape

The first truly mass auxiliary storage device was the *magnetic tape drive*. A magnetic tape drive is most often used to back up the data on a disk in case the disk is ever damaged. Tapes come in several varieties, from small streaming-tape cartridges to large reel-to-reel models.

Tape drives have one serious drawback. In order to access data in the middle of the tape, all the data before the one you want must be accessed and discarded. Although the modern streaming-tape systems have the capability of skipping over segments of tape, the tape must physically move through the read/write heads. Any physical movement of the type is time-consuming.

Magnetic Disks

A disk drive is a cross between a compact disk player and a tape recorder. A read/write head travels across a spinning magnetic disk, retrieving or recording data. Like a compact disk, the heads travel directly to the information desired, and a tape, the information is stored magnetically.

Disks come in several varieties, but they all use a thin disk made of magnetic material. The surface of each disk is logically organized into tracks and sectors. Tracks are concentric circles around the surface of the disk. Each track is divided into sectors. Each sector holds a block of information as a continuous sequence of bits. Although tracks nearer the center look smaller, each track has the same number of sectors, and each sector has the same number of bits. The blocks of data nearer the center are just more densely packed. The actual number of tracks per surface and the number of sectors per track vary, but 512 bytes and 1024 bytes are common. The location of the tracks and sectors are marked magnetically when a disk is formatted; they are not physically part of the disk.

There is a variety of disks. One classification of disks is hard versus floppy. These terms refer to the flexibility of the disk itself. The original floppy disk, introduced in the 1970s, was 8" in diameter and very floppy. Today's generic "floppy" disks are 3-1/2" in diameter, encased in a hard plastic cover, and capable of storing 1.44MB of data.

Hard disks, the disks on the hard drive that comes with the computer, consist of several disks (platters) attached to a spindle that rotates. Each platter has its own read/write head. Hard drives rotate at a much higher speed than floppy drives, and the read/write heads do not actually touch the surface of the platters but, rather, float above them.

Compact Disks

The world of compact disks and their drivers looks like acronym soup: CD-DA, CD-ROM, CD-RW, CD-WORM, DVD.

A CD drive uses a laser to read information stored optically on a plastic disk. Rather than having concentric tracks, there is one track that spirals from the inside out. Like other disks, the track is broken into sectors. Unlike magnetic disks where the tracks near the center are more densely packed, a CD has the data evenly packed over the whole disk, thus more information is stored in the track on the outer edges and read in a single revolution. In order to make the transfer rate consistent throughout the disk, the rotation speed varies depending on the position of the laser beam.

The other letters attached to CD refer to various properties of the disk, such as formatting, and whether or not the information on them can be changed. CD-DA (Compact Disk-Digital Audio) is the format used in audio recordings. Certain fields in the format are used for timing information.

CD-ROM (Read-Only Memory) is the same as CD-DA but the disk is formatted differently. Data is stored in the sectors reserved for timing information in CD-DA. Read-only memory means that the data is permanent on the disk and cannot be changed.

The acronym CD-WORM stands for Write Once, Read Many. This format, which is also called CD-R (Recordable), is used typically for archiving data, where the data is not to be changed after being recorded.

CD-RW (Rewritable) allows you to erase disks and reuse them.

DVD, which stands for Digital Versatile Disk, can store multi-media presentations that combine audio and video. DVDs also come in several formats: DVD-ROM, DVD-R, DVD-RW.

Flash Memory

Flash memory is a type of non-volatile memory that can be electronically erased and reprogrammed. Its name was invented by Toshiba to express how much faster it could be erased- ‘in a flash’, which means ‘very quickly’.

Unlike RAM, which is volatile, flash memory retains the information stored in the chip when the power is turned off. This makes it ideal for use in digital cameras, laptops, network switches, video game cards, mobile phones and portable multimedia players. In addition, it offers fast read access times (although not as fast as RAM), with transfer rates of 12MB per second. Unlike ROM chips, flash memory chips are rewritable, so you can update programs via software.

Flash memory is used in several ways:

- Many PCs have their BIOS (basic input/output system) stored on a flash memory chip, so it can be updated if necessary.
- Modems use flash memory because it allows the manufacturer to support new protocols.
- USB flash drives are used to save and move MP3s and other data files between computers.

New U3 smart drives allow users to store both applications and data. They have two drive partitions and can carry applications that run on the host computer without requiring installation.

Flash memory cards are used to store images on cameras, to back up data on PDAs, to transfer games in video consoles, to record voice and music on MP3 players or to store movies on MP4 players. Their capacity can range from 8 MB to several gigabytes. The only limitation is that flash cards are often not interchangeable.

Non-von Neumann Architectures

The linear fetch-execute cycle of the von Neumann architecture still dominates the technology today. However, since 1990, alternative parallel-processing systems have entered the marketplace. They have the potential to process much more data at much higher speeds.

One approach to parallelism is to have multiple processors apply the same program to multiple data sets. In this approach, processors often execute the same instructions at the same time, i.e., a common program is run at each processor. This approach is called *synchronous processing* and is effective when the same process needs to be applied to many data sets. This approach is similar to that of the NASA backup system in which three computers do the same thing as a security measure. However, here there are multiple processors applying the same process to different data sets in parallel.

Another configuration arranges processors in tandem, where each processor contributes one part to an overall computation. This approach is called *pipelining*, and is reminiscent of an assembly line. When this organization is applied to data, the first processor does the first task. Then the second processor starts working on the output from the first processor, while the first processor applies its computation to the next data set. Eventually, each processor is working on one phase of the job, each getting material or data from the previous stage of processing, and each in turn handing over its work to the next stage.

The third approach is to have different processors doing different things with different data. This configuration allows processors to work independently much of the time, but introduces the problems of coordination among the processors. This leads to a configuration where the processors each have local memory and a shared memory. The processors use the shared memory for communication, and the configuration is thus called a *shared memory* configuration.

Notes:

Per track - на трек, на каждую дорожку

Drive partition - разбиение жесткого диска на логические разделы

NASA (National Aeronautics and Space Administration) backup system - In the early days of manned space flights, NASA used a backup system composed of three mainframe computers, each of which calculated exactly the same thing. If one computer failed, there were still two computers carrying out the necessary calculations. If two computers failed, there was still one computer left to do the necessary processing. If three computers failed - fortunately, that never happened.

PDA (Personal Digital Computer) - карманный персональный компьютер (КПК)

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Since most of main memory is volatile and limited, it is essential that there be other types of storage devices where programs and data can be stored.
2. Rather than having concentric tracks, there is one track that spirals from the inside out.
3. In order to make the transfer rate consistent throughout the disk, the rotation speed varies depending on the position of the laser beam.
4. Eventually, each processor is working on one phase of the job, each getting material or data from the previous stage of processing, and each in turn handing over its work to the next stage.

2. Answer the following questions:

1. Why is it essential that there be other types of storage devices?
2. Name the types of secondary storage devices.
3. Describe the process of formatting a disk.
4. What formats do CDs and DVDs come in?
5. When is synchronous processing used?

6. Why is pipelining compared to an assembly line?
7. What is a shared memory configuration?

3. Translate into English:

С самого начала компьютерной эры существовала необходимость во все более и более производительных системах. В основном это достигалось в результате эволюции технологий производства компьютеров. Наряду с этим имели место попытки использовать несколько процессоров в одной вычислительной системе в расчете на то, что будет достигнуто соответствующее увеличение производительности. Первой такой попыткой, осуществленной в начале 70-х годов, является ILLIAC IV. Сейчас имеется масса параллельных компьютеров и проектов их реализации.

Архитектуры параллельных компьютеров могут значительно отличаться друг от друга. Параллельные компьютеры состоят из трех основных компонентов: процессоры, модули памяти и коммутирующая сеть. Коммутирующая сеть соединяет процессоры друг с другом и иногда также с модулями памяти.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- The hierarchy of computer memory.
- The non-von Neumann architectures.

PROGRAMMING LANGUAGES AND PROGRAM LANGUAGE CONCEPTS

BRIEF HISTORY OF COMPUTER PROGRAMMING LANGUAGES

Read the following words and word combinations and use them for understanding and translation of the text:

to specify - уточнять
to morph into - трансформироваться в ...
logical branching - логическое ветвление
tedious - утомительный
to affect - влиять
shared program - разделенная (совместно используемая) программа
hand-wired - установленный вручную
array - массив
explicit - точный
restrictive - ограничивающий
to handle - загружать, обрабатывать
to surpass - превосходить
parse tree - дерево синтаксического разбора
dynamic variable - динамическая переменная
to wind up - завершиться
to implement - внедрять
sequentially - последовательно
slider bar - скользящий маркер, бегунок
duct tape - скотч
oddball - эксцентричный

Ever since the invention of Charles Babbage's difference engine in 1822, computers have required a means of instructing them to perform a specific task - a **program**. This means is known as a programming language. A **programming language** is a set of rules that specify which sequences of symbols constitute a program, and what computation the program describes.

A programming language is an abstraction mechanism. It enables a programmer to specify a computation abstractly, and to let a program (usually called an assembler, compiler or interpreter) implement the specification in the detailed form needed for execution on a computer. Programming languages are the medium of expression in the art of computer programming. An ideal programming language will make it easy for programmers to write programs clearly. Because programs are meant to be understood, modified, and maintained over their lifetime, a good programming language will help others read programs and understand how they work. In evaluating programming languages, we must consider the tasks of designing, implementing, testing, and maintaining software, asking how well each language supports each part of the software life cycle.

Computer languages were first composed of a series of steps to wire a particular program; these morphed into a series of steps keyed into the computer and then executed; later these languages acquired advanced features such as logical branching and object orientation. The computer languages of the last fifty years have come in two stages, the first major languages and the second major languages, which are in use today.

In the beginning, Charles Babbage's difference engine could only be made to execute tasks by changing the gears which executed the calculations. Thus, the earliest form of a computer language was physical motion. Eventually, physical motion was replaced by electrical signals when the US Government built the ENIAC in 1942. It followed many of the same principles of Babbage's engine and hence, could only be "programmed" by presetting switches and rewiring the entire system for each new "program" or calculation. This process proved to be very tedious.

In 1945, John Von Neumann was working at the Institute for Advanced Study. He developed two important concepts that directly affected the path of computer programming languages. The first was known as “shared-program technique”. This technique stated that the actual computer hardware should be simple and not need to be hand-wired for each program. Instead, complex instructions should be used to control the simple hardware, allowing it to be reprogrammed much faster.

The second concept was also extremely important to the development of programming languages. Von Neumann called it “conditional control transfer”. This idea gave rise to the notion of subroutines, or small blocks of code that could be jumped to in any order, instead of a single set of chronologically ordered steps for the computer to take. The second part of the idea stated that computer code should be able to branch based on logical statements such as IF (expression) THEN, and looped such as with a FOR statement. “Conditional control transfer” gave rise to the idea of “libraries,” which are blocks of code that can be reused over and over.

In 1949, a few years after Von Neumann’s work, the language Short Code appeared. It was the first computer language for electronic devices and it required the programmer to change its statements into 0s and 1s by hand. Still, it was the first step towards the complex languages of today. In 1951, Grace Hopper wrote the first compiler, A-0. A compiler is a program that turns the language’s statements into 0s and 1s for the computer to understand. This led to faster programming, as the programmer no longer had to do the work by hand.

FORTRAN

In 1957, the first of the major languages appeared in the form of FORTRAN. FORTRAN was the first programming language that significantly rose above the level of assembly language. Its name stands for FORmula TRANslating system. The language was designed at IBM for scientific computing. The main innovation of FORTRAN was that it became possible to use ordinary mathematical notation in expressions. FORTRAN also had subroutines (a form of procedure or function), arrays, formatted input and output, and declarations that gave programmers explicit control over the placement of variables and arrays in memory. However, that was about it. Today, this language would be considered restrictive as it only included IF, DO, and GOTO statements, but at the time, these commands were a big step forward.

Nevertheless, the advantages of the abstraction quickly won over most programmers: quicker and more reliable development, and less machine dependence since register and machine instructions are abstracted away. The basic types of data in use today got their start in FORTRAN, these included logical variables (TRUE or FALSE), and integer, real, and double-precision numbers. Because most early computing was on scientific problems, FORTRAN became the standard language in science and engineering, and is only now being replaced by other languages.

COBOL

Though FORTAN was good at handling numbers, it was not so good at handling input and output, which mattered most to business computing. Business computing started to take off in 1959, and because of this, COBOL was developed. The language was designed by a committee consisting of representatives of the US Department of Defense, computer manufacturers and commercial organizations such as insurance companies. The primary designer of COBOL was Grace Murray Hopper, an important computer pioneer.

COBOL was intended to be only a short-range solution until a better design could be created; instead, the language as defined quickly became the most widespread language in its field (as FORTRAN has in science), and for a similar reason: the language provides a natural means of expressing computations that are typical in its field, although current versions of FORTRAN and COBOL differ substantially from forms of these languages of the 1950s. Business data processing is characterized by the need to do relatively simple calculations on vast numbers of complex data records, and COBOL's data structuring capabilities far surpass those of algorithmic languages like FORTRAN or C. It also allowed for these to be grouped into arrays and records, so that data could be tracked and organized better. It is interesting to note that a COBOL program is built in a way similar to an essay, with four or five major sections that build into an elegant whole. COBOL statements also have a very English-like grammar, making it quite easy to learn. All of these features were designed to make it easier for the average business to learn and adopt it.

LISP

In 1958, John McCarthy of MIT created the LISt Processing (or LISP) language. It was designed for Artificial Intelligence (AI) research. Since it was designed for a specialized field, the original release of LISP had a unique syntax: essentially none. Programmers wrote code in parse trees, which are usually a compiler-generated intermediary between higher syntax (such as in C or Java) and lower-level code. Another obvious difference between this language (in original form) and other languages is that the basic and only type of data is the list; in the mid-1960s, LISP acquired other data types. A LISP list is denoted by a sequence of items enclosed by parentheses. LISP programs themselves are written as a set of lists, so that LISP has the unique ability to modify itself, and hence grow on its own. LISP remains in use today because of its highly specialized and abstract nature.

Pascal

Pascal was begun in 1968 by Niklaus Wirth. Its development was mainly out of necessity for a good teaching tool. The motivation for Pascal was to create a language that could be used to demonstrate ideas about type declarations and type checking. In the beginning, the language designers had no hopes for it to enjoy widespread adoption. Instead, they concentrated on developing good tools for teaching such as a debugger and editing system and support for common early microprocessor machines which were in use in teaching institutions.

Pascal was designed in a very orderly approach, it combined many of the best features of the languages in use at the time, COBOL, FORTRAN, and ALGOL. While doing so, many of the irregularities and oddball statements of these languages were cleaned up, which helped it gain users. The combination of features, input/output and solid mathematical features, made it a highly successful language. Pascal also improved the "pointer" data type, a very powerful feature of any language that implements it. It also added a CASE statement, that allowed instructions to branch like a tree.

Pascal also helped the development of dynamic variables, which could be created while a program was being run, through the NEW and DISPOSE commands. However, Pascal did not implement dynamic arrays, or groups of variables, which proved to be needed and led to its downfall. Wirth later created a successor to Pascal, Modula-2, but by the time it appeared, C was gaining popularity and users at a rapid pace.

C

C was developed by Dennis Ritchie of Bell Laboratories in the early 1970s as an implementation language for the UNIX operating system. Operating systems were traditionally written in assembly language because high-level languages were considered inefficient. C abstracts away the details of assembly language programming by offering structured control statements and data structures (arrays and records), while at the same time it retains all the flexibility of low-level programming in assembly language (pointers and bit-level operations).

Since UNIX was readily available to universities, and since it is written in a portable language rather than in raw assembly language, it quickly became the system of choice in academic and research institutions. When new computers and applications moved from these institutions to the commercial marketplace, they took UNIX and C with them.

C is designed to be close to assembly language so it is extremely flexible; the problem is that this flexibility makes it extremely easy to write programs with obscure bugs because unsafe constructs are not checked by the compiler as they would be in Pascal. C is a sharp tool when used expertly on small programs, but can cause serious trouble when used on large software systems developed by teams of varying ability.

The C language was standardized in 1989 by the American National Standards Institute (ANSI); essentially the same standard was adopted by the International Standards Organization (ISO) a year later.

C++

In the 1980s Bjarne Stroustrup, also from Bell Laboratories, used C as the basis of the C++ (known as "C With Classes") language, extending C to include support for object-oriented programming similar to that provided by the Simula language. In addition, C++ fixes many mistakes in C and should be used in preference to C, even on small programs where the object-oriented features may not be needed.

C++ is an evolving language and the natural language to use when upgrading a system written in C.

Java

In the early 1990s, interactive TV was the technology of the future. Sun Microsystems decided that interactive TV needed a special, portable (can run on many types of machines), language. This language eventually became Java. In 1994, the Java project team changed their focus to the Web, which was becoming "the cool thing" after interactive TV failed. The next year, Netscape licensed Java for use in their internet browser, Navigator. At this point, Java became the language of the future and several companies announced applications which would be written in Java, none of which came into use.

Though Java has very lofty goals and is a text-book example of a good language, it may be the “language that wasn’t.” It has serious optimization problems, meaning that programs written in it run very slowly. And Sun has hurt Java’s acceptance by engaging in political battles over it with Microsoft. But Java may wind up as the instructional language of tomorrow as it is truly object-oriented and implements advanced techniques such as true portability of code and garbage collection.

Visual Basic is often taught as a first programming language today as it is based on the BASIC language developed in 1964 by John Kemeny and Thomas Kurtz. BASIC is a very limited language and was designed for non-computer science people. Statements are chiefly run sequentially, but program control can change based on IF..THEN, and GOSUB statements which execute a certain block of code and then return to the original point in the program’s flow.

Microsoft has extended BASIC in its Visual Basic (VB) product. The heart of VB is the form, or blank window on which you drag and drop components such as menus, pictures, and slider bars. These items are known as “widgets.” Widgets have properties (such as its color) and events (such as clicks and double-clicks) and are central to building any user interface today in any language. VB is most often used today to create quick and simple interfaces to other Microsoft products such as Excel and Access without needing a lot of code, though it is possible to create full applications with it.

Perl

Perl has often been described as the “duct tape of the Internet,” because it is most often used as the engine for a Web interface or in scripts that modify configuration files. It has very strong text matching functions which make it ideal for these tasks. Perl was developed by Larry Wall in 1987 because the UNIX sed and awk tools (used for text manipulation) were no longer strong enough to support his needs. Depending on whom you ask, Perl stands for Practical Extraction and Reporting Language or Pathologically Eclectic Rubbish Lister.

Programming languages have been under development for years and will remain so for many years to come. They got their start with a list of steps to wire a computer to perform a task. These steps eventually found their way into software and began to acquire newer and better features. The first major languages were characterized by the simple fact that they were intended for one purpose and one purpose only, while the languages of today are differentiated by the way they are programmed in, as they can be used for almost any purpose. And perhaps the languages of tomorrow will be more natural with the invention of quantum and biological computers.

Notes:

Sed – потоковый редактор

Awk – язык обработки шаблонов

Charles Babbage (1791 — 1871) – английский математик, изобретатель первой аналитической вычислительной машины

ENIAC – первый электронный цифровой компьютер

John von Neumann (1903 – 1957) – венгеро-американский математик, с его именем связывают архитектуру большинства современных компьютеров

Artificial Intelligence – искусственный интеллект

UNIX - семейство переносимых, многозадачных, многопользовательских операционных систем

Widget – примитив графического интерфейса пользователя

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Computer languages were first composed of a series of steps to wire a particular program; these morphed into a series of steps keyed into the computer and then executed; later these languages acquired advanced features such as logical branching and object orientation.
2. ...It followed many of the same principles of Babbage's engine and hence, could only be "programmed" by presetting switches and rewiring the entire system for each new "program" or calculation.
3. The second concept was also extremely important to the development of programming languages. Von Neumann called it "conditional control transfer".
4. The second part of the idea stated that computer code should be able to branch based on logical statements such as IF (expression) THEN, and looped such as with a FOR statement.
5. FORTRAN also had subroutines (a form of procedure or function), arrays, formatted input and output, and declarations that gave programmers explicit control over the placement of variables and arrays in memory.
6. Programmers wrote code in parse trees, which are usually a compiler-generated intermediary between higher syntax (such as in C or Java) and lower-level code.
7. Java may wind up as the instructional language of tomorrow as it is truly object-oriented and implements advanced techniques such as true portability of code and garbage collection.

2. Answer the following questions:

1. What were programming languages devised for?
2. Which two concepts devised by von Neumann affected the development of programming languages?
3. What made FORTRAN so useful for scientific applications?

4. What features of COBOL made it easier for business to adopt it?
5. Is Pascal good for teaching purposes?
6. What are the advantages and disadvantages of the C language?
7. Why does the number of new programming languages tend to increase?

3. Translate into English:

Эволюция языков программирования

Языки программирования претерпели большие изменения с тех пор, как в 40-х годах XX века началось их использование. Первые языки программирования были очень примитивными и мало чем отличались от формализованных упорядочений двоичных чисел (единиц и нулей), понятных компьютеру. Их называют языками программирования низкого уровня. Использование таких языков было крайне неудобно с точки зрения программиста, так как он должен был знать числовые коды всех машинных команд и собственноручно распределять память под команды программы и данные. Чтобы упростить общение человека с компьютером, были разработаны языки программирования типа Ассемблер, в которых переменные величины стали изображаться символическими именами, а числовые коды операций были заменены на мнемонические (словесные) обозначения, которые легче запомнить. Язык программирования приблизился к человеческому языку, но удалился от языка машинных команд.

В 50-х годах XX века в связи с широким применением компьютеров в различных областях науки и техники возникла серьезная проблема: простые пользователи не могли работать с компьютером из-за сложности языков программирования, а профессиональные программисты были не в состоянии обслужить огромное количество пользователей. Решением данной проблемы явилось создание языков программирования высокого уровня, Важным преимуществом языков программирования высокого уровня является машинная независимость. К недостаткам программ, написанных на языках высокого уровня, относятся большой объем занимаемой памяти и более медленное выполнение, чем у программ на машинных языках или языках Ассемблера. Первыми популярными языками высокого уровня, появившимися в 50-х годах XX века, были FORTRAN, COBOL и Algol.

В 1971 году профессор Н. Вирт разработал новый язык, получивший название Pascal (в честь математика XVII века Блеза Паскаля). Язык Pascal основан на Алголе и создавался как учебный язык, в нем строго соблюдена структурная линия программирования. В силу своих достоинств Pascal послужил источником для создания многих современных языков программирования, таких как Ada, С и Modula-2.

Язык С первоначально был разработан для компьютеров, использующих операционную систему UNIX. Он является относительно простым языком, в нем нет операций над символьными строками и списками, но, в отличие от Pascal, в нем заложены возможности непосредственного обращения к некоторым машинным командам, к определенным участкам памяти компьютера. Язык С широко используется как инструментальный язык для разработки операционных систем, трансляторов, баз данных, а также других системных и прикладных программ.

4. Give the summary of the text using the key terms.

GENERATIONS OF LANGUAGES

Read the following words and word combinations and use them for understanding and translation of the text:

generation – поколение
to be referred to as ...- называться
neural networks – нейронные сети
low-level language – язык низкого уровня
high-level language – язык высокого уровня
to execute – выполнять
advantage – преимущество
disadvantage - недостаток
source code – исходный код
loop – цикл
human-readable – понятный человеку
to convert into - преобразовывать

There are currently five generations of computer programming languages. In each generation, the languages syntax has become easier to understand and more human-readable.

- Languages of the First Generation (1GL). These languages represent the earliest form of computer languages, consisting entirely of 0s and 1s. This is the representation of the language the computers actually understand (machine language).
- Languages of the Second Generation (2GL). This language improvement provided a way for programmers to incorporate symbolic names into the language in addition to the numbers. This is also referred to as assembly language, and is converted into 1GL.
- Languages of the Third Generation (3GL). Words and commands were added to the usage of program creation. This is in addition to the numbers and symbols that were previously used in 1GL and 2GL. Also considered as high-level languages, they have a syntax that is relatively easy to

understand. Some examples of programming languages in the third generation include Javascript, Java, C and C++.

- Languages of the Fourth Generation (4GL). These languages are extremely close to the human language, and are generally used for database access. Among others, this language consists of ColdFusion and SQL.
- Languages of the Fifth Generation (5GL). These languages are most commonly used for neural networks, which are forms of artificial intelligence in an attempt to imitate the way the human mind works.

There are three basic types of computer programming languages. They are machine, assembly and high-level.

Machine language

Machine language is the only language that a computer understands. Each statement in a machine language program is a sequence of bits. Each bit may be set to 0 or 1. Series of bits represent instructions that a computer can understand. For example, the number 455 is represented by the bit sequence 111000111. Machine language is a low-level programming language. It is easily understood by computers but difficult to read by people. This is why people use higher level programming languages. Programs written in high-level languages are compiled and/or interpreted into machine language so computers can execute them.

Assembly language

Assembly language is a representation of machine language. In other words, each assembly language instruction translates to a machine language instruction. The advantage of assembly language is that its instructions are readable. For example, assembly language statements like MOV and ADD are more recognizable than sequences of 0s and 1s. Though assembly language statements are readable, the statements are still low-level. Another disadvantage of assembly language is that it is not portable. In other words, assembly language programs are specific to a particular hardware. But this can be an advantage for programmers who are targeting a specific platform and need full control over the hardware.

High-level language

High-level languages are what most programmers use. Languages such as C++ and Java are all high-level languages. One advantage of high-level languages is that they are easily readable. The statements in these languages are English-like. For example, you can gain a basic understanding of what a Java program is doing by simply reading the program source code. High-level languages use English words as statements. Loops in Java programs are indicated by the words *for*, *while* and *do*. Another advantage of high-level languages is that they are less tedious to use. A single statement in a high-level language can translate into many machine language statements. Finally, high-level languages are usually portable.

A disadvantage of high-level languages is that they are usually less powerful and less efficient. Since statements are high-level, you cannot code at the bit level the way you can with assembly language. High-level languages also need to be compiled and/or interpreted into machine language before execution.

Notes:

Neural network – математическая или кибернетическая модель биологической нейронной сети

Machine language – набор бинарных цифр или битов, которые компьютер считывает и воспринимает

Assembly language – промежуточный язык между машинным кодом и языком высокого уровня

High-level language – компьютерный язык высокого уровня, позволяет разрабатывать программные приложения, используя обычные слова и символы, а не двоичные числа

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Languages of the Second Generation provide a way for programmers to incorporate symbolic names into the language in addition to the numbers.
2. Language of the Fourth Generation is extremely close to the human language, and is generally used for database access.
3. Languages of the Fifth Generation are most commonly used for neural networks, which are forms of artificial intelligence in an attempt to imitate the way the human mind works.
4. Programs written in high-level languages are compiled and/or interpreted into machine language so computers can execute them.
5. Assembly language statements like MOV and ADD are more recognizable than sequences of 0s and 1s.
6. You can gain a basic understanding of what a Java program is doing by simply reading the program source code.

2. Answer the following questions:

1. Why are there five generations of programming languages?
2. What is the principal difference between low-level and high-level languages?
3. To what extent is assembly language similar to a machine language?

4. Can the programs written in high-level languages be executed by the computer directly?
5. What are the advantages and disadvantages of high-level languages?

3. Translate into English:

Пять поколений языков программирования

Иногда различают пять поколений языков программирования, правда данное разделение является спорным:

Первое поколение

Начало 1950-х годов - язык первых компьютеров. Первый язык ассемблера, созданный по принципу «одна инструкция - одна строка». Основная отличительная особенность: ориентирование на конкретный компьютер.

Второе поколение

Конец 1950-х - начало 1960-х г.г. Разработан символьный ассемблер, в котором появилось понятие переменной. Это первый полноценный язык. Основная отличительная особенность: ориентирование на абстрактный компьютер с такой же системой команд.

Третье поколение

1960-е г.г. - Языки программирования высокого уровня. Их характеристики:

- относительная простота;
- независимость от конкретного компьютера;
- возможность использования мощных синтаксических конструкций.

Простота языка позволяет писать небольшие программы и людям, которые не являются профессиональными программистами. Основная отличительная особенность языка третьего поколения: ориентирование на алгоритм (алгоритмические языки).

Четвертое поколение

Начало 1970-х г.г. до сегодняшнего времени. Создаются языки, предназначенные для реализации крупных проектов. Проблемно-ориентированные языки, оперирующие конкретными понятиями узкой области. Как правило, в такие языки встраивают мощные операторы, позволяющие одной строкой описывать функции, для описания которых языкам младших поколений потребовалось бы сотни или даже тысячи строк исходного кода. Основная отличительная особенность языка четвертого поколения: приближение к человеческой речи (декларативные языки).

Пятое поколение

Пятого поколения языков программирования пока не существует. По прогнозам, 5GL будет оперировать мета-мета-данными.

Сейчас существует единственный язык, который работает с мета-мета-данными, - это язык команд менеджеров пакетов или менеджеров зависимостей, таких как apt, yum, smart, maven, cpan и другие. Использование apt-get, yum и smart чрезвычайно повысило производительность системных администраторов - примерно в 1000-и раз. Использование менеджеров зависимостей, таких как maven, cpan, rakudo, pim, easy_install, действительно значительно повысило производительность программистов, примерно в 10 раз. К сожалению, эти языки являются языками командной строки и не являются языками программирования.

4. Give the summary of the text using the key terms.

PROGRAMMING LANGUAGE APPLICATIONS

Read the following words and word combinations and use them for understanding and translation of the text:

spreadsheet – (динамическая) электронная таблица, табличная программа
to embed – внедрять, встраивать
diversity – разнообразие
to be content – быть согласным
elaborate – детально разработанный
decimal numbers – десятичные числа
rather than - ...а не...
therefore – следовательно
to burden – обременять
mark-up language – язык разметки
pervasive – широко распространенный
floating-point computations – вычисления с плавающей точкой

Computers have been applied to a myriad of different areas, from controlling nuclear power plants to providing video games in mobile phones. Because of this great diversity in computer use, programming languages with very different goals have been developed. We can also understand why there are hundreds of programming languages: two different classes of problems may demand different levels of abstraction, and different programmers have different ideas on how abstraction should be done. A C programmer is perfectly content to work at a level of abstraction that requires specification of computations using arrays and indices, while an author of a report prefers to “program” using a language consisting of the functions of a word-processor.

Virtually all successful programming languages were originally designed for one specific use. This is not to say that each language is good for only one purpose.

Scientific applications. The first digital computers, which appeared in the 1940s, were used and indeed invented for scientific applications. Typically, scientific applications have simple data structures but require large numbers of floating-point arithmetic computations. The most common data structures are arrays and matrices; the most common control structures are counting loops and selections. The early high-level programming languages invented for scientific applications were designed to provide for those needs. Their competitor was assembly language, so efficiency was a primary concern. The first language for scientific applications was FORTRAN.

Business applications. The use of computers for business applications began in the 1950s. Special computers were developed for this purpose, along with special language. The first successful high-level language for business was COBOL, the initial version of which appeared in 1960. Business languages are characterized by facilities for producing elaborate reports, precise ways of describing and storing decimal numbers and character data, and the ability to specify arithmetic operations.

With the advent of personal computers came new ways for businesses to use computers. Two specific tools that can be used on small computers, spreadsheet systems and database systems, were developed for business and now are widely used, in both homes and businesses.

Artificial Intelligence. Artificial Intelligence (AI) is a broad area of computer applications characterized by the use of symbolic rather than numeric computations. Symbolic computation means that symbols, consisting of names rather than numbers, are manipulated. Also, symbolic computation is more conveniently done with linked lists of data rather than arrays. This kind of programming sometimes requires more flexibility than other programming domains.

The first widely used programming language developed for AI applications was the functional language LISP, which appeared in 1959. During the early 1970s, however, an alternative approach to some of these applications appeared – logic programming using the Prolog language. More recently, some AI applications have been written in scientific languages such as C.

Systems programming. The operating system and all of the programming support tools of a computer system are collectively known as its systems software. Systems software is used continuously and therefore must be efficient. Therefore, a language for this domain must provide fast execution. Furthermore, it must have low-level features that allow the software interfaces to external devices to be written.

In the 1960s and 1970s, some computer manufacturers, such as IBM, Digital, and Burroughs (now UNYSYS), developed special machine-oriented high-level languages for systems software on their machines: PL/S, BLISS, and ALGOL.

The UNIX operating system is written almost entirely in C, which has made it relatively easy to port to different machines. Some of the characteristics of C make it a good choice for systems programming. It is low-level, execution-efficient, and does not burden the user with many safety restrictions.

Web software. The WWW is supported by an eclectic collection of languages, ranging from mark-up languages, such as XHTML, which is not a programming language, to general-purpose programming languages, such as Java. XHTML provides a way of embedding presentation instructions in the pages of information, which could include text, images, sound, or animation, that constitute Web content. These instructions are targeted to presentation devices, which could be browser displays, printers, or other devices. Because of the pervasive need for dynamic Web content, some computation capability is often included in the technology of content presentation. This functionality can be provided by embedding a programming code in an XHTML document. Such code is often in the form of a scripting language, such as PHP or Python.

Notes:

UNYSYS – компьютерная компания в США, основана в 1986 г.

XHTML (Extensible Hypertext Markup Language) – расширяемый язык гипертекстовой разметки

PHP (Hypertext Preprocessor) – скриптовый язык программирования общего назначения, применяемый для разработки веб-приложений

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Computers have been applied to a myriad of different areas, from controlling nuclear power plants to providing video games in mobile phones.
2. A C programmer is perfectly content to work at a level of abstraction that requires specification of computations using arrays and indices, while an author of a report prefers to “program” using a language consisting of the functions of a word-processor.
3. The most common data structures are arrays and matrices; the most common control structures are counting loops and selections.
4. Special computers were developed for this purpose, along with special language.

5. Symbolic computation is more conveniently done with linked lists of data rather than arrays.
6. Systems software is used continuously and therefore must be efficient.
7. It is low-level, execution-efficient, and does not burden the user with many safety restrictions.

2. Answer the following questions:

1. Why are there programming languages with different goals?
2. What makes a programming language successful?
3. What requirements should a language for scientific applications meet?
4. How can business languages be characterized?
5. What is an efficient language for systems programming?
6. Why is the WWW supported by an eclectic collection of languages?

3. Translate into English:

Информационные технологии играют все более значимую роль в человеческом обществе. Они проникли во все сферы деятельности. Для обслуживания общественных потребностей в автоматизации труда, хранения данных, связи и др. развиваются языки программирования. Если раньше языки программирования использовались лишь для создания программ для автоматизации вычислительных процессов, то на сегодняшний день они используются для решения более разнообразных задач.

Изучение истории языков программирования, их разнообразия и особенностей позволяет программисту сделать правильный выбор при выборе языка для решения определенной задачи.

Все многообразие языков программирования делят на различные классы в зависимости от решаемых ими задач.

Было замечено, что в процессе развития языки программирования, входящие в один класс, сближаются между собой. Хотя само разнообразие классов увеличивается, т.к. увеличивается сфера задач, решаемых с помощью компьютерных технологий.

Следует также отметить и развитие языков программирования в сторону спецификации, когда определенные языки наиболее пригодны для решения узкого класса задач.

4. Give the summary of the text using the key terms.

BASIC TYPES OF PROGRAMMING LANGUAGES

Read the following words and word combinations and use them for understanding and translation of the text:

assignment statement – инструкция присваивания
imperative language – императивный язык, процедурный язык
iterative – итеративный, повторный
simulation package – пакет имитационного моделирования
functional, or applicative language – язык функционального программирования, аппликативный язык
conciseness – краткость
to be consistent with – согласовываться с...
overhead - затраты
dispatching – диспетчеризация, управление
to enhance – усиливать
drag-and-drop – буксировка, перетаскивание
control flow – управляющая логика, поток управления
declaration – объявление, описание
query – запрос

to retrieve data – извлекать данные
custom tags – пользовательские теги
data encryption – шифрование данных
gate model – модель логической схемы на вентильном уровне
quantum cellular automata – квантовая сотовая автоматика
to be embedded into – встроенный в...
insofar – в такой степени...

The basic architecture of computers has had a profound effect on language design. Most of the popular languages of the past 50 years have been designed around the von Neumann architecture. These languages are called **imperative languages**.

Because of the von Neumann architecture, the central features of imperative languages are variables, assignment statements, and the iterative form of repetition. In these languages an algorithm is specified in great detail, and the specific order of execution of the instructions or statements must be included.

The syntax and the semantics of the imperative languages are very complex. In an imperative language, the programmer must make a static division of the program into its concurrent parts, which are then written as tasks. Concurrent execution in imperative language can be a complicated process. The most efficient imperative languages are C and FORTRAN.

An imperative language uses a sequence of statements to determine how to reach a certain goal. These statements are said to change the state of the program as each one is executed in turn.

All the languages we have discussed have one trait in common: the basic statement is the assignment statement which commands the computer to move data from one place to another. This is actually a relatively low level of abstraction compared to the level of the problems we want to solve with computers. Newer languages prefer to describe a problem and let the computer figure out how to solve it, rather than specifying in great detail how to move data around.

Modern software packages are really highly abstract programming languages. An application generator lets you describe a series of screen and database structures, and then automatically creates the low-level commands needed to implement the program. Similarly, spreadsheets, desktop publishing software, simulation packages and so on have extensive facilities for abstract programming.

It should be obvious that programs in an abstract, **non-imperative language** cannot hope to be as efficient as hand-coded C programs. Non-imperative languages are to be preferred whenever a software system must search through large amounts of data, or solve problems whose solution cannot be precisely described. Examples are: language processing (translation, style checking), pattern matching (vision, genetics) and process optimization (scheduling). As implementation techniques improve and as it becomes ever more difficult to develop reliable software systems in ordinary languages, these languages will become more widespread.

A functional, or applicative, language is one in which the primary means of computation is applying functions to given parameters. Programming can be done in a functional language without the kind of variables that are used in imperative languages, without assignment statements, and without iteration. This makes the syntax and the semantics of the functional languages simple compared to that of the imperative languages.

Programs in functional languages can be divided into concurrent parts dynamically, by the execution system, making the process highly adaptable to the hardware on which it is running. The closeness of functional programming to mathematics, while resulting in conciseness and elegance, may in fact make functional programming languages less accessible to many programmers.

The most prominent among these languages are: LISP, COMMON LISP, and Scheme, which is widely used to teach functional programming.
In the early days of programming several very influential languages were designed and implemented that had one characteristic in common: the languages each had a preferred data structure and an extensive set of operations for the preferred structure. These languages made it possible to create sophisticated programs that were otherwise difficult to write in languages such as FORTRAN that simply manipulated computer words.

Data-oriented languages are somewhat less popular than they once were, partly because by using object-oriented techniques it is possible to embed such data-oriented operations into ordinary languages like C++ and Ada, but also because of competition from newer language concepts like functional and logic programming. Nevertheless, the languages are technically interesting and quite practical for the programming tasks for which they were designed.

Object-oriented programming (OOP) is a method of structuring programs by identifying real world or other objects, and then writing modules each of which contains all the data and executable statements needed to represent one class of objects. Within such a module, there is a clear distinction between the abstract properties of the class which are exported for use by other objects, and the implementation which is hidden so that it can be modified without affecting the rest of the system.
C++ showed that it was possible to implement the entire machinery of OOP in a manner that is consistent with static allocation and type-checking, and with fixed overhead for dispatching; the dynamic requirements of OOP are used only as needed. Ada 95 based its support for OOP on ideas similar to those found in C++.

The latest step in the evolution of software development is object-oriented design. Object-oriented methodology begins with data abstraction, and adds inheritance and dynamic method binding. Inheritance greatly enhances the potential reuse of existing software, providing the possibility of significant increases in software development productivity. This is an important factor in the increase in the popularity of **object-oriented languages**, such as Smalltalk, Ada 95, Java, and C++.

Another kind of language, **the visual language**, forms subcategory of the imperative languages. The most popular visual language is Visual BASIC, which is now being replaced by Visual BASIC.NET. These languages include capabilities for drag-and-drop generation of code segments. The characterizing feature of a visual language provides a simple way to generate graphical user interfaces to programs.

Languages used for logic programming are called **logic programming languages, or declarative languages**, because programs written in them consist of declarations rather than assignments and control flow statements. These declarations are actually statements in symbolic logic.

Declarative semantics is considerably simpler than the semantics of the imperative languages.

Programming in a logic programming language is nonprocedural. Programs in such languages do not state exactly how a result is to be computed but rather describe the form of the result. The difference is that we assume the computer system can somehow determine how the result is to be computed. What is needed to provide this capability for logic programming languages is a concise means of supplying the computer with both the relevant information and a method of inference for computing the desired result.

Logic programming in general and Prolog language in particular are a natural match to the needs of implementing an RDBMS: only a single language is required, the deductive capability is built in.

Prolog can be used to construct expert systems. It can easily fulfill the basic needs of expert systems, using resolution as the basis for query processing, using its ability to add facts and rules to provide the learning capability, and using its trace facility to inform the user of the 'reasoning' behind a given result.

In recent years, a new category of languages has emerged, **a mark-up/programming hybrid languages**. Mark-up languages, including the most widely used mark-up language, XHTML, are not programming languages. They are used to specify the layout of information in Web documents.

Web Languages are used for creating and editing pages on the Web. They can do anything from putting plain text on Webpage, to accessing and retrieving data from a database and vary greatly in terms of power and complexity.

- **HTML (**Hyper Text Markup Language) is the core language of the World Wide Web that is used to define the structure and layout of Web pages by using various tags and attributes. Although a fundamental language of the Web, HTML is static - content created with it does not change.
- **HML (**Extensible Markup Language) works like HTML, but unlike HTML, allows for custom tags that are defined by programmers. XML allows for the transmission of data between applications and organizations through the use of its custom tags.
- **Javascript** is developed by Netscape used to provide dynamic and interactive content on Webpages. With Javascript it is possible to communicate with HTML, create animations, create calculators, validate forms, and more. Javascript is often confused with Java, but they are two different languages.

- **PHP** (Hypertext Preprocessor (it's a recursive acronym)) is a powerful language used for many tasks such as data encryption, database access, and form validation. PHP was originally created in 1994 by Rasmus Lerdorf.
- **Java is a** powerful and flexible language created by Sun MicroSystems that can be used to create applets (a program that is executed from within another program) that run inside Webpages as well as software applications. Things you can do with Java include interacting with the user, creating graphical programs, reading from files, and more. Java is often confused with Javascript, but they are two different languages.

Software languages are used for creating executable programs. They can create anything from simple console programs that print some text to the screen to entire operating systems and vary greatly in terms of power and complexity.

- **C** is an advanced programming language used for software application development. It has proven itself to be able to be used for various software applications such as business programs, engineering programs, and even games.
- **C++** is a descendant of the C language. The difference between the two languages is that C++ is object-oriented. C++ is a very popular language for graphical applications.
- **Visual Basic** is a language developed by Microsoft based on the BASIC language. Visual Basic is used for creating Windows applications. The VBScript language (also developed by Microsoft) is based on Visual Basic.

Within the area of 'Quantum Computing' a new field of research has emerged: designing and realizing experimental '**Quantum Programming Languages**' (QPLs) complement the research fields of 'computational models' (gate model, one-way quantum computer, quantum cellular automata), 'error correction' and 'quantum algorithms'.

From a pragmatical point of view QPLs are formal systems, which serve as a means to control the execution of programs on a quantum computer or on a classical computer simulating a quantum computer. A possible model describing this scenario is Knill's QRAM model, which is based on the idea that the program proper runs on a classical computer which controls the quantum computer, i.e. which controls a device driving a quantum experiment.

QPLs can also be classified into the traditional categories of functional and imperative/object-oriented programming languages.

Just as with classical programming languages there are many more aspects on QPLs which are a matter of debate. A QPL could be designed as a completely new language or as an extension of an existing classical language. The extension itself could either be embedded into the classical language or be realized in form of a library.

One of the problems which language designers face is the relatively small number of quantum algorithms which could help to demonstrate the expressiveness of their language. Another open problem is the construction of high level structures analogous to the structures, which are nowadays common in all modern programming languages such as modules, abstract data types, etc. Up to now, quantum sublanguages of QPLs are still close to assembly languages insofar as they operate directly on registers of qubits.

Notes:

RDBMS (Relational Database Management System) – объектно-реляционная система управления базами данных компании Oracle.

Quantum gate – базовый элемент квантового компьютера, преобразующий входные данные кубитов в выходные по определенному закону

QRAM (Quantum Random Access Memory) – запоминающее устройство квантового компьютера

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Because of the von Neumann architecture, the central features of imperative languages are variables, assignment statements, and the iterative form of repetition.
2. Newer languages prefer to describe a problem and let the computer figure out how to solve it, rather than specifying in great detail how to move data around.
3. Spreadsheets, desktop publishing software, simulation packages and so on have extensive facilities for abstract programming.
4. A functional, or applicative, language is one in which the primary means of computation is applying functions to given parameters.
5. The closeness of functional programming to mathematics, while resulting in conciseness and elegance, may in fact make functional programming languages less accessible to many programmers.
6. In the early days of programming several very influential languages were designed and implemented that had one characteristic in common: the languages each had a preferred data structure and an extensive set of operations for the preferred structure.

7. C++ showed that it was possible to implement the entire machinery of OOP in a manner that is consistent with static allocation and type-checking, and with fixed overhead for dispatching; the dynamic requirements of OOP are used only as needed.
8. Logic programming in general and Prolog language in particular are a natural match to the needs of implementing an RDBMS: only a single language is required, the deductive capability is built in.

2. Answer the following questions:

1. What are the central features of imperative languages? Why?
2. Is there a principal difference between older and newer languages?
3. How can you justify for the applicability of non-imperative languages?
4. Why are data-oriented languages less popular nowadays?
5. What contributes to the growing popularity of object-oriented languages?
6. What kind of problems do language designers face when developing a quantum programming language?

3. Translate into English:

В развитии языков программирования выделяются два основных направления: процедурное и непроцедурное. В процедурных языках программа явно описывает действия, которые необходимо выполнить, а результат задается способом получения его при помощи некоторой процедуры – определенной последовательности действий. Основными средствами, применяемыми в этих языках, являются величины (в том числе и табличные), присваивания, циклы, процедуры. При построении процедурной программы необходимо ясно представлять, какие действия и в какой последовательности будут производиться при ее выполнении. Непроцедурное (декларативное) программирование появилось в начале 70-х годов, но его развитие началось в 80-е годы в связи с проектом по созданию компьютеров пятого поколения, целью которого явилась подготовка почвы для создания интеллектуальных машин. К непроцедурному программированию относятся функциональные и логические языки. В функциональных языках программа описывает вычисление некоторой функции. Обычно эта функция задается как композиция других, более простых, те в свою очередь разбиваются на еще более простые, и т.д. Один из основных элементов в функциональных языках – рекурсия, т.е. вычисление значения функции через значение этой же функции от других элементов. Наиболее распространенными среди функциональных языков являются Lisp и Refal.

Можно выделить еще один класс языков программирования – объектно-ориентированные языки сверхвысокого уровня. На таких языках не описывают подробной последовательности действий для решения задачи, хотя они содержат элементы процедурного программирования. Объектно-ориентированные языки, благодаря богатому пользовательскому интерфейсу, предлагают человеку решить задачу в удобной для него форме. Примером такого языка может служить язык программирования визуального общения SmallTalk. Трудно провести четкую границу между системами программирования сверхвысокого уровня и прикладным программным обеспечением. Как те, так и другие системы позволяют работать с ними неквалифицированному пользователю, не являющемуся программистом.

4. Give the summary of the text using the key terms.

LANGUAGE EVALUATION CRITERIA. IMPLEMENTATION METHODS

Read the following words and word combinations and use them for understanding and translation of the text:

criterion (pl. criteria) – критерий
maintenance - поддержка
ease – простота
crossover – перекрестный переход
writability – легкость создания программ
domain – предметная область
realm – область действия
two-dimensional – двумерный
time-consuming - времязатратный
pure – чистый
simulation - моделирование
fetch-execute cycle – цикл выборки и исполнения
to debug – отладить
run-time – время исполнения
reliability - надежность

In order to examine and evaluate the concepts of the various constructs and capabilities of programming languages it is necessary to have a set of evaluation criteria.

Readability. Perhaps one of the most important criteria for judging a programming language is the ease with which programs can be read and understood. Before 1970 the primary positive characteristics of programming languages were efficiency and machine readability. Language constructs were designed more from the point of view of the computer than of computer users. In the 1970s, however, the software life cycle concept was developed, maintenance was recognized as a major part of the cycle. Because ease of maintenance is determined by the readability of programs, readability became an important measure of the quality of programs and programming languages. There was a distinct crossover from a focus on machine orientation to a focus on human orientation.

Writability. Writability is a measure of how easily a language can be used to create programs for a chosen problem domain. It is simply not reasonable to compare the writability of two languages in the realm of a particular application when one was designed for that application and the other was not. For example, the writabilities of COBOL and FORTRAN are dramatically different for creating a program to deal with two-dimensional arrays for which FORTRAN is ideal. Their writabilities are also quite different for producing financial reports with complex formats, for which COBOL was designed.

Reliability. A program is said to be reliable if it performs to its specifications under all conditions. Type checking is an important factor in language reliability. The earlier errors in programs are detected, the less expensive it is to make the required repairs.

Cost. The ultimate total cost of a programming language is a function of many of its characteristics. First, there is the cost of training programmers to use the language. Second is the cost of writing programs in the language. Both the cost of training programmers and the cost of writing programs in a language can be significantly reduced in a good programming environment. Third is the cost of compiling programs in the language. Fourth, the cost of executing programs written in a language is greatly influenced by that language's design. The fifth factor is the cost of the language implementation system. Sixth is the cost of poor reliability.

Portability. This is the ease with which programs can be moved from one implementation to another. Portability is most strongly influenced by the degree of standardization of the language, which is a time-consuming and difficult process.

Implementation Methods

Programming languages can be implemented by any of three general methods. At one extreme, programs can be translated into machine language, which can be executed directly on the computer. This method is called **a compiler implementation**, and has the advantage of very fast program execution, once the translation process is complete.

Pure interpretation lies at the opposite end (from compilation) of implementation methods. With this approach, programs are interpreted by another program called an interpreter, with no translation whatever. The interpreter program acts as a software simulation of a machine whose fetch-execute cycle deals with high-level language program statements rather than machine instructions.

Pure interpretation has the advantage of allowing easy implementation of many source-level debugging operations, because all run-time error messages can refer to source-level units. But the execution is 10 to 100 times slower than in compiled systems and it often requires more space.

Some language implementation systems are a compromise between compilers and pure interpreters, they translate high-level language programs to an intermediate language designed to allow easy interpretation. This method is faster than pure interpretation because the source language statements are decoded only once. Such implementations are called **hybrid implementation systems.**

Notes:

Compiler – программа выполняющая компиляцию — трансляцию программы, составленной на языке высокого уровня в эквивалентную программу на языке, близком машинному.

Interpreter – программа, обеспечивающая перевод с алгоритмического языка высокого уровня на машинный с одновременным выполнением операторов программы.

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. In order to examine and evaluate the concepts of the various constructs and capabilities of programming languages it is necessary to have a set of evaluation criteria.
2. Perhaps one of the most important criteria for judging a programming language is the ease with which programs can be read and understood.
3. In the 1970s the software life cycle concept was developed, maintenance was recognized as a major part of the cycle.
4. There was a distinct crossover from a focus on machine orientation to a focus on human orientation.
5. It is simply not reasonable to compare the writability of two languages in the realm of a particular application when one was designed for that application and the other was not.
6. The earlier errors in programs are detected, the less expensive it is to make the required repairs.
7. Both the cost of training programmers and the cost of writing programs in a language can be significantly reduced in a good programming environment.
8. The interpreter program acts as a software simulation of a machine whose fetch-execute cycle deals with high-level language program statements rather than machine instructions.
9. Pure interpretation has the advantage of allowing easy implementation of many source-level debugging operations, because all run-time error messages can refer to source-level units.

2. Answer the following questions:

1. What are the criteria to evaluate the programming languages?
2. Why is readability an important measure of the quality of programming languages?
3. Can the writabilities of different languages be different? Why?
4. What can influence the cost of a programming language?
5. What is the difference between the implementation methods?

3. Translate into English:

Интерпретатор - программа или устройство, осуществляющее пооператорную трансляцию и выполнение исходной программы. В отличие от компилятора, интерпретатор не порождает на выходе программу на машинном языке. Распознав команду исходного языка, он тут же выполняет ее. Как в компиляторах, так и в интерпретаторах используются одинаковые методы анализа исходного текста программы. Но интерпретатор позволяет начать обработку данных после написания даже одной команды. Это делает процесс разработки и отладки программ более гибким. Кроме того, отсутствие выходного машинного кода позволяет не «захламлять» внешние устройства дополнительными файлами, а сам интерпретатор можно достаточно легко адаптировать к любым машинным архитектурам, разработав его только один раз на широко распространенном языке программирования. Поэтому интерпретируемые языки, типа Java Script, VB Script, получили широкое распространение. Недостатком интерпретаторов является низкая скорость выполнения программ. Обычно интерпретируемые программы выполняются в 50-100 раз медленнее программ, написанных в машинных кодах.

4. Give the summary of the text using the key terms.
Topics for essays (you might need additional information):

- Programming Languages Categorization.
- The Diversity in Programming Languages Application.
- Language Evaluation Criteria
- Comparative Analysis of Two Programming Languages Belonging to Two Different Categories

OPERATING SYSTEMS

Read the following words and word combinations and use them for understanding and translation of the text:

application software - прикладное программное обеспечение
system software - системное программное обеспечение
core - ядро, цеитр, сущность
to accomplish - выполнять, достигать
handheld - портативный, переносной
booting - загрузка
dual-boot system - система с двойной загрузкой
kernel - ядро
shell - оболочка
in between - между, посередине
to swap - обменивать,
to amount to - составлять, равняться
time-sharing - разделение времени
offshoot - ответвление
to recapitulate -резюмировать
to debug - отлаживать
command-line - командная строка
to tuck away - спрятать
sophistication - сложность, изощренность

Modern software can be divided into two categories, application software and system software, reflecting this separation of goals**. Application software** is written to address our specific needs - to solve problems in the real world. Word processing program, games, inventory control systems, automobile diagnostic programs are all application software. **System software** manages a computer system at a fundamental level. It provides the tools and an environment in which application software can be created and run. System software often interacts directly with the hardware and provides more functionality than the hardware itself does.

The **operating system** of a computer is the core of its system software. An operating system manages computer resources, such as memory, input/output devices, and provides an interface through which a human can interact with the computer. An amazing aspect of operating systems is how varied they are in accomplishing these tasks. Mainframe operating systems are designed primarily to optimize utilization of hardware. Personal computer (PC) operating systems support complex games, business applications, and everything in between. Operating systems for handheld computers are designed to provide an environment in which a user can easily interface with the computer to execute programs. Thus, some operating systems are designed to be convenient, others to be efficient, and others are some combination of the two.

A computer generally has one operating system that becomes active and takes control when the system is turned on. Computer hardware is wired to initially load a small set of system instructions stored in permanent memory (ROM). These instructions load a larger portion of system software from secondary memory, usually a magnetic disk. Eventually all key elements of the operating system software are loaded, start-up programs are executed, the user interface is provided, and the system is ready for use. This activity is often called **booting** the computer. The term **boot** comes from the idea of "pulling yourself up by your own bootstraps," which is essentially what a computer does when it is turned on.

A computer could have two or more operating systems from which the user chooses when the computer is turned on. The configuration is often called a *dual-boot* or *multi-boot* system. Note that only one operating system is in control of the computer at a given time.

DEVELOPMENT

The earliest computers were started with rudimentary "loader" program that could be used to configure the system to run the main application program. Gradually a more sophisticated way to schedule and load programs, link programs together, and assign system resources to them was developed.

As systems were developed that could run more than one program at a time, the duties of the operating systems became more complex. Programs had to be assigned individual portions of memory and prevented from accidentally overwriting another program's memory area. A technique called *virtual memory* was developed to enable a disk drive to be treated as an extension of the main memory, with data "*swapped*" to and from the disk as necessary. This enabled the computer to run more and/or larger applications. The operating system, too, became larger, amounting to millions of bytes worth of code.

During the 1960s, time sharing became popular particularly on new smaller machines such as the DEC PDP series, allowing multiple users to run programs and otherwise interact with the same computer. Operating systems such as MULTICS and its highly successful offshoot UNIX developed ways to assign security levels to files access levels to users. The UNIX architecture featured a relatively small *kernel* that provides essential process control, memory management, and file system services, while drivers performed the necessary low-level control of devices and a *shell* provided user control.

Starting in the late 1970s, the development of personal computers recapitulated in many ways the earlier evolution of operating systems in the mainframe world. Early microcomputers had a program loader in read-only memory (ROM) and often rudimentary facilities for entering, running, and debugging assembly language programs.

During the 1980s, more complete operating systems appeared in the form of Apple DOS, CP/M, and MS-DOS for IBM PCs. These operating systems provided such facilities as a file system for floppy or hard disk and a command-line interface for running programs or system utilities. These systems could run only one program at a time (although exploiting a little-known feature of MS-DOS allowed additional small programs to be tucked away in memory).

As PC memory increased from 640 KB to multiple megabytes, operating systems became more powerful. Apple Macintosh operating system and Microsoft Windows could manage multiple tasks. Today PC operating systems are comparable in sophistication and capability to those used on mainframes.

An interesting development that began in the mid-1980s is the growth of networks of personal computers running network operating systems and distributed operating systems. Network operating systems are not fundamentally different from single-processor operating systems. They obviously need a network interface controller and some low-level software to drive it, as well as programs to achieve remote login and remote file access. But these additions do not change the essential structure of the operating system.

A distributed operating system, in contrast, is one that appears to its users as a traditional uniprocessor system, even though it is actually composed of multiple processors. Distributed systems often allow applications to run on several processors at the same time, thus requiring more complex processor scheduling algorithms in order to optimize the amount of parallelism.

Notes:

DEC PDP - Digital Equipment Corporation Programmed (Personal) Data Processor - торговая марка корпорации Digital Equipment для выпускающегося ею семейства недорогих миникомпьютеров.

MULTICS - Multiplexed Information and Computing Service - одна из первых операционных систем с разделением времени исполнения программ

Apple DOS - Apple Disk Operating System - Дисковая операционная система

CP/M - Control Program for Microprocessors-операционная систем CP/M - популярная в 1980-х гг. ОС для 8-разрядных ПК

MS/DOS - Microsoft Disk Operating System - дисковая операционная система для компьютеров на базе архитектуры x86 (80-е годы - сер. 90-х годов)

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Eventually, all key elements of the operating system software are loaded, start-up programs are executed, the user interface is provided, and the system is ready for use.
2. Today PC operating systems are comparable in sophistication and capability to those used in mainframes.
3. They obviously need a network interface controller and some low-level software to drive it, as well as programs to achieve remote login and remote file access.
4. Thus, some operating systems are designed to be convenient, others to be efficient, and others some combination of the two.

2. Answer the following questions:

1. What is the difference between application software and system software?
2. Explain the term “booting the computer”.

3. In multiprogramming a technique called virtual memory is used. What does it mean?
4. What is a kernel?
5. Describe the evolution of the operating system.

3. Translate into English:

BIOS - первый шаг к созданию операционных систем.

Первые ЭВМ (40-е годы XX века) в силу специфики применения (решение единственной задачи) не использовали никакой операционной системы. Вскоре ЭВМ начали успешно применять для решения других задач: анализ текстов и решение сложных прикладных задач из области физики. Однако для решения каждой конкретной задачи в то время необходимо было заново написать не только код, реализующий алгоритм решения, но и процедуры ввода-вывода и другие процедуры управления процессом вычисления. Существенные издержки такого подхода вскоре стали очевидными:

- код процедур ввода-вывода обычно является довольно объемным и сложным в отладке;

- необходимость каждый раз заново писать довольно большой вспомогательный код повышает трудоемкость разработки прикладных программ.

Поэтому для разрешения указанных проблем были созданы специальные библиотеки процедур ввода-вывода (BIOS - Basic Input-Output System). Тщательно отлаженные и эффективные процедуры из BIOS можно было легко использовать с любыми новыми программами, не затрачивая время и силы на разработку и отладку стандартных процедур для ввода и вывода данных.

Таким образом, с появлением BIOS программное обеспечение разделилось на системное и прикладное программное обеспечение. Причем прикладное программное обеспечение непосредственно ориентировано на решение полезных задач, в то время как системное программное обеспечение ориентировано исключительно на поддержку работы и упрощение разработки прикладного программного обеспечения.

Однако BIOS еще не является операционной системой, так как не выполняет важнейшую для операционной системы функцию- управление процессом вычислений прикладной программы. BIOS и библиотеки математических процедур, которые появились примерно в то же время, просто облегчали процесс разработки и отладки прикладных программ. Тем не менее, создание BIOS стало первым шагом на пути к созданию полноценной операционной системы.

4. Give the summary of the text using the key terms.

FUNCTIONS

Read the following words and word combinations and use them for understanding and translation of the text:

memory management - управление памятью
thread - поток (единица диспетчеризации в современных ОС)
to allocate - распределять, выделять
logical address - логический адрес, адрес в виртуальной памяти
physical address - физический адрес
mapping - преобразование, отображение
to keep track - отслеживать
contiguous - смежный, прилегающий, непрерывный
partition - разделение
fixed partition - статическое (фиксированное) распределение памяти
dynamic partition - динамическое распределение памяти

first fit - метод первого подходящего
best fit - метод наилучшей подкачки, наилучшее размещение
frame- рамка, фрейм
paging - подкачка (замещение) страниц, страничная организация памяти
demand paging - замещение страниц по запросу
process management - управление процессом
process control block (PCB) - блок управления процессом
context switch - контекстный коммутатор, переключение в зависимости от контекста
CPU scheduling - планирование (диспетчеризация) процессора
non-preemptive scheduling - невытесняющее (бесприоритетное) планирование
preemptive scheduling - вытесняющее планирование
full-fledged - полноценный
coherence - согласованность, слаженность
snooping - отслеживание
deadlock - взаимоблокировка, зависание

While the architecture and features of operating systems differ considerably, there are general functions common to almost every system. The "core" functions include "booting" the system and initializing devices, process management (loading programs into memory assigning them a share of processing time), and allowing processes to communicate with the operating system or one another (kernel). Multiprogramming systems often implement not only processes (running programs) but also *threads*, or sections of code within programs that can be controlled separately.

A **memory management** scheme is used to organize and address memory, handle requests to allocate memory, free up memory no longer being used, and rearrange memory to maximize the useful amount.

In a multiprogramming environment, multiple programs are stored in main memory at the same time. Thus, operating systems must employ techniques to:

- track where and how a program resides in memory,
- convert logical program addresses into actual memory addresses.

A logical address (sometimes called a virtual or relative address) is a value that specifies a generic location, relative to the program but not to the reality of main memory. A *physical address* is an actual address in the main memory device. When the program is eventually loaded into memory, each logical address finally corresponds to a specific physical address. The mapping of a logical address to a physical address is called address binding. Logical addresses allow a program to be moved around in memory or loaded in different places at different times. As long as we keep track of where the program is stored, we are always able to determine the physical address that corresponds to any given logical address. There are three techniques:

- single contiguous memory management,
- partition memory management,
- paged memory management.

Single contiguous memory management is the approach to memory management in which the entire application program is loaded into one continuous area of memory. Only one program other than the operating system can be processed at one time. The advantage of this approach is that it is simple to implement and manage. However, memory space and CPU time are almost certainly wasted. It is unlikely that an application program needs all of the memory not used by the operating system, and CPU time is wasted when the program has to wait for some resource.

A more sophisticated approach - *partition memory management* - is to have more than one application program in memory at a time, sharing memory space and CPU time. Thus, memory must be divided into more than two partitions. There are two strategies that can be used to partition memory: fixed partitions and dynamic partitions. When using *fixed partitions*, main memory is divided into a particular number of partitions. The partitions do not have to be the same size, but their size is fixed when the operating system initially boots. The OS keeps a table of addresses at which each partition begins and the length of the partition.

When using *dynamic partitions*, the partitions are created to fit the need of the programs. Main memory is initially viewed as one large empty partition. As programs are loaded, space is "carved out", using only the space needed to accommodate the program and leaving a new, smaller empty partition, which may be used by another program later. The OS maintains a table of partition information, but in dynamic partitions the address information changes as programs come and go. At any point in time in both fixed and dynamic partitions, memory is divided into a set of partitions, some empty and some allocated to programs.

Which partition should we allocate to a new program? There are three general approaches to partition selection:

- *First fit,* in which the program is allocated to the first partition big enough to hold it
- *Best fit,* in which the program is allocated to the smallest partition big enough to hold it
- *Worst fit,* in which the program is allocated to the largest partition big enough to hold it.

Worst fit does not make sense to use in fixed partitions because it would waste the larger partitions. First fit or best fit work for fixed partitions. But in dynamic partitions, worst fit often works best because it leaves the largest possible empty partition, which may accommodate another program later on.

Partition memory management makes efficient use of main memory by having several programs in memory at one time.

Paged memory management puts much more burden on the operating system to keep track of allocated memory and to resolve addresses. But the benefits gained by this approach are generally worth the extra effort.

In the paged memory management, main memory is divided into small fixed-size blocks of storage called *frames*. A process is divided into *pages* that we assume are the same size as a frame. When a program is to be executed, the pages of the process are loaded into the various unused frames distributed through memory. Thus, the pages of a process may be scattered around, out of order, and mixed among the pages of other processes. To keep track of all this, the OS maintains a separate page-map table (PMT) for each process in memory; it maps each page to the frame in which it is loaded.

The advantage of paging is that a process no longer needs to be stored contiguously in memory. The ability to divide a process into pieces changes the challenge of loading a process from finding one available large chunk of space to finding enough small chunks.

An important extension to the idea of paged memory management is the idea of *demand paging,* which takes advantage of the fact that not all parts of a program actually have to be in memory at the same time. At any given instance in time, the CPU is accessing one page of a process. At that point, it does not really matter if the other pages of that process are even in memory.

Process management. Another important resource that an operating system must manage is the use of the CPU by individual processes. Processes move through specific states as they are managed in a computer system. A process enters the system (the new state), is ready to be executed (the ready state), is executing (the running state), is waiting for a resource (the waiting state), or is finished (the terminated state). Note that many processes may be in the ready state or the waiting state at the same time, but only one process can be in the running state.

While running, the process might be interrupted by the operating system to allow another process its chance on CPU.

In that case, the process simply returns to the ready state. Or, a running process might request a resource that is not available or require I/O to retrieve a newly referenced part of the process, in which case it is moved to the waiting state. A running process finally gets enough CPU time to complete its processing and terminate normally. When a waiting process gets the resource it is waiting for, it moves to the ready state again.

The OS must manage a large amount of data for each active process. Usually that data is stored in a data structure called *a process control block (PCB*). Generally, each state is represented by a list of PCBs, one for each process in that state. When a process moves from one state to another, its corresponding PCB is moved from one state list to another in the operating system. A new PCB is created when a process is first created (the new state) and is kept around until the process terminates.

The PCB stores a variety of information about the process, including the current value of the program counter, which indicates which instruction in the process is to be executed next. As the life cycle indicates, a process may be interrupted many times during its execution. *Interrupts* are handled by the operating system's kernel. Interrupts may come from either the computer's hardware or from the running program. At each point, its program counter must be stored so that the next time it gets into the running state it can pick up where it left off.

The PCB also stores the values of all other CPU registers for that process. These registers contain the values for the currently executing process (the one in the running state). Each time a process is moved to the running state, the register values for the currently running process are stored into its PCB, and the register values of the new running state are loaded into the CPU. This exchange of register information, which occurs when one process is removed from the CPU and another takes its place, is called a *context switch*.

PCB also maintains information about CPU scheduling.

CPU scheduling is the act of determining which process in the ready state should be moved to the running state. There are two types of CPU scheduling:

- *non-preemptive scheduling*, which occurs when the currently executing process gives up the CPU voluntarily (when a process switches from the running state to the waiting state, or when a program terminates);
- *preemptive scheduling*, which occurs when the operating system decides to favor another process preempting the currently executing process.

First-come, first-served CPU scheduling gives priority to the earliest arriving job. The-shortest-job-next algorithm gives priority to jobs with short running times. Round-robin scheduling rotates the CPU among active processes giving a little time to each.

For many applications, a process needs exclusive access to not one resource, but several. Suppose, for example, two processes each want to record a scanned document on a CD. Process A requests permission to use the scanner and is granted it. Process B is programmed differently and requests the CD recorder first and is also granted it. Now A asks for the CD recorder, but the request is denied until B releases it. Unfortunately, instead of releasing the CD recorder B asks for the scanner. At this point both processes are blocked. This situation is called a *deadlock.* Deadlocks can occur both on hardware and software resources.

Features

Multiprocessing

Multiprocessing involves the use of more than one processing unit, which increases the power of a computer.

Multiprocessing can be either asymmetric or symmetric. Asymmetric multiprocessing essentially maintains a single main flow of execution with certain tasks being "handed over" by the CPU to auxiliary processors.

Symmetric multiprocessing (SMP) has multiple, full-fledged CPUs, each capable of the full range of operations. The processors share the same memory space, which requires that each processor that accesses a given memory location be able to retrieve the same value. This *coherence* of memory is threatened if one processor is in the midst of a memory access while another is trying to write data to that same memory location. This is usually handled by a "locking" mechanism that prevents two processors from simultaneously accessing the same location.

A subtler problem occurs with the use by processors of separate internal memory for storing data that is likely to be needed. One way to deal with this problem is called *bus snooping*. Each CPU includes a controller that monitors the data line for memory location being used by other CPUs. Alternatively, all CPUs can be given a single shared cache. While less complicated, this approach limits the number of CPUs to the maximum data-handling capacity of the bus.

Larger-scale multiprocessing systems consist of latticelike arrays of hundreds or even thousands of CPUs, which are referred to as *nodes.*

Multiprogramming

In order for a program to take advantage of the ability to run on multiple CPUs, the operating system must have facilities to support multiprocessing, and the program must be structured so that its various tasks are most efficiently distributed among the CPUs. These separate tasks are generally called *threads.* A single program can have many threads, each executing separately, perhaps on a different CPU, although that is not required.

The operating system can use a number of approaches to scheduling the execution of processes or threads. It can simply assign the next idle (available) CPU to the thread. It can also give some threads higher priority for access to CPUs, or let a thread continue to own its CPU until it has been idle for some specified time.

The use of threads is particularly natural for applications where a number of activities must be carried on simultaneously.

Support for multiprogramming and threads can now be found in versions of most popular programming languages, and some languages such as Java are explicitly designed to accommodate it.

Multiprogramming often uses groups or clusters of separate machines linked by a network. Running software on such systems involves the use of communication protocols such as MPI (message-passing interface).

Multitasking

Users of modern operating systems such as Microsoft Windows are familiar with multitasking, or running several programs at the same time. Each running program takes turns in using the PC's central processor. In early versions of Windows, multitasking was cooperative, with each program expected to periodically yield the processor to the Windows so it could be assigned to the next program in the queue. Modern versions of Windows (as well as operating systems such as UNIX) use preemptive multitasking. The operating system assigns a slice of processing (CPU) time to a program and then switches it to the next program regardless of what of what might be happening to the previous program.

Systems with preemptive multitasking often give programs or tasks different levels of priority that determine how big a slice of CPU time they will get. Also, the operating system can more intelligently assign CPU time according to what a given program is doing.

Even operating systems with preemptive multitasking can provide facilities that programs can use to communicate their own sense of their priority. In UNIX systems, this is referred to as niceness. A nice program gives the operating system permission to interrupt lengthy calculations so other programs can have a turn, even if the program's priority would ordinarily entitle it to a greater share of the CPU.

Multitasking should be distinguished from two similar-sounding terms. Multitasking refers to entirely separate programs taking turns executing on a single CPU.

Multithreading, on the other hand, refers to separate pieces of code within a program executing simultaneously but sharing the program's common memory space. Finally, multiprocessing or parallel processing refers to the use of more than one CPU in a system, with each program or thread having its own CPU.

Notes:
PCB (Process Control Block) - блок управления процессором (БУП)
PMT (Page-Map Table) - таблица страниц

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. As long as we keep track of where the program is stored, we are always able to determine the physical address that corresponds to any given logical address.
2. At any point in time in both fixed and dynamic partitions, memory is divided into a set of partitions, some empty and some allocated to programs.
3. Paged memory management puts much more burden on the operating system to keep track of allocated memory and to resolve addresses.
4. Thus, the pages of a process may be scattered around, out of order, and mixed among the pages of other processes.
5. Interrupts may come from either the computer's hardware or from the running program.
6. Symmetric multiprocessing (SMP) has multiple, full-fledged CPUs, each capable of the full range of operations.

7. The processors share the same memory space, which requires that each processor that accesses a given memory location be able to retrieve the same value.

2. Answer the following questions:

1. Name general functions common to almost every operating system.
2. What is memory management used for?
3. Distinguish between a logical address and a physical address.
4. Name three memory management techniques and describe them.
5. Distinguish between a fixed partition and a dynamic partition.
6. What specific states does a process move through as it is managed by an OS?
7. Define process control block (PCB). What information does it store?
8. What is a context switch?
9. There are two types of CPU scheduling. What are they?
10. Distinguish between multitasking, multithreading and multiprocessing.

3. Translate into English:

Ядро - центральная часть операционной системы, управляющая выполнением процессов, ресурсами вычислительной системы и предоставляющая процессам координированный доступ к этим ресурсам. Основными ресурсами являются процессорное время, память и устройства ввода-вывода. Доступ к файловой системе и сетевое взаимодействие также могут быть реализованы на уровне ядра.

Как основной элемент операционной системы, ядро представляет собой наиболее низкий уровень абстракции для доступа приложений к ресурсам вычислительной системы, необходимым для их работы. Как правило, ядро предоставляет такой доступ исполняемым процессам соответствующих приложений за счет использования механизмов межпроцессного взаимодействия и обращения приложений к системным вызовам ОС.

4. Give the summary of the text using the key terms.

EXAMPLES OF OPERATING SYSTEMS

Read the following words and word combinations and use them for understanding and translation of the text:

proprietary - патентованный, частный
command-line driven - управляемый командной строкой
to release - выпускать
to be targeted at - быть направленным на
high-end server - сервер высокого класса, лидирующий сервер
to evolve - развиваться
incrementally - постепенно, поэтапно
along with - наряду с
robust - надежный, устойчивый к ошибкам
to enhance - повышать, усиливать. совершенствовать
consistent - последовательный, совместимый
learning curve - кривая обучения, распределение сложности
elaborate - сложный, тщательно разработанный
versatility - многосторонность, гибкость
to overshadow - затмевать
progeny - потомок, результат
to spring from - возникать
maverick - имеющий независимое мнение
viable - жизнеспособный, реальный

Microsoft Windows

Often simply called Windows, Microsoft Windows is a family of proprietary operating systems. Windows PCs run Intel or Intel-compatible microprocessors and use IBM-compatible hardware architecture. It originated in 1981 as an add-on to the older MS-DOS operating system for the IBM PC, which was command-line driven. Released in 1985 (Windows 1.0), Microsoft struggled to improve Windows through the rest of the 1980s.

Windows called NT (New Technology) was first released in 1993. NT, which progressed through several versions, was targeted at the high-end server market, while the consumer version of Windows continued to evolve incrementally as Windows 95 and Windows 98 (released in those respective years). These versions included improved support for networking (including TCP/IP, the Internet standard) and a feature called "plug and play" that allowed automatic installation of drivers for new hardware.

Toward the end of the century, Microsoft began to merge the consumer and server versions of Windows. Windows 2000 incorporated some NT features and provided somewhat greater security and stability for consumers. With Windows XP, released in 2001, the separate consumer and NT versions of Windows disappeared entirely, to be replaced by home and "professional' versions of XP.

Windows XP supports multiple operating environments and symmetric multiprocessing. The use of kernel objects to provide basic services, along with support for client-server computing, enables Windows XP to support a wide variety of application environments. It provides virtual memory, integrated caching, and preemptive scheduling. Windows XP supports a security model stronger than those of previous Microsoft operating systems and includes internationalization features.

Introduced in early 2007, Microsoft Windows Vista includes a number of new features, including a 3D user interface ("Aero"), easier and more robust networking, built-in multimedia capabilities (such as photo management and DVD authoring), improved file navigation and desktop search. Perhaps the most important feature is enhanced security, including User Account Control, which halts suspect programs and requests permission for them to continue.

Windows 7 was released to manufacturing on July 22, 2009. Unlike its predecessor, Windows 7 was intended to be a more focused, incremental upgrade to the Windows line. Presentations given by Microsoft in 2008 focused on multi-touch support, a redesigned Windows Shell with a new taskbar, referred to as the Superbar, a home networking system called HomeGroup and performance improvements.

Windows 8, which was released to manufacturing on August 1, 2012, introduced major changes to the operating system's platform and user interface to improve its user experience on tablets, where Windows was now competing with mobile operating systems. These changes included a touch-optimized Windows shell, the Start screen (which displays programs and dynamically updated content on a grid of tiles), a new platform for developing apps with an emphasis on a touchscreen input, integration with Microsoft SmartScreen phishing filtering service.

From the user's point of view, Windows is a way to control and view what is going on with the computer. The user interface consists of a standard set of objects (windows, menus, buttons sliders, and so on) that behave in generally consistent way. This consistency reduces the learning curve for mastering a new application.

Windows includes a number of features designed to make it easier for users to control their PC. Most settings can be specified through windows called dialog boxes, which include buttons, check boxes or other controls.

From the programmer's point of view, Windows is a multitasked, event-driven environment. Responsible programs allocate no more memory than they need, and release memory as soon as it is no longer needed. If the pool of free memory becomes too low, Windows starts swapping the least recently used segments of memory to the hard drive. This scheme, called virtual memory, allows a PC to run more and larger programs than would otherwise be possible.

While Windows still remains the dominant PC operating system with tens of thousands of applications and at least several hundred million users around the world, it is likely that the PC operating systems of 2020 will be as different from today's Windows as the latter is from the MSDOS of the early 1980s

UNIX/Linux

By the 1970s, time-sharing computer systems were in use at many universities and engineering and research organizations. Such systems required a new kind of operating system that could manage the resources for each user as well as the running of multiple programs. An elaborate project called MULTICS had been begun in the 1960s in an attempt to create such an operating system. However, as the project began to bog down, two of its participants, Ken Thompson and Dennis Ritchie decided to create a simple, more practical operating system for their PDP-7. The result would become UNIX.

The essential core of the UNIX system is the kernel, which provides facilities to organize and access files, move data to and from devices, and control the running of programs. In designing UNIX, Thompson deliberately kept the kernel small, noting that he wanted maximum flexibility for users.

A UNIX system typically has many users, each of whom may be running a number of programs. The interface that processes user commands is called the *shell*. It is important to note that in UNIX a shell is just another program, so there can be (and are) many different shells reflecting varying tastes and purposes. Traditional UNIX shells include the Bourne shell (sh), C shell (csh), and Korn shell (ksh). Modern UNIX systems can also have graphical user interfaces similar to those found on Windows and Macintosh personal computers.

The versatility of UNIX quickly made it the operating system of choice for many most campuses and laboratories, as well as for many software developers.

Although UNIX has been somewhat overshadowed by its Linux progeny, a variety of open-source versions of traditional UNIX systems have become available. In 2005 Sun Microsystems released OpenSolaris (based on UNIX system V). There is also OpenBSD, derived from the UC Berkeley Software Distribution (but with stronger security features) and available for most major platforms. Finally, the continuing influence of UNIX can also be seen in the current generation of operating systems for Apple Macintosh (OS X).

LINUX is an increasingly popular alternative to proprietary operating systems. Its development sprang from two sources. First was the creation of open-source versions of UNIX utilities by maverick programmer Richard Stallman as part of the GNU project during the 1980s. Starting in 1991, another creative programmer, Linus Torvalds, began to release open-source versions of the UNIX kernel. The combination of the kernel and utilities became known as Linux (a combination of Linus and UNIX).

As an open-source product, LINUX is continually being developed by a community of thousands of loosely organized programmers.

LINUX is very versatile and probably runs on more kinds of devices than any other operating system. These include supercomputer clusters, Web and file servers, desktops (including PCs designed for Windows and Macs), laptops, PDAs, and even a few smart phones.

A Linux distribution such as UBUNTU is now a viable alternative to Windows unless one has to use certain programs that do not have Linux versions. However, such options as dual-booting, emulation, or virtual machines offer the ability to use both LINUX and Windows on the same machine.

OS X

Jaguar, panther, tiger, and leopard - these and other names of sleek big cats represent versions of Apple's Macintosh operating system, OS X (pronounced "OS 10"). Unlike the previous Mac OS, OS X, while broadly maintaining Apple's user interface style, is based on a version of UNIX called OpenStep developed by NeXT starting in the 1980s.

At the core of OS X is a free and open-source version of UNIX called 'Darwin", with a kernel XNU. On top of this Apple built a distinctive and subtly colorful user interface called Aqua and a new version of the Macintosh Finder file and program management system.

OS X introduced a number of new capabilities to provide a more stable and reliable platform than its predecessor, Mac OS 9. For example, pre-emptive multitasking and memory protection improved the system's ability to run multiple applications simultaneously without them interrupting or corrupting each other.

Notes:

TCP/IP - Transmission Control Protocol/Internet Protocol - набор сетевых протоколов передачи данных

"Plug and Play" - принцип « подключи и работай»

User Account Control - контроль учетной записи пользователя

Superbar - «суперпанель» задач в Windows7

Phishing - (от fishing-рыбная ловля, выуживание)- вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей - логинам и паролям.

PDP-7 - миникомпьютер, производившийся Digital Equipment Corporation (1965г.)

OpenBSD (Berkeley Software Distribution) - свободная многоплатформенная операционная система, основанная на BSD- реализации UNIX-системы

GNU - рекурсивное сокращение фразы “GNU is Not UNIX”- свободна операционная система типа UNIX
PDA - Personal Digital Assistant- карманный персональный компьютер (КПК)
UBUNTU - (от зулу ubuntu –“человечность”)- операционная система, основанная на Debian GNU/Linux.
XNU - акроним англ. “X is Not UNIX”- ядро операционных систем, используемое в ОС семейства OS X

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Toward the end of the century, Microsoft began to merge the consumer and server versions of Windows.
2. Unlike its predecessor Windows7 was intended to be a more focused, incremental upgrade to the Windows line.
3. This consistency reduces the learning curve for mastering a new application.
4. This scheme, called virtual memory, allows a PC to run more and larger programs than would otherwise be possible.
5. It is likely that the PC operating systems of 2020 will be as different from today’s Windows as the latter is from the MSDOS of the early 1980s.
6. Although UNIX has been somewhat overshadowed by its Linux progeny, a variety of open-source versions of traditional UNIX systems have become available
7. On top of this Apple built a distinctive and subtly colorful user interface called Aqua and a new version of the Macintosh Finder file and program management system.

2. Answer the following questions:

1. What architecture do Windows PCs use?
2. What does a feature "plug and play" mean?
3. What did home and "professional" versions of Windows XP replace?
4. Name the changes introduced by Windows 8.
5. Describe Windows from the user's and the programmer's points of view.
6. Name the main features of UNIX.
7. LINUX is an open-source product. What does this mean?
8. What capabilities did OS X introduce?

3. Translate into English:

Операционная система - это комплекс взаимосвязанных системных программ, назначение которого – помогать взаимодействию пользователя с компьютером и выполнению всех других программ. Операционная система выступает как связующее звено между аппаратурой компьютера и выполняемыми программами с одной стороны, а также пользователем с другой стороны.

Операционную систему можно назвать программным продолжением устройства управления компьютером.

Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой. В результате этого люди освобождаются от очень трудоемкой работы по организации взаимодействия с аппаратурой компьютера.

Для управления внешними устройствами компьютера используются специальные программы-драйверы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода, которая обычно заносится в постоянное ЗУ компьютера.

В различных моделях компьютера используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы.

Операционная система обычно хранится во внешней памяти компьютера - на диске. При включении компьютера она считывается с дисковой памяти и размещается в ОЗУ. Этот процесс называется загрузкой операционной системы. Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в операционную память и их запуск, осуществляет командный процессор операционной системы.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и КЭШа, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтеров и др.)

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- The kernel concept.
- Types of operating systems.
- Advantages and disadvantages of any modern operating system (on your choice).
- Operating systems: the reasons of gaining in popularity and fading.

INTERNET

Read the following words and word combinations and use them for understanding and translation of the text:

to route data to - направлять данные в…
physical means - физические средства
to dispatch - отправлять
wireless or satellite transmission - беспроводные или спутниковые связи
to trace to… - проследить
key military computers - основные военные компьютеры
redundancy - избыточность
impetus - толчок, импульс
to bring into consciousness - привести к осознанию
medium (pl. media) - средство, способ
ancestral- наследственный
vulnerability - уязвимость
mascot - талисман
versatile - разносторонний
flexibility - гибкость
to enhance - повысить
to spread (spread, spread) - распространяться
to be spurred by… - быть вызванным
destination – пункт назначения
sophisticated - сложный
to evolve - развиваться
ubiquity – повсеместность
to subsume – включать в какую-л. категорию
to proliferate - распространяться

The Internet is the worldwide network of all computers (or networks of computers) that communicate using a particular protocol for routing data from one computer to another. As long as the programs they run follow the rules of the protocol, the computers can be connected by a variety of physical means including ordinary and special phone lines, cable, fiber optics, and even wireless or satellite transmission.

History and Development

The Internet's origins can be traced to a project sponsored by the U.S. Defense Department. Its purpose was to find a way to connect key military computers (such as those controlling air defense radar and interceptor systems). Such a system required a great deal of redundancy, routing communications around installations that had been destroyed by enemy nuclear weapons. The solution was to break data up into individually addressed packets that could be dispatched by routing software that could find whatever route to the destination was viable or most efficient. At the destination, packets would be reassembled into messages or data files.

By the early 1970s, a number of research institutions including the pioneer networking firm Bolt Beranek and Newman (BBN), Stanford Research Institute (SRI), Carnegie-Mellon University, and the University of California at Berkeley were connected to the government-funded and administered ARPANET (named for the Defense Department's Advanced Research Projects Agency). Gradually, as use of the ARPANET's protocol spread, gateways were created to connect it to other networks such as the National Science Foundation's NSFnet. The growth of the network was also spurred by the creation of useful applications including e-mail and Usenet, a sort of bulletin-board service. Meanwhile, a completely different world of online networking arose during the 1980s in the form of local bulletin boards, often connected using a store-and-forward system called FidoNet, and proprietary online services such as CompuServe and America On-line. At first there were few connections between these networks and the ARPANET, which had evolved into a general-purpose network for the academic community under the rubric of NSFnet. (It was possible to send e-mail between some networks using special gateways, but a number of different kinds of address syntax had to be used.)

In the 1990s, the NSFnet was essentially privatized, passing from government administration to a corporation that assigned domain names. However, the impetus that brought the Internet into the daily consciousness of more and more people was the development of the World Wide Web by Tim Berners-Lee at the European particle research laboratory, CERN. With a standard way to display and link text (and the addition of graphics and multimedia by the mid-1990s), the Web is the Internet as far as most users are concerned. What had been a network for academics and adventurous professionals became a mainstream medium by the end of the decade.

Applications

A number of applications are (or have been) important contributors to the utility and popularity of the Internet.

- E-mail was one of the earliest applications on the ancestral ARPANET and remains the single most popular Internet application. Standard e-mail using SmTP (Simple mail Transfer Protocol) has been implemented for virtually every platform and operating system. In most cases once a user has entered a person's e-mail address into the "address book," e-mail can be sent with a few clicks of the mouse. While failure of the outgoing or destination mail server can still block transmission of a message, e-mail today has a high degree of reliability.
- Netnews (also called Usenet, for UNIX User Net- work) is in effect the world's largest computer bulletin board. It began in 1979, when Duke University and the University of North Carolina set up a simple mechanism for "posting" text files that could be read by other users. Today there are tens of thousands of topical "newsgroups" and millions of messages (called articles). Although still impressive in its quantity of content, many Web users now rely more on discussion forums based on Web pages.
- Ftp (File Transport Protocol) enables the transfer of one or more files between any two machines connected to the Internet. This method of file transfer has been largely supplanted by the

use of download links on Web pages, except for high-volume applications (where an ftp server is often operated "behind the scenes" of a Web link). FTP is also used by Web developers to upload files to a Web site.

- Telnet is another fundamental service that brought the Internet much of its early utility. Telnet allows a user at one computer to log into another machine and run a program there. This provided an early means for users at PCs or workstations to, for example, access the Library of Congress catalog online. However, if program and file permissions are not set properly on the "host" system, "telnet" can cause security vulnerabilities. The telnet user is also vulnerable to having IDs and passwords stolen, since these are transmitted as clear (unencrypted) text. As a result, some online sites that once supported telnet access now limit access to Web-based forms.
- Gopher was developed at the University of Minnesota and named for its mascot. Gopher is a system of servers that organize documents or other files through a hierarchy of menus that can be browsed by the remote user. Gopher became very popular in the late 1980s, only to be almost completely supplanted by the more versatile World Wide Web.
- WAIS (Wide Area Information Service) is a gateway that allows databases to be searched over the Internet. WAIS provided a relatively easy way to bring large data resources online. It, too, has largely been replaced by Web-based database services.
- The World Wide Web as mentioned above is now the main means for displaying and transferring information of all kinds over the Internet. Its flexibility, relative ease of use, and ubiquity (with Web browsers available for virtually all platforms) has caused it to subsume earlier services. The utility of the Web has been further enhanced by the development of many search engines that vary in thoroughness and sophistication.
- Streaming media protocols allow for a flow of video and/or audio content to users. Player applications for Windows and

other operating systems, and growing use of high-speed consumer Internet connections have made it possible to present "live" TV and radio shows over the Internet.

- E-commerce, having boomed in the late 1990s and in the early 2000s, continued to grow and proliferate later in the decade, finding new markets and applications and spreading into the developing world
- Blogs and other forms of online writing have become prevalent among people ranging from elementary school students to corporate CEOs.
- Social networking sites such as mySpace and Facebook are also very popular, particularly among young people.
- Wikis have become an important way to share and build on knowledge bases .

Notes:

USA Defense Department – Министерство обороны США

Interceptor system – система наведения перехватчиков

Usenet –американский кабельный телеканал, запущенный в 1971г.

Fido Net – международная мобильная компьютерная сеть, построенная по технологии «из точки в точку».

Bulletin board – электронная доска объявлений

IDs – система обнаруживания вторжений

A mainstream medium (MSM) - крупнейшие средства массовой информации

Congress catalog online – онлайн доступ к каталогу Библиотеки Конгресса США.

CEO (Chief Executive Officer) – исполнительный директор.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The Internet's origins can be traced to a project sponsored by the U.S. Defense Department.
2. The solution was to break data up into individually addressed packets that could be dispatched by routing software that could find whatever route to the destination was viable or most efficient.
3. Meanwhile, a completely different world of online networking arose during the 1980s in the form of local bulletin boards, often connected using a store-and-forward system called FidoNet.
4. In most cases once a user has entered a person's e-mail address into the "address book," e-mail can be sent with a few clicks of the mouse.
5. While failure of the outgoing or destination mail server can still block transmission of a message, e-mail today has a high degree of reliability.
6. Gopher was developed at the University of Minnesota and named for its mascot.
7. Gopher became very popular in the late 1980s, only to be almost completely supplanted by the more versatile World Wide Web.
8. As long as the programs they run follow the rules of the protocol, the computers can be connected by a variety of physical means including ordinary and special phone lines, cable, and fiber optics.
9. What had been a network for academics and adventurous professionals became a mainstream medium by the end of the decade.

2. Answer the following questions:

1. What definition can you give to the "Internet"?
2. What was the origin of the Internet?
3. What brought the Internet to the daily life of ordinary people?

4. What are the general applications of the Internet?
5. How do you see the prospects of the Internet?

3. Translate into English:

Своим появлением Интернет обязан Министерству обороны США и секретным исследованиям, которые проводились в 1969 году с целью протестировать методы сохранения компьютерных сетей при помощи динамической перемаршрутизации сообщений при военных действиях. Первой в своем роде была сеть ARPAnet, которая объединила в Калифорнии три сети по набору правил, которые были названы InternetProtocol (сокращенно IP или Интернет-протоколом).

Далее, в 1972 году доступ был открыт для исследовательских организаций и университетов, после этого сеть смогла объединить 50 исследовательских организаций и университетов, которые имели контракты с Министерством обороны США.

Затем, в 1973 году сеть смогла вырасти до международных масштабов, потому что объединяла сети, которые находились в Норвегии и Англии. 10 лет спустя IP стал расширяться при помощи набора коммуникационных протоколов, которые имели поддержку как локальных, так и глобальных сетей. Таким образом, появился на свет TCP/IP. В скором времени после этих событий, NSF (National Science Foundation) открыла NSFnet, ее целью являлось связка пяти суперкомпьютерных центров. С внедрением протокола TCP/IP сеть NSFnet в скором времени смогла заменить ARPAnet в качестве «хребта» (backbone) сети Интернета

4. Give the summary of the text using the key terms.

E-MAIL

Read the following words and word combinations and use them for understanding and translation of the text:

ubiquitous - повсеместный
accessible - доступный
routinely - обычно
enticing attachment -заманчивое приложение
mischief - вред
revenue - доход
annoyance - раздражение
impact – удар, толчок, импульс, воздействие
to eliminate – устранять, исключать
competing standard - конкурирующий стандарт
to establish – внедрять, устанавливать
security vulnerabilities – уязвимые точки для безопасности системы
extension – расширение, распространение
consumer-oriented – ориентированный на потребителя

Electronic mail is perhaps the most ubiquitous computer application in use today. E-mail can be defined as sending of a message to one or more individuals via a computer connection. The simplest form of e-mail began in the 1960s as a way that users on a time-sharing computer system could post and read messages. The messages consisted of a text in a file that was accessible to all users. A user could simply log into the e-mail system, open the file, and look for messages. In 1971, however, the ARPANET (ancestor of the Internet) was used by researchers at Bolt Beranek and Newman (BBN) to send messages from a user at one computer to a user at another. As e-mail use increased and new features were developed, the question of a standardized protocol for messages became more important. By the mid-1980s, the world of e-mail was rather fragmented, much like the situation in the early history of the telephone, where users often had to choose between two or more incompatible systems. ARPANET (or Internet) users used SmTP (Simple mail Transfer Protocol) while a competing standard (OSI mHS, or message Handling System) also had its supporters. Meanwhile, the development of consumer-oriented online services such as CompuServe and America Online threatened a further balkanization of e-mail access, though systems called gateways were developed to transport messages from one system to another. By the mid-1990s, however, the nearly universal adoption of the Internet and its TCP/IP protocol had established SmTP and the ubiquitous Sendmail mail transport program as a uniform infrastructure for e-mail. The extension of the Internet protocol to the creation of intranets has largely eliminated the use of proprietary corporate e-mail systems.

Instead, companies such as Microsoft and Google compete to offer full-featured e-mail programs that include group-oriented features such as task lists and scheduling.

The integration of e-mail with HTmL for Web-style formatting and mImE (for attaching graphics and multimedia files) has greatly increased the richness and utility of the e-mail experience. E-mail is now routinely used within organizations to distribute documents and other resources.

However, the addition of capabilities has also opened security vulnerabilities. For example, Microsoft Windows and the popular Microsoft Outlook e-mail client together provide the ability to run programs (scripts) directly from attachments (files associated with e-mail messages). This means that it is easy to create a virus program that will run when an enticing-looking attachment is opened. The virus can then find the user’s mailbox and mail copies of itself to the people found there. E-mail has thus replaced the floppy disk as the preferred medium for such mischief. Beyond security issues, e-mail is having considerable social and economic impact. E-mail has largely replaced postal mail (and even long-distance phone calls) as a way for friends and relatives to keep in touch. As more companies begin to use e-mail for providing routine bills and statements, government-run postal systems are seeing their first-class mail revenue drop considerably. Despite the risk of viruses or deception and the annoyance of electronic junk mail, e-mail has become as much a part of our way of life as the automobile and the telephone.

Notes:

OSI – Open System Interconnection - базовая эталонная модель

SmTP - Simple mail Transfer Protocol простой протокол передачи почты

Sendmail – один из старейших агентов передачи

HTmL - HyperText Markup Language - стандартный язык разметки документов во всемирной паутине

mImE - Multipurpose Internet Mail Extensions - многоцелевые расширения интернет-почты

balkanization – разделение многонациональных государств на более мелкие субъекты

Via - (лат.) через

Time-sharing –«разделение времени» - совместное использование вычислительного ресурса

Spam – an electronic junk mail

Assignments.

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. E-mail can be defined as the sending of a message to one or more individuals via a computer connection.
2. By the mid-1980s, the world of e-mail was rather fragmented, much like the situation in the early history

of the telephone, where users often had to choose between two or more incompatible systems.
3. ARPANET (or Internet) users used SmTP (Simple mail Transfer Protocol) while a competing standard (OSI mHS, or message Handling System) also had its supporters.
4. Meanwhile, the development of consumer-oriented online services such as CompuServe and America Online.
5. Despite the risk of viruses or deception and the annoyance of electronic junk mail, e-mail has become as much a part of our way of life as the automobile and the telephone.

2. Answer the following questions.

1. What was the beginning of e-mail?
2. Why did the need for standardized protocol for messages become vital?
3. What happened to the extension of the Internet protocol?
4. What are security vulnerabilities of e-mail use?
5. How can you describe the social and economic impact of e-mail?

3. Translate into English:

В современном мире каждому человеку необходимо знать, что такое e-mail и иметь свой собственный. При знакомстве с новыми людьми, на деловых переговорах или просто, встретившись со старым приятелем, вас могут попросить оставить свой e-mail. Кроме того, в самой сети на многих сайтах при регистрации требуют указывать e-mail. Итак, e-mail - это адрес электронной почты от английского "electronic mail". Если у человека в сети есть свой e-mail адрес (выглядит как набор символов, например, ivanov@yandex.ru), он может давать его другим людям точно так же, как свой домашний адрес для обычных писем. Но для входа в свою электронную почту нужен уникальный пароль, которым будете обладать лишь вы, словно ключи от квартиры. Адрес знают знакомые и друзья, а ключи (пароль) есть только у вас. То есть вашу переписку с друзьями по интернету никто кроме вас видеть не сможет, если вы не дадите никому свой пароль. По электронной почте можно передавать друг другу не только текстовые сообщения, но и картинки, фотографии, музыку, таблицы и другие файлы любого формата.

4. Give the summary of the text using the key terms.

WORLD WIDE WEB

Read the following words and word combinations and use them for understanding and translation of the text:

to retrieve - восстанавливать, извлекать

to overshadow - затмевать

to dispense - распределять, раздавать

auction - аукцион

to emerge - появляться, проявляться

remote, distant - отдаленный

highlighted links - выделенные ссылки

to handle with - обращаться с чем-то, трактовать

burgeon - росток, почка

entrepreneur - предприниматель

estimate – оценка

to improve - улучшать

profitability - прибыльность, доходность

community - сообщество, объединение

resilient - эластичный, жизнерадостный

to constrain - вынуждать, сдерживать

adaptable - легко приспособляющийся
challenge - задача, проблема, вызов
to redefine - переопределить

By the beginning of the 1990s, the Internet had become well established as a means of communication between relatively advanced computer users, particularly scientists, engineers, and computer science students—primarily using UNIX-based systems. A number of services used the Internet protocol to carry messages or data. These included e-mail, file transfer protocol and newsgroups.

A Wide Area Information Service (WAIS) even provided a protocol for users to retrieve information from databases on remote hosts. Another interesting service, Gopher, was developed at the University of Minnesota in 1991. It used a system of nested menus to organize documents at host sites so they could be browsed and retrieved by remote users. Gopher was quite popular for a few years, but it was soon overshadowed by a rather different kind of networked information service.

A physicist/programmer working at CERN, the European particle physics laboratory in Switzerland, devised in 1989 a system that he eventually called the World Wide Web (sometimes called WWW or W3). By 1990, he was running a prototype system and demonstrating it for CERN researchers and a few outside participants.

The Web consists essentially of three parts. Berners-Lee devised a markup language: that is, a system for indicating document elements (such as headers), text characteristics, and so on. Any document could be linked to another by specifying that document's unique address (called a Uniform Resource Locator or URL) in a request. Berners-Lee defined the HyperText Transport Protocol, or HTTP, to handle the details needed to retrieve documents. (Although HTTP is most often used to retrieve HTmL-formatted Web documents, it can also be used to specify documents using other protocols, such as ftp, news, or gopher.) A program responds to requests for documents sent over the network (usually the Internet, that is, TCP/IP). The requests are issued by a client program as a result of the user clicking on highlighted links or buttons or specifying addresses. The browser in turn interprets the HTmL codes on the page to display it correctly on the user's screen.

At first the Web had only text documents. However, thanks to Berners-Lee's flexible design, improved Web browsers could be created and used with the Web as long as they followed the rules for HTTP. The most successful of these new browsers was Mosaic, created by Marc Andreesen at the National Center for Supercomputing Applications. NCSA mosaic was available for free download and could run on Windows, Macintosh, and UNIX-based systems. Mosaic not only dispensed with the text commands used by most of the first browsers, but it also had the ability to display graphics and play sound files. With Mosaic the text-only hypertext of the early Web rapidly became a richer hypermedia experience. And thanks to the ability of browsers to accept modules to handle new kinds of files, the Web could also accommodate real-time sound and video transmissions.

In 1994, Andreessen left NCSA and co-founded a company called Netscape Communications, which improved and commercialized Mosaic. Microsoft soon entered with a competitor, Internet Explorer; today these two browsers dominate the market with Microsoft having taken the lead. Together with relatively low-cost Internet access these user-friendly Web browsers brought the Web (and thus the underlying Internet) to the masses. Schools and libraries began to offer Web access while workplaces began to use internal webs to organize information and organize operations.

Meanwhile, companies such as the on-line bookseller Amazon.com demonstrated new ways to deliver traditional products, while the on-line auction site eBay took advantage of the unique characteristics of the on-line medium to redefine the auction. The burgeoning Web was soon offering millions of pages, especially as entrepreneurs began to find additional business opportunities in the new medium. Two services emerged to help Web users make sense of the flood of information. Today users can search for words or phrases or browse through structured topical listings. Estimates from various sources suggest that as of 2007 approximately 1.2 billion people worldwide access the Web, with usage increasing most rapidly in the emerging industrial super- powers of India and China.

The Web is rapidly emerging as an important news medium. The medium combines the ability of broadcasting to reach many people from one point with the ability to customize content to each person's preferences. Traditional broadcasting and publishing are constrained by limited resources and the need for profitability, and thus the range and diversity of views made available tend to be limited. With the Web, anyone with a PC and a connection to a service provider can put up a Web site and say just about anything. Millions of people now display aspects of their lives and interests on their personal Web pages. The Web has also provided a fertile medium for the creation of online communities while contributing to significant issues. As the new century continues, the Web is proving itself to be truly worldwide, resilient, and adaptable to many new communications and media technologies. Nevertheless, the Web faces legal and political challenges as well as technical challenges.

Notes:

By the beginning of the 1990s – к началу тысяча девяностых

UNIX - операционная система, существующая во многих вариантах

Gopher - сетевой протокол распределенного поиска и передачи документов, широко распространенный в интернете до 1993 года.

CERN – ЦЕРН (Европейский центр ядерных исследований)

Headers - заголовки, сопровождающие сообщения для связывания ячеек между собой.
Hypermedia - главные средства массовой информации.
HTTP - HyperText Transport Protocol
URL - Uniform Resource Locator
NCSA - National Center for Supercomputing Applications

Assignments

1. Translate the sentences from the texts into Russian paying attention to the underlined words and phrases:

1. By the beginning of the 1990s, the Internet had become well established as a means of communication between relatively advanced computer users, particularly scientists, engineers, and computer science students — primarily using UNIX-based systems.
2. It used a system of nested menus to organize documents at host sites so they could be browsed and retrieved by remote users.
3. However, thanks to Berners-Lee's flexible design, improved Web browsers could be created and used with the Web as long as they followed the rules for HTTP.
4. In 1994, Andreessen left NCSA and co-founded a company called Netscape Communications, which improved and commercialized Mosaic.
5. Meanwhile, companies such as the on-line bookseller Amazon.com demonstrated new ways to deliver traditional products, while the on-line auction site eBay took advantage of the unique characteristics of the on-line medium to redefine the auction.
6. Two services emerged to help Web users make sense of the flood of information.

7. With the Web, anyone with a PC and a connection to a service provider can put up a Web site and say just about anything.
8. Nevertheless, the Web faces legal and political challenges as well as technical challenges.

2.Answer the following questions:

1. Who is the designer and founder of the World Wide Web?
2. What is the concept of WWW functioning?
3. What led to the rise of the World Wide Web?
4. What abilities does WWW offer to customers?
5. Why has the World Wide Web become the main means for transferring information over the Internet?

3. Translate into English:

История WWW (World Wide Web) Всемирной информационной паутины началась в марте 1989 года, когда Тим Бернерс-Ли предложил новый способ обмена результатами исследований и идеями между участниками коллектива исследователей-физиков, работавших в разных странах. Для передачи документов и установления связи предлагалось использовать просто систему гипертекста (тогда никто еще не задумывался о возможности передачи фотографических изображений, звука или видео; речь шла только о распространении текстовых документов, содержащих гиперссылки на фрагменты других таких же текстовых документов, но располагающихся на удаленных компьютерах, подключенных к глобальной сети Интернет).

Собственно, гипертекст не был изобретением Бернерса-Ли. Да и HTML вовсе не был первым языком описания страниц. Но в нем впервые были связаны воедино не просто отдельные главы одного документа, а документы, располагающиеся на самых различных серверах Интернета по всему земному шару.

4. Give the summary of the text using the key terms.

WEB BROWSER

Read the following words and word combinations and use them for understanding and translation of the text:

to resolve to – принимать решение
relevant – относящийся к делу
requested - предлагаемый, запрашиваемый
cache - кэш, тайный запас
to achieve- достигать
to be available for - доступный для
attractive - привлекательный
to reduce - сокращать
fray- схватка,битва
to be inferior to - уступать, быть хуже (ниже) по сравнению с
sufficiently - достаточно
to bundle - связывать в узел, отсылать
to fetch - извлекать, вызывать
various - разнообразный
controversial - противоречивый
to embed - встроить, внедрить.
by default - по умолчанию
layout – планирование, разметка.
to intertwine with –переплетаться с…
triggered by - приводимый в действие
to quit –уходить, покидать
flagship - флагман
to ruin - разорять
litigation-судебное дело

The World Wide Web consists of millions of sites that provide hyper-text documents that can include not only text but still images, video, and sound. To access these pages, the user runs a Web-browsing program. The basic function of a Web browser is to request a page by specifying its address (URL, uniform [or universal] resource locator). This request resolves to a request (HTTP, HyperText Transport Protocol) that is processed by the relevant Web server. The server sends the HTML document to the browser, which then displays it for the user. Typically, the browser stores recently requested documents and files in a local cache on the user's PCs. Use of the cache reduces the amount of data that must be resent over the Internet. However, sufficiently skilled snoopers can examine the cache to find details of a user's recent Web surfing. (Caching is also used by Internet Service Providers so they can provide frequently requested pages from their own server rather than having to fetch them from the hosting sites.)

When the Web was first created in the early 1990s it consisted only of text pages, although there were a few experimental graphical Web extensions developed by various researchers. The first graphical Web browser to achieve widespread use was Mosaic created by Marc Andreessen, developed at the National Center for Supercomputing Applications (NCSA). By 1993, Mosaic was available for free download and had become the browser of choice for PC users. Andreessen left NCSA in 1994 to found Netscape Corporation. The Netscape Navigator browser improved Mosaic in several ways, making the graphics faster and more attractive. Netscape included a facility called Secure Sockets Layer (SSL) for carrying out encrypted commercial transactions on-line. Microsoft, which had been a latecomer to the Internet boom, entered the fray with its Microsoft Internet Explorer. At first the program was inferior to Netscape, but it was steadily improved. Aided by Microsoft's controversial tactic of bundling the free browser starting with Windows 95, Internet Explorer has taken over the leading browser position.

A Web browser such as Microsoft Internet Explorer or Firefox makes it easy to find and move between linked Web pages. Browser users can record or “bookmark” favorite pages. Browser plug-ins provide support for services such as streaming video and audio.

Web browsers communicate with web servers primarily using HTTP (Hyper-text Transfer Protocol) to fetch web pages. HTTP allows web browsers to submit information to web servers as well as fetch web pages from them. As of writing, the most commonly used HTTP is HTTP/1.1, which is fully defined in RFC 2616. HTTP/1.1 has its own required standards which Internet Explorer does not fully support, but most other current-generation web browsers do.

Pages are located by means of a URL (Uniform Resource Locator), which is treated as an address, beginning with http: for HTTP access. Many browsers also support a variety of other URL types and their corresponding protocols, such as ftp: for FTP (file transfer protocol), gopher: for Gopher, and https: for HTTPS (an SSL encrypted version of HTTP).

The file format for a web page is usually HTML (hyper-text markup language) and is identified in the HTTP protocol using a MIME content type. Most browsers support a variety of formats in addition to HTML, such as the JPEG, PNG and GIF image formats, and can be extended to support more through the use of plug-ins. The combination of HTTP content type and URL protocol specification allows web page designers to embed images, animations, video, sound, and streaming media into a web page, or to make them accessible through the web page.

Early web browsers supported only a very simple version of HTML. The rapid development of proprietary web browsers led to the development of non-standard dialects of HTML, leading to problems with Web interoperability. Modern web browsers (Mozilla, Opera, and Safari) support standards-based HTML and XHTML (starting with HTML 4.01), which should display in the same way across all browsers. Internet Explorer does not fully support XHTML 1.0 or 1.1 yet. Currently many sites are designed using WYSIWYG HTML generation programs such as Macromedia Dreamweaver or Microsoft Frontpage. These often generate non-standard HTML by default, hindering the work of the W3C in developing standards, specifically with XHTML and CSS (cascading style sheets, used for page layout).

Some of the more popular browsers include additional components to support Usenet news, IRC (Internet relay chat), and e-mail. Protocols supported may include NNTP (network news transfer protocol), SmTP (Simple mail Transfer Protocol), IMAP (Internet Message Access Protocol), and POP (Post Office Protocol).

Brief history

Tim Berners-Lee, who pioneered the use of hypertext for sharing information, created the first web browser, named the WorldWideWeb, in 1990, and introduced it to colleagues at CERN in March 1991. Since then the development of web browsers has been inseparably intertwined with the development of the web itself.

The explosion in popularity of the web was triggered by NCSA Mosaic which was a graphical browser running originally on UNIX but soon ported to the Apple Macintosh and Microsoft Windows platforms. Version 1.0 was released in September 1993. Marc Andreessen, who was the leader of the Mosaic team at NCSA, quitted forming a company that would later become known as Netscape Communications Corporation.

Netscape released its flagship Navigator product in October 1994, and it took off the next year. Microsoft, which had so far missed the Internet wave, now entered the fray with its Internet Explorer product, hastily purchased from Spyglass Inc. This began the browser wars, the fight for the web browser market between the software giant Microsoft and the start-up company largely responsible for popularizing the World Wide Web, Netscape.

The wars put the web in the hands of millions of ordinary PC users, but showed how commercialization of the internet could ruin standards efforts. Both Microsoft and Netscape liberally incorporated proprietary extensions to HTML in their products, and tried to gain an edge by product differentiation. The wars ended in 1998 when it became clear that Netscape's declining market share trend was irreversible. This was in part due to Microsoft's integrating its browser with its operating system and bundling deals with OEMs; the company faced antitrust litigation on these charges.

Netscape responded by open sourcing its product, creating Mozilla. This did nothing to slow Netscape's declining market share. The company was purchased by America Online in late 1998. Mozilla has since evolved into a stable and powerful browser suite with a small but steady market share.

Opera, a speedy browser popular in handheld devices and in some countries was released in 1996 and remains a niche player in the PC web browser market.

The Lynx browser remains popular in the Linux market and with vision impaired users due to its entirely text-based nature. There are also several text-mode browsers with advanced features, such as links and its forks.

While the Macintosh scene too has traditionally been dominated by Internet Explorer and Netscape, the future appears to belong to Apple's Safari which is based on the KHTML rendering engine of the open source conqueror browser.

In 2003, Microsoft announced that Internet Explorer would no longer be made available as a separate product but would be part of the evolution of its Windows platform.

Web and web browser features

Different browsers can be distinguished from each other by the features they support. Modern browsers and web pages tend to utilize many features and techniques that did not exist in the early days of the web. As noted earlier, with the browser wars there was a rapid and chaotic expansion of browser and World Wide Web feature sets.

Notes:

Hyper Text Transport Protocol (HTML) - «протокол передачи гипертекста»

Secure Sockets Layer (SSL) – криптографический протокол, обеспечивающий безопасность связи.

Netscape – браузер, шестой в мире по популярности.

Streaming media - потоковый метод для передачи данных.

Web browser - программное обеспечение для извлечения и прохождения информационных ресурсов в World Wide Web.

Interoperability – интероперабельность, способность к взаимодействию.

Front page - редактор, входящий в состав пакета приложений Microsoft Office.

Macromedia - редактор компании Adobe.

Dreamweaver - один из крупнейших производителей программ, связанных с WEB.

CERN – Европейский центр ядерных исследований

Mozilla - браузер нового поколения

Lynx browser - один из первых текстовых браузеров

Open Source Browser - браузер, не посылающий идентифицирующей информации разработчикам

Antitrust litigation - судебный процесс по антитрестовскому делу

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The World Wide Web consists of millions of sites that provide hyper-text documents that can include not only text but still images, video, and sound.
2. However, sufficiently skilled snoopers can examine the cache to find details of a user’s recent Web surfing
3. Caching is also used by Internet Service Providers so they can provide frequently requested pages from their own server rather than having to fetch them from the hosting sites.
4. Netscape included a facility called Secure Sockets Layer (SSL) for carrying out encrypted commercial transactions on-line.
5. Modern web browsers (Mozilla, Opera, and Safari) support standards-based HTML and XHTML (starting with HTML 4.01), which should display in the same way across all browsers. Internet Explorer does not fully support XHTML 1.0 or 1.1 yet.
6. Marc Andreessen, who was the leader of the Mosaic team at NCSA, quitted forming a company that would later become known as Netscape Communications Corporation.
7. Netscape released its flagship Navigator product in October 1994, and it took off the next year. Microsoft, which had so far missed the Internet wave, now entered the fray with its Internet Explorer product.
8. Both Microsoft and Netscape liberally incorporated proprietary extensions to HTML in their products, and tried to gain an edge by product differentiation.
9. This was in part due to Microsoft's integrating its browser with its operating system and bundling deals with OEMs; the company on faced antitrust litigation on these charges.
10. Different browsers can be distinguished from each other by the features they support.

2. Answer the following questions.

1. What is the main function of Web Browser?
2. What is the file format for a web page?
3. What standards do the modern browsers support?
4. When was the first web browser created?

3. Translate into English:

В декабре 1992-го Марк Андрессен вместе с Эриком Бина задумали написать свою собственную программу-клиент для просмотра гипертекстовых WWW-документов, и всего через три месяца на свет появилась Mosaic мощная графическая интерфейсная программа для работы с Всемирной информационной паутиной и другими ресурсами Интернета.

С этого момента пользователь уже мог не вдаваться в технические подробности реализации протоколов компьютерных сетей, а, используя очень простой и удобный инструмент, посвятить всего себя работе с интересующей его информацией, распределенной по огромному множеству компьютеров, расположенных в разных городах, странах и даже на разных континентах.

Mosaic в кратчайший срок стала самой быстро распространяющейся по миру программой. Таким образом, недавний выпускник колледжа Марк Андрессен добавил последний штрих, позволивший превратить всю Сеть, насчитывавшую в то время от одного до двух миллионов компьютеров, в единый гигантский суперисточник информации.

Чуть позже Андрессен вместе с Джимом Кларком основали в Калифорнии компанию Netscape Communications, на долгие годы ставшей законодателем мод в Интернете.

4. Give the summary of the text using the key terms.

WEB PAGE DESIGN

Read the following words and word combinations and use them for understanding and translation of the text:

to emerge - появляться
to involve, to include - включать в себя
to stream audio and video - передавать аудио и видео
distinctive - отличительный
cluttered - запутанный, с помехами
to make sure - убедиться
background - фон
to elaborate - разработать
to separate - разделять
emphasis - акцент
permission - разрешение
to grab a restrained style - выбрать сдержанный стиль
to abound - быть в большом количестве
purpose –цель
to raise performance - поднять производительность, повысить эффективость
consistent - последовательный
bizarre typefaces - причудливые шрифты
to extend - расширять
to encounter - сталкиваться, встречать
to be intended for - быть предназначеным для
to implement - внедрять
speedy access - скоростной доступ
to undergo - подвергаться,испытывать
to grab users' attention - привлекать внимание пользователей

The World Wide Web has existed for fewer than two decades, so it is not surprising that the principles and practices for the design of attractive and effective Web pages are still emerging. Creating Web pages involves many skills. In addition to the basic art of writing, many skills that had belonged to separate professions in the print world now often must be exercised by the same individual. These include typography (the selection and use of type and type styles), composition (the arrangement of text on the page), and graphics. To this mix must be added nontraditional skills such as designing interactive features and forms, interfacing with other facilities (such as databases), and perhaps the incorporation of features such as animation or streaming audio or video.

However new the technology, the design process still begins with the traditional questions any writer must ask: What is the purpose of this work? Who am I writing for? What are the needs of this audience? A Web site that is designed to provide background information and contact for a university department is likely to have a printlike format and a restrained style.

Nevertheless, the designer of such a site may be able to imaginatively extend it beyond the traditional bounds—for example, by including streaming video interviews that introduce faculty members.

A site for an online store is likely to have more graphics and other attention-getting features than an academic or government site. However, despite the pressure to "grab eyeballs," the designer must resist making the site so cluttered with animations, pop-up windows, and other features that it becomes hard for readers to search for and read about the products they want.

A site intended for an organization's own use should not be visually unattractive, but the emphasis is not on grabbing users' attention, since the users are already committed to using the system. Rather, the emphasis will be on providing speedy access to the information people need to do their job, and in keeping information accurate and up to date. Once the general approach is settled on, the design must be implemented.

The most basic tool is HTmL, which has undergone periodic revisions and expansions. Even on today's large, high-resolution monitors a screen of text is not the same as a page in a printed book or magazine. There are many ways text can be organized.

A page that is presenting a manual or other lengthy document can mimic a printed book by having a table of contents. Clicking on a chapter takes the reader there. Shorter presentations (such as product descriptions) might be shown in a frame with buttons for the reader to select different aspects such as features and pricing. Frames (independently scrollable regions on a page) can turn a page into a "window" into many kinds of information without the user having to navigate from page to page, but there can be browser compatibility issues. Tables are another important tool for page designers. Setting up a table and inserting text into it allows pages to be formatted automatically.

Many sites include several different navigation systems including buttons, links, and perhaps menus. This can be good if it provides different types of access to serve different needs, but the most common failing in Web design is probably the tendency to clutter pages with features to the point that they are confusing and actually harder to use.

Although the Web is a new medium, much of the traditional typographic wisdom still applies. Just as many people who first encountered the variety of Windows or Macintosh fonts in the 1980s filled their documents with a variety of often bizarre typefaces, beginning Web page designers sometimes choose fonts that they think are "edgy" or cool, but may be hard to read—especially when shown against a purple background!

Today it is quite possible to create attractive Web pages without extensive knowledge of HTmL. Programs such as FrontPage and DreamWeaver mimic the operation of a word processor and take a WYSIWYg (what you see is what you get) approach. Users can build pages by selecting and arranging structural elements, while choosing styles for headers and other text as in a word processor. These programs also provide "themes" that help keep the visual and textual elements of the page consistent. Of course, designing pages in this way can be criticized as leading to a “canned” product. People who want more distinctive pages may choose instead to learn the necessary skills or hire a professional Web page designer.

A feature called Cascading Style Sheets (CSS) allows designers to precisely control the appearance of Web pages while defining consistent styles for elements such as headings and different types of text. Most Web pages include graphics, and this raises an additional set of issues.

Page designers must also make sure that the graphics they are using are created in-house, are public domain, or are used by permission. Animated graphics can raise performance and compatibility issues. Generally, if a site offers, for example, Flash animations, it also offers users an alternative presentation to accommodate those with slower connections or without the necessary browser plug-ins. The line between Web page design and other Web services continues to blur as more forms of media are carried online. Web designers need to learn about such media technologies and find appropriate ways to integrate them into their pages. Web pages may also need to provide or link to new types of forums.

Since the start of the 21st century the Web has become more and more integrated into people`s lives, as this has happened the technology of the Web has also moved on. There have also been significant changes in the way people use and access the Web, and this has changed how sites are designed.

Since the end of the browsers wars there have been new browsers coming onto the scene. Many of these are open source meaning that they tend to have faster development and are more supportive of new standards.

Notes:

HTmL - от англ. HyperText Markup Language — «язык гипертекстовой разметки»;) — стандартный язык разметки документов во Всемирной паутине.

Cascading Style Sheets (CSS) - Каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
FrontPage - программа для создания сайта и Web-страниц.
DreamWeaver - HTML-редактор компании Adobe

Assignments:

1. Translate the sentences from the text into Russian paying attention to the underlined words and phrases:

1. In addition to the basic art of writing, many skills that had belonged to separate professions in the print world now often must be exercised by the same individual.
2. The most basic tool is HTmL, which has undergone periodic revisions and expansions.
3. Many sites include several different navigation systems including buttons, links, and perhaps menus.
4. The emphasis will be on providing speedy access to the information people need to do their job, and in keeping information accurate and up to date.
5. These programs also provide "themes" that help keep the visual and textual elements of the page consistent.
6. Page designers must also make sure that the graphics they are using are created in-house, are public domain, or are used by permission.
7. Although the Web is a new medium, much of the traditional typographic wisdom still applies. Just as many people who first encountered the variety of Windows or Macintosh fonts in the 1980s filled their documents with a variety of often bizarre typefaces, beginning Web page designers sometimes choose fonts that they think are "edgy" or cool, but may be hard to read—especially when shown against a purple background!

2. Answer the following questions.

1. What are the basic skills for designing web pages?
2. What can help to create with web design?
3. What should technologies web designers know?
4. What can be said about the line between Web page design and other web services?

3. Translate into English:

Web-дизайн – это не только внешнее оформление сайта. На нем должны быть грамотно расположены элементы, текст должен легко читаться, и самое главное – он должен быть удобным для пользователя. Размещение информации на сайте должно быть четко продумано и оформлено соответствующим образом. Веб-дизайнер должен уметь не только хорошо рисовать и правильно подбирать цветовую гамму, но и смотреть на свой шедевр глазами посетителя.

Разработка веб-приложений — это общий термин для процесса создания веб-страниц или сайтов. Веб-страницы создаются с использованием HTML, CSS и JavaScript. Эти страницы могут содержать простой текст и графику, напоминая собой статичный документ. Страницы также могут быть интерактивными или отображать меняющуюся информацию. Создавать интерактивные страницы немного сложнее, но они позволяют создавать веб-сайты с богатым содержимым. Сегодня большинство страниц интерактивны и предоставляют современные интерактивные услуги, такие как корзины интернет-магазинов, динамическая визуализация и даже сложные социальные сети.

4. Give the summary of the text using the key terms.

INTERNET PROTOCOLS

Read the following words and word combinations and use them for understanding and translation of the text:

to communicate across - общаться через
scope - сфера, уровень
to facilitate - облегчать
corresponding - соответствующий
dissimilar - непохожие,различные
to span - охватывать
goal - цель
to consist of - состоять из
flow - поток
reference model - эталонная модель
to overflow internal buffers – переполнить буферы обмена
benefits - преимущества
full-duplex operation - спаренный режим работы
to be acknowledged - быть признанным

The Internet protocols are the world's most popular open-system (nonproprietary) protocol suite because they can be used to communicate across any set of interconnected networks and are equally well suited for LAN and WAN communications. The Internet protocols consist of a suite of communication protocols, of which the two best known are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The Internet protocol suite not only includes lower-layer protocols (such as TCP and IP), but it also specifies common applications such as electronic mail, terminal emulation, and file transfer.

Internet protocols were first developed in the mid-1970s, when the Defense Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network that would facilitate communication between dissimilar computer systems at research institutions. With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN). The result of this development effort was the Internet protocol suite, completed in the late 1970s.

TCP/IP later was included with Berkeley Software Distribution (BSD) UNIX and has since become the foundation on which the Internet and the World Wide Web (WWW) are based.

Documentation of the Internet protocols (including new or revised protocols) and policies are specified in technical reports called Request For Comments (RFCs), which are published and then reviewed and analyzed by the Internet community. Protocol refinements are published in the new RFCs. Internet protocols span the complete range of OSI model layers maps, many of the protocols of the Internet protocol suite and their corresponding OSI layers.

Internet Protocol (IP)

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.

Transmission Control Protocol (TCP)

The TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the OSI reference model. Among the services TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.

With stream data transfer, TCP delivers an unstructured stream of bytes identified by sequence numbers. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.

TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork. It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate, or misread packets. A time-out mechanism allows devices to detect lost packets and request retransmission.

TCP offers efficient flow control, which means that, when sending acknowledgments back to the source, the receiving TCP process indicates the highest sequence number it can receive without overflowing its internal buffers. Full-duplex operation means that TCP processes can both send and receive at the same time.

Finally, TCP's multiplexing means that numerous simultaneous upper-layer conversations can be multiplexed over a single connection.

Notes:

LAN - Локальная вычислительная сеть (ЛВС, локальная сеть, сленг. локалка; англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт)

WAN - Wide Area Network (WAN) - Глобальная вычислительная сеть охватывает целые области, страны и даже континенты.

TCP - Transmission Control Protocol (TCP) (протокол управления передачей) — один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

IP - Internet Protocol (IP, досл. «межсетевой протокол») — маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет.

OSI - Сетевая модель OSI (англ. open systems interconnection basic reference model) — базовая эталонная модель взаимодействия открытых систем

DARPA (Defense Advanced Research Projects Agency) - Управление передовых оборонных исследовательских проектов Министерства обороны США, целью которого является сохранение технологического превосходства вооруженных сил США, предотвращение внезапного для США появления новых технических средств вооруженной борьбы, поддержка прорывных исследований, преодоление разрыва между фундаментальными исследованиями и их применением в военной сфере.

Assignments

1. Translate the sentences from the text into Russian paying in writing paying attention to the underlined words and phrases:

1. The Internet protocols consist of a suite of communication protocols, of which the two best known are the Transmission Control Protocol (TCP) and the Internet Protocol (IP).
2. The Internet protocol suite not only includes lower-layer protocols (such as TCP and IP), but it also specifies common applications such as electronic mail, terminal emulation, and file transfer.

3. With stream data transfer, TCP delivers an unstructured stream of bytes identified by sequence numbers.
4. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.
5. TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork.
6. It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted.

2. Answer the following questions.

1. When were the Internet protocols used for the first time?
2. What does the Internet Protocol contain?
3. What are the major responsibilities of IP?
4. What services does TCP provide?
5. What is the general use of the Internet Protocols?

3. Translate into English:

Очевидно, что рано или поздно компьютеры, расположенные в разных точках земного шара, по мере увеличения своего количества должны были обрести некие средства общения. Такими средствами стали компьютерные сети. Сети бывают локальными и глобальными. Локальная сеть - это сеть, объединяющая компьютеры, географически расположенные на небольшом расстоянии друг от друга - например, в одном здании. Глобальные сети служат для соединения сетей и компьютеров, которых разделяют большие расстояния - в сотни и тысячи километров. Интернет относится к классу глобальных сетей.

Простое подключение одного компьютера к другому - шаг, необходимый для создания сети, но не достаточный. Чтобы начать передавать информацию, нужно убедиться, что компьютеры "понимают" друг друга. Как же компьютеры "общаются" по сети? Чтобы обеспечить эту возможность, были разработаны специальные средства, получившие название "протоколы". Протокол - это совокупность правил, в соответствии с которыми происходит передача информации через сеть. Понятие протокола применимо не только к компьютерной индустрии. Даже те, кто никогда не имел дела с Интернетом, скорее всего работали в повседневной жизни с какими-либо устройствами, функционирование которых основано на использовании протоколов. Так, обычная телефонная сеть общего пользования тоже имеет свой протокол, который позволяет аппаратам, например, устанавливать факт снятия трубки на другом конце линии или распознавать сигнал о разъединении и даже номер звонящего.

Исходя из этой естественной необходимости, миру компьютеров потребовался единый язык (то есть протокол), который был бы понятен каждому из них.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- The origins of the Internet
- World Wide Web and its pioneers
- Browser Wars

COMPUTER SECURITY AND RISKS

BASIC COMPUTER SECURITY CONCEPTS

Read the following words and word combinations and use them for understanding and translation of the text:

confidentiality - конфиденциальность
integrity - целостность
availability - доступность
to pertain to - иметь отношение к...
authentication - аутентификация, подлинность
authorization - авторизация
nonrepudiation - реотрицаемость, строгое выполнение обязательств
to corrupt – повреждать, искажать
tampering - взлом
password - пароль
to refute - опровергать
trustworthy - заслуживающий доверия
intruder - злоумышленник
weak link - слабое звено
innocuous - безвредный
break-in - проникновение в систему
to compromise - раскрывать
denial-of-service - отказ в обслуживании

Computer security (also known as **cybersecurity** or **IT security**) is information security as applied to computing devices such as computers and smartphones, as well as computer networks such as private and public networks, including the Internet as a whole.

There are many characterizations of computer security. Information technology security is defined in a document created by the European Community, which has gained some recent international acceptance. The document defines information technology (IT) security to include the following:

- Confidentiality. Prevention of unauthorized disclosure of information.
- Integrity. Prevention of unauthorized modification of information.
- Availability. Prevention of unauthorized withholding of information or resources.

Availability pertains to both information and resources, such as computer systems themselves. Confidentiality and integrity pertain only to information itself. Concepts relating to the people who use that information are authentication, authorization, and nonrepudiation.

When information is read or copied by someone not authorized to do so, the result is known as loss of **confidentiality**. For some types of information, confidentiality is a very important attribute. Examples include research data, medical and insurance records, new product specifications, and corporate investment strategies. In some locations, there may be a legal obligation to protect the privacy of individuals. This is particularly true for banks and loan companies; debt collectors; businesses that extend credit to their customers or issue credit cards; hospitals, doctors' offices, and medical testing laboratories; individuals or agencies that offer services such as psychological counseling or drug treatment; and agencies that collect taxes.

Information can be corrupted when it is available on an insecure network. When information is modified in unexpected ways, the result is known as loss of **integrity**. This means that unauthorized changes are made to information, whether by human error or intentional tampering. Integrity is particularly important for critical safety and financial data used for activities such as electronic funds transfers, air traffic control, and financial accounting.

Information can be erased or become inaccessible, resulting in loss of **availability**. This means that people who are authorized to get information cannot get what they need. Availability is often the most important attribute in service-oriented businesses that depend on information (for example, airline schedules and online inventory systems).

Availability of the network itself is important to anyone whose business or education relies on a network connection. When users cannot access the network or specific services provided on the network, they experience a denial-of-service.

To make information available to those who need it and who can be trusted with it, organizations use **authentication and authorization**. Authentication is proving that a user is the person he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user that proves the person's identity (such as a fingerprint). Authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program.

Authentication and authorization go hand in hand. Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted — the user cannot later deny that he or she performed the activity. This is known as **nonrepudiation**.

The Internet users want to be assured that

- they can trust the information they use
- the information they are responsible for will be shared only in the manner that they expect
- the information will be available when they need it
- the systems they use will process information in a timely and trustworthy manner

It is remarkably easy to gain unauthorized access to information in an insecure networked environment, and it is hard to catch the intruders. Even if users have nothing stored on their computer that they consider important, that computer can be a "weak link," allowing unauthorized access to the organization's systems and information.

Seemingly innocuous information can expose a computer system to compromise. Information that intruders find useful includes which hardware and software are being used, system configuration, type of network connections, phone numbers, and access and authentication procedures. Security-related information can enable unauthorized individuals to access important files and programs, thus compromising the security of the system. Examples of important information are passwords, access control files and keys, personnel information, and encryption algorithms.

The consequences of a break-in cover a broad range of possibilities: a minor loss of time in recovering from the problem, a decrease in productivity, a significant loss of money or staff-hours, a devastating loss of credibility or market opportunity, a business no longer able to compete, legal liability, and the loss of life. Individuals may find that their credit card, medical, and other private information has been compromised.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. Information can be corrupted when it is available on an insecure network.
2. When information is modified in unexpected ways, the result is known as loss of integrity. This means that unauthorized changes are made to information, whether by human error or intentional tampering**.**
3. Availability is often the most important attribute in service-oriented businesses that depend on information.

4. When users cannot access the network or specific services provided on the network, they experience a denial of service.
5. Security is strong when the means of authentication cannot later be refuted - the user cannot later deny that he or she performed the activity.
6. It is remarkably easy to gain unauthorized access to information in an insecure networked environment, and it is hard to catch the intruders.

2. Answer the following questions:

1. In what spheres of human activity does availability play an essential role? Why?
2. When will security be the strongest?
3. Which concept is the most remarkable for the provision of overall security?
4. What can be the consequences of an unauthorized break-in?
5. How can intruders benefit from the access to innocuous information?
6. Is there a principal difference between authorization and authentication?

3. Translate into English:

Под информационной безопасностью понимается защищенность информации и поддерживающей ее инфраструктуры от любых случайных или злонамеренных воздействий, результатом которых может явиться нанесение ущерба самой информации, ее владельцам или поддерживающей инфраструктуре.

Цель информационной безопасности - обезопасить ценности системы, защитить и гарантировать точность и целостность информации и минимизировать разрушения, которые могут иметь место, если информация будет модифицирована или разрушена.

На практике важнейшими являются три аспекта информационной безопасности:

1. Доступность (возможность за разумное время получить требуемую информационную услугу);
2. Целостность (ее защищенность от разрушения и несанкционированного изменения);
3. Конфиденциальность (защита от несанкционированного прочтения).

4. Give the summary of the text using the key terms.

TYPES OF INCIDENTS

Read the following words and word combinations and use them for understanding and translation of the text:

threat	**угроза**
violation	**нарушение**
disruption	**прерывание**
usurpation	**захват, неправомерное присвоение**
ubiquitous	**повсеместный**
snooping	**перехват**
wiretapping	**перехват при подключении к линии связи**
man-in-the-middle attack	**атака через посредника**
recipient	**получатель**
intermediary	**посредник**
masquerading	**выдача себя за другое лицо**
spoofing	**подмена**
to lure	**заманивать**
repudiation	**опровержение**
pending	**отложенный**
probe	**зонд**

packet sniffer — перехватчик пакетов
downtime — нерабочее время, простой

Threats

A threat is a potential violation of security. The violation need not actually occur for there to be a threat. The fact that the violation might occur means that those actions that could cause it to occur must be guarded against (or prepared for). Those actions are called attacks.

Threats can be divided into four broad classes: disclosure, or unauthorized access to information; deception, or acceptance of false data; disruption, or interruption or prevention of correct operation; and usurpation, or unauthorized control of some part of a system. These four broad classes encompass many common threats. Since the threats are ubiquitous, an introductory discussion of each one will present issues that recur throughout the study of computer security.

Snooping, the unauthorized interception of information, is a form of disclosure. It is passive, suggesting simply that some entity is listening to (or reading) communications or browsing through files or system information. Wiretapping, or passive wiretapping, is a form of snooping in which a network is monitored.

Modification or alteration, an unauthorized change of information. Active wiretapping is a form of modification in which data moving across a network is altered; the term "active" distinguishes it from snooping ("passive" wiretapping). An example is the man-in-the-middle attack, in which an intruder reads messages from the sender and sends (possibly modified) versions to the recipient, in hopes that the recipient and sender will not realize the presence of the intermediary.

Masquerading or spoofing, an impersonation of one entity by another. It lures a victim into believing that the entity with which it is communicating is a different entity. This may be a passive attack (in which the user does not attempt to authenticate the recipient, but merely accesses it), but it is usually an active attack (in which the masquerader issues responses to mislead the user about its identity). It is often used to usurp control of a system by an attacker impersonating an authorized manager or controller.

Repudiation of origin, a false denial that an entity sent (or created) something. Suppose a customer sends a letter to a vendor agreeing to pay a large amount of money for a product. The vendor ships the product and then demands payment. The customer denies having ordered the product and by law is therefore entitled to keep the unsolicited shipment without payment. The customer has repudiated the origin of the letter. If the vendor cannot prove that the letter came from the customer, the attack succeeds.

Delay, a temporary inhibition of a service. Typically, delivery of a message or service requires some time *t*; if an attacker can force the delivery to take more than time *t*, the attacker has successfully delayed delivery. This requires manipulation of system control structures, such as network components or server components, and hence is a form of usurpation.

Denial-of-service

The goal of denial-of-service attacks is not to gain unauthorized access to machines or data, but to prevent legitimate users of a service from using it. A denial-of-service attack can come in many forms. Attackers may "flood" a network with large volumes of data or deliberately consume a scarce or limited resource, such as process control blocks or pending network connections. They may also disrupt physical components of the network or manipulate data in transit, including encrypted data.

Attacks

An attempt to breach system security may not be deliberate; it may be the product of environmental characteristics rather than specific actions of an attacker. Incidents can be broadly classified into several kinds: the probe, scan, account compromise, root compromise, packet sniffer, denial of service, exploitation of trust, malicious code, and Internet infrastructure attacks.

Probe

A probe is characterized by unusual attempts to gain access to a system or to discover information about the system. Probing is the electronic equivalent of testing doorknobs to find an unlocked door for easy entry. Probes are sometimes followed by a more serious security event, but they are often the result of curiosity or confusion.

Scan

A scan is simply a large number of probes done using an automated tool. Scans can sometimes be the result of a misconfiguration or other error, but they are often a prelude to a more directed attack on systems that the intruder has found to be vulnerable.

Account compromise

An account compromise is the unauthorized use of a computer account by someone other than the account owner, without involving system-level or root-level privileges (privileges a system administrator or network manager has). An account compromise might expose the victim to serious data loss, data theft, or theft of services.

Root compromise

A root compromise is similar to an account compromise, except that the account that has been compromised has special privileges on the system. Intruders who succeed in a root compromise can do just about anything on the victim's system, including run their own programs, change how the system works, and hide traces of their intrusion.

Packet sniffer

A packet sniffer is a program that captures data from information packets as they travel over the network. That data may include user names, passwords, and proprietary information that travels over the network in clear text. With perhaps hundreds or thousands of passwords captured by the sniffer, intruders can launch widespread attacks on systems.

Exploitation of trust

Computers on networks often have trust relationships with one another. For example, before executing some commands, the computer checks a set of files that specify which other computers on the network are permitted to use those commands. If attackers can forge their identity, appearing to be using the trusted computer, they may be able to gain unauthorized access to other computers.

Malicious code

Malicious code is a general term for programs that, when executed, would cause undesired results on a system. Users of the system usually are not aware of the program until they discover the damage. Malicious code includes Trojan horses, viruses, and worms. Trojan horses and viruses are usually hidden in legitimate programs or files that attackers have altered to do more than what is expected. Worms are self-replicating programs that spread with no human intervention after they are started. Viruses are also self-replicating programs, but usually require some action on the part of the user to spread inadvertently to other programs or systems. These sorts of programs can lead to serious data loss, downtime, denial-of-service, and other types of security incidents.

Internet infrastructure attacks

These rare but serious attacks involve key components of the Internet infrastructure rather than specific systems on the Internet. Examples are network name servers, network access providers, and large archive sites on which many users depend. Widespread automated attacks can also threaten the infrastructure. Infrastructure attacks affect a large portion of the Internet and can seriously hinder the day-to-day operation of many sites.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The fact that the violation might occur means that those actions that could cause it to occur must be guarded against (or prepared for).
2. Threats can be divided into four broad classes: disclosure, or unauthorized access to information; deception, or acceptance of false data; disruption, or interruption or prevention of correct operation; and usurpation, or unauthorized control of some part of a system.
3. Active wiretapping is a form of modification in which data moving across a network is altered; the term "active" distinguishes it from snooping ("passive" wiretapping).
4. Masquerading or spoofing is often used to usurp control of a system by an attacker impersonating an authorized manager or controller.
5. Attackers may "flood" a network with large volumes of data or deliberately consume a scarce or limited resource, such as process control blocks or pending network connections.
6. Scans can sometimes be the result of a misconfiguration or other error, but they are often a prelude to a more directed attack on systems that the intruder has found to be vulnerable.
7. An account compromise is the unauthorized use of a computer account by someone other than the account owner, without involving system-level or root-level privileges (privileges a system administrator or network manager has).

8. A packet sniffer is a program that captures data from information packets as they travel over the network. That data may include user names, passwords, and proprietary information that travels over the network in clear text.

2. Answer the following questions:

1. What is the principal difference between threats and attacks?
2. What are the four classes that encompass common threats?
3. What is the difference between passive and active wiretapping?
4. How do various types of denial-of-service attacks work?
5. Can the breaches of the system security be unintentional?
6. What are the consequences of a malicious code execution?

3. Translate into English:

Для подготовки и проведения атак могут использоваться либо специально разработанные для этих целей программные средства, либо легальные программы «мирного» назначения. Так, последний пример показывает, как легальная программа ping, которая создавалась в качестве инструмента диагностики сети, может быть применена для подготовки атаки. При проведении атак злоумышленнику важно не только добиться своей цели, заключающейся в причинении ущерба атакуемому объекту, но и уничтожить все следы своего участия в этом. Одним из основных приемов, используемых злоумышленниками для «заметания следов», является подмена содержимого пакетов (spoofing). В частности, для сокрытия места нахождения источника вредительских пакетов (например, при атаке отказа в обслуживании) злоумышленник изменяет значение поля адреса отправителя в заголовках пакетов. Поскольку адрес отправителя генерируется автоматически системным программным обеспечением, злоумышленник вносит изменения в соответствующие программные модули так, чтобы они давали ему возможность отправлять со своего компьютера пакеты с любыми IP-адресами.

4. Give the summary of the text using the key terms.

IMPROVING SECURITY

Read the following words and word combinations and use them for understanding and translation of the text:

vulnerability - уязвимость
vigilance - бдительность
dissemination - распределение
to retrieve information – извлекать информацию
encryption - шифрование
decryption - дешифровка
patch — исправление уязвимости
alert - тревога
log — журнал регистрации событий
to install - устанавливать
gateway - шлюз
paging — страничная организация памяти
plaintext — незашифрованный текст
ciphertext — зашифрованный текст
confidentiality - секретность
checksum — контрольная сумма

In the face of the vulnerabilities and incident trends discussed above, a robust defense requires a flexible strategy that allows adaptation to the changing environment, well-defined policies and procedures, the use of robust tools, and constant vigilance.

Security policy

Factors that contribute to the success of a security policy include management commitment, technological support for enforcing the policy, effective dissemination of the policy, and the security awareness of all users. Management assigns responsibility for security, provides training for security personnel, and allocates funds to security. Technological support for the security policy moves some responsibility for enforcement from individuals to technology. The result is an automatic and consistent enforcement of policies, such as those for access and authentication.

Security-related procedures

Procedures are specific steps to follow that are based on the computer security policy. Procedures address such topics as retrieving programs from the network, connecting to the site's system from home or while traveling, using encryption, authentication for issuing accounts, configuration, and monitoring.

Security practices

System administration practices play a key role in network security. Checklists and general advice on good security practices are readily available. Below are examples of commonly recommended practices:

- Ensure all accounts have a password and that the passwords are difficult to guess. A one-time password system is preferable.
- Use tools such as MD5 checksums, a strong cryptographic technique, to ensure the integrity of system software on a regular basis.
- Use secure programming techniques when writing software.
- Be vigilant in network use and configuration, making changes as vulnerabilities become known.

- Regularly check with vendors for the latest available fixes, and keep systems current with upgrades and patches.
- Regularly check online security archives, such as those maintained by incident response teams, for security alerts and technical advice.
- Audit systems and networks, and regularly check logs. Many sites that suffer computer security incidents report that insufficient audit data is collected, so detecting and tracing an intrusion is difficult.

Security technology

A variety of technologies have been developed to help organizations secure their systems and information against intruders. These technologies help protect systems and information against attacks, detect unusual or suspicious activities, and respond to events that affect security.

One-time passwords. Intruders often install packet sniffers to capture passwords as they traverse networks during remote login processes. Therefore, all passwords should at least be encrypted as they traverse networks. A better solution is to use one-time passwords. These passwords are never repeated and are valid only for a specific user during the period that each is displayed. In addition, users are often limited to one successful use of any given password. One-time password technologies significantly reduce unauthorized entry at gateways requiring an initial password.

Firewalls. Intruders often attempt to gain access to networked systems by pretending to initiate connections from trusted hosts. They squash the emissions of the genuine host using a denial-of-service attack and then attempt to connect to a target system using the address of the genuine host. To counter these address-spoofing attacks and enforce limitations on authorized connections into the organization's network, it is necessary to filter all incoming and outgoing network traffic. Because firewalls are typically the first line of defense against intruders, their configuration must be carefully implemented and tested before connections are established between internal networks and the Internet.

Monitoring Tools. Continuous monitoring of network activity is required if a site is to maintain confidence in the security of its network and data resources. Network monitors may be installed at strategic locations to collect and examine information continuously that may indicate suspicious activity. It is possible to have automatic notifications alert system administrators when the monitor detects anomalous readings, such as a burst of activity that may indicate a denial-of-service attempt. Such notifications may use a variety of channels, including electronic mail and mobile paging. Sophisticated systems capable of reacting to questionable network activity may be implemented to disconnect and block suspect connections, limit or disable affected services, isolate affected systems, and collect evidence for subsequent analysis.

Cryptography

One of the primary reasons that intruders can be successful is that most of the information they acquire from a system is in a form that they can read and comprehend. One solution to this problem is, through the use of cryptography, to prevent intruders from being able to use the information that they capture.

Encryption is the process of translating information from its original form (called plaintext) into an encoded, incomprehensible form (called ciphertext). Decryption refers to the process of taking ciphertext and translating it back into plaintext. Any type of data may be encrypted, including digitized images and sounds.

Cryptography secures information by protecting its confidentiality. Cryptography can also be used to protect information about the integrity and authenticity of data. For example, checksums are often used to verify the integrity of a block of information. Cryptographic checksums (also called message digests) help prevent undetected modification of

information by encrypting the checksum in a way that makes the checksum unique.
The authenticity of data can be protected in a similar way. For example, to transmit information to a colleague by email, the sender first encrypts the information to protect its confidentiality and then attaches an encrypted digital signature to the message. When the colleague receives the message, he or she checks the origin of the message by using a key to verify the sender's digital signature and decrypts the information using the corresponding decryption key.

Notes:
MD5sum – программа, позволяющая вычислять значения хеш-сумм (контрольных сумм) файлов по алгоритму MD5.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. In the face of the vulnerabilities and incident trends, a robust defense requires a flexible strategy that allows adaptation to the changing environment, well-defined policies and procedures, the use of robust tools, and constant vigilance.
2. Factors that contribute to the success of a security policy include management commitment, technological support for enforcing the policy, effective dissemination of the policy, and the security awareness of all users.
3. Procedures address such topics as retrieving programs from the network, connecting to the site's system from home or while traveling, using encryption,

authentication for issuing accounts, configuration, and monitoring.

4. Intruders often install packet sniffers to capture passwords as they traverse networks during remote login processes.
5. They squash the emissions of the genuine host using a denial-of-service attack and then attempt to connect to a target system using the address of the genuine host. To counter these address-spoofing attacks and enforce limitations on authorized connections into the organization's network, it is necessary to filter all incoming and outgoing network traffic.
6. Sophisticated systems capable of reacting to questionable network activity may be implemented to disconnect and block suspect connections, limit or disable affected services, isolate affected systems, and collect evidence for subsequent analysis.
7. When the colleague receives the message, he or she checks the origin of the message by using a key to verify the sender's digital signature and decrypts the information using the corresponding decryption key.

2. Answer the following questions:

1. How can management contribute to the development of the security policy?
2. What are good security practices for the users?
3. What are the advantages of a one-time password?
4. How can the authenticity of data be protected?
5. What is a checksum for?

3. Translate into English:

Меры по защите.

1) Установите файрволл (firewall).2) Установите антивирусное и антишпионское ПО. Антивирусное ПО

должно запускаться автоматически при загрузке Windows и работать постоянно, проверяя запускаемые вами программы в фоновом режиме. Обязательно проверяйте на вирусы перед первым запуском любые программы, которые вы где-либо скачиваете или покупаете. 3) Не устанавливайте или удалите лишние ненужные службы Windows, которые не используете. Это ограничит возможности хакеров по доступу к вашему компьютеру. 4) Не открывайте подозрительные письма странного происхождения, не поддавайтесь на содержащиеся в них сомнительные предложения лёгкого заработка, не высылайте никому пароли от ваших аккаунтов, не открывайте прикреплённые к письмам подозрительные файлы и не переходите по содержащимся в них подозрительным ссылкам. 5) Не используйте простые пароли. Не используйте один и тот же пароль на все случаи жизни. 6) Будьте осторожны при выходе в Интернет из мест общего пользования (например, Интернет-кафе), а также при использовании прокси-серверов. Пароли, который вы вводите, в этом случае, с большей вероятностью могут быть украдены. 7) При использовании электронных платёжных систем типа webmoney или Яндекс-деньги, работа с ними через веб-интерфейс является менее безопасной, чем если вы скачаете и установите специальную программу (webmoney keeper).

4. Give the summary of the text using the key terms.

BIOMETRIC SECURITY TECHNOLOGY

Read the following words and word combinations and use them for understanding and translation of the text:

to forge - подделать
retina - сетчатка

iris — радужка (глаза)
gait - походка
moire fringe patterns — интерференционный муар
ultrasonics — ультразвуковое излучение
optical coupler — оптический разветвитель
pupil - зрачок
template-matching — сравнение с шаблонами
scam artist - аферист
frequent-flyer — постоянный авиапассажир
lock-down — запор, блокировка
sensitive data — уязвимые данные
to enlist — включить в список
covert surveillance — скрытое наблюдение
template - шаблон

Biometrics is gaining increasing attention these days. Security systems, having realized the value of biometrics, use biometrics for two basic purposes: to verify or identify users. There are a number of biometrics and different applications need different biometrics.

Biometric is the most secure and convenient authentication tool. It can not be borrowed, stolen, or forgotten and forging one is practically impossible. Biometrics measure individual's unique physical or behavioral characteristics to recognize or authenticate their identity. Common physical biometrics include fingerprints, hand or palm geometry, retina, iris, and facial characteristics. Behavioral characters characteristics include signature, voice, keystroke pattern, and gait. Of this class of biometrics, technologies for signature and voice are the most developed.

There are many biometric technologies to suit different types of applications. Here comes a list of biometrics:

Fingerprints - A fingerprint looks at the patterns found on a fingertip. There are a variety of approaches to fingerprint verification, such as traditional police method, using pattern-matching devices, and things like moire fringe patterns and ultrasonics. This seems to be a very good choice for in-house systems.

Hand geometry. This involves analyzing and measuring the shape of the hand. It might be suitable where there are more users or where users access the system infrequently. Accuracy can be very high if desired, and flexible performance tuning and configuration can accommodate a wide range of applications. Organizations are using hand geometry readers in various scenarios, including time and attendance recording.

Retina. A retina-based biometric involves analyzing the layer of blood vessels situated at the back of the eye. This technique involves using a low intensity light source through an optical coupler to scan the unique patterns of the retina. Retinal scanning can be quite accurate but does require the user to look into a receptacle and focus on a given point.

Iris. An iris-based biometric involves analyzing features found in the colored ring of tissue that surrounds the pupil. This uses a fairly conventional camera element and requires no close contact between the user and the reader. Further, it has the potential for higher than average template-matching performance.

Face. Face recognition analyses facial characteristics. It requires a digital camera to develop a facial image of the user for authentication. Because facial scanning needs extra peripheral things that are not included in basic PCs, it is more of a niche market for network authentication. However, the casino industry has capitalized on this technology to create a facial database of scam artists for quick detection by security personnel.

Signature. Signature verification analyses the way user signs his name. Signing features such as speed, velocity, and pressure are as important as the finished signature's static shape. People are used to signatures as a means of transaction-related identity verification.

Voice. Voice authentication is based on voice-to-print authentication, where complex technology transforms voice into text. Voice biometrics requires a microphone, which is available with PCs nowadays. Voice biometrics is to replace the currently used methods, such as PINs, passwords, or account names. But voice will be a complementary technique for finger-scan technology as many people see finger scanning as a higher authentication form.

Uses of Biometrics

For decades, many highly secure environments have used biometric technology for entry access. Today, the primary application of biometrics is in physical security: to control access to secure locations (rooms or buildings). Biometrics permit unmanned access control. Biometric devices, typically hand geometry readers, are in office buildings, hospitals, casinos, health clubs and lodges. Biometrics are useful for high-volume access control. There are several promising prototype biometric applications. One of them, EyeTicket, links a passenger's frequent-flyer number to an iris scan. Some of the US airports use a sort of hand geometry biometric technology for performing citizen-verification functions.

It is also expected that virtual access is the application that will move biometrics for network and computer access. Physical lock-downs can protect hardware, and passwords are currently the most popular way to protect data on a network. Biometrics can increase a company's ability to protect its sensitive data by implementing a more secure key than a password. Using biometrics also allows a hierarchical structure of data protection, making the data even more secure. Biometric technologies further help to enhance security levels of access to network data.

E-commerce developers are exploring the use of biometrics and smart cards to more accurately verify a trading party's identity. Banks are bound to use this combination to better authenticate customers and ensure non-repudiation of online banking, trading and purchasing transactions. Point-of-sales (POS) system vendors are working on the cardholder verification method, which would enlist smart cards and biometrics to replace signature verification. Biometrics can help to obtain secure services over the telephone through voice authentication..

The last interesting application is for covert surveillance. Using facial and body recognition technologies, researchers hope to use biometrics to automatically identify known suspects entering buildings or traversing crowded security areas such as airports.

Future Research Directions

Although companies are using biometrics for authentication in a variety of situations, biometric technologies are evolving and emerging towards a large scale of use. Standards are coming out to provide a common software interface to allow sharing of biometric templates and to permit effective comparison and evaluation of different biometric technologies. One of them is the Common Biometric Exchange File Format, which defines a common means of exchanging and storing templates collected from a variety of biometric devices.

Biometric assurance - confidence that a biometric can achieve the intended level of security - is another active research area. Another interesting thing to be examined is combining biometrics with smart cards and public-key infrastructure (PKI). A major problem with biometrics is how and where to store the user's template. Because the template represents the user's personal characters, its storage introduces privacy concerns. Also storing the template in a centralized database paves for attack and compromise. On the other hand, storing the template on a smart card enhances individual privacy and increased protection from attack, because individual users control their own templates. Vendors enhance security by placing more biometric functions directly on the smart card. Some vendors like Biometric Associates, have built a fingerprint sensor directly into the smart card reader, which in turn passes the biometric to the smart card for verification.

PKI uses public- and private-key cryptography for user identification and authentication. It has some advantages over biometrics as it is mathematically more secure and it can be used across the Internet. The main drawback of PKI is the management of the user's private key. To be secure, the private key must be protected from compromise and to be useful, the private key must be portable. The solution is to store the private key on a smart card and protect it with biometric. There are proposals for integrating biometrics, smart cards and PKI technology for designing Smart Access common government ID cards.

Notes:

PIN (Personal Identification Number) – личный опознавательный номер, аналог пароля.

EyeTicket – программное обеспечение для распознавания пассажира по радужной оболочке глаза.

Smart card – пластиковая карта со встроенной микросхемой, контролирующей устройство и доступ к объектам памяти.

Common Biometric Exchange File Format – единый формат представления биометрических данных.

Point-of-Sales system – программно-аппаратный комплекс, функционирующий на базе фискального регистратора. За системой закреплен типичный набор кассовых функций.

PKI (Public Key Infrustructure) – инфраструктура открытых ключей. Набор средств, распределенных служб и компонентов, используемых для поддержки криптозадач на основе открытого и закрытого ключей.

ID (Identifier) – идентификатор, уникальный признак объекта, позволяющий отличать его от других объектов.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. Common physical biometrics include fingerprints, hand or palm geometry, retina, iris, and facial characteristics. Behavioral characters characteristics include signature, voice, keystroke pattern, and gait.
2. There are a variety of approaches to fingerprint verification, such as traditional police method, using pattern-matching devices, and things like moire fringe patterns and ultrasonics. This seems to be a very good choice for in-house systems.
3. Accuracy can be very high if desired, and flexible performance tuning and configuration can accommodate a wide range of applications.
4. A retina-based biometric involves analyzing the layer of blood vessels situated at the back of the eye. This technique involves using a low intensity light source through an optical coupler to scan the unique patterns of the retina.
5. An iris-based biometric involves analyzing features found in the colored ring of tissue that surrounds the pupil.
6. Voice authentication is based on voice-to-print authentication, where complex technology transforms voice into text.
7. There are several promising prototype biometric applications. One of them, EyeTicket, links a passenger's frequent-flyer number to an iris scan.
8. Point-of-sales (POS) system vendors are working on the cardholder verification method, which would enlist smart cards and biometrics to replace signature verification.
9. Some vendors like Biometric Associates, have built a fingerprint sensor directly into the smart card reader, which in turn passes the biometric to the smart card for verification.

2. Answer the following questions:

1. What is the goal of biometrics?
2. Why is biometrics the most secure means of security?
3. What are biometric technologies? Characterize them in brief.
4. How can biometrics be used for secure network and computer access?
5. Why does storing a user's biometrics templates present a problem?

3. Translate into English:

На данный момент системы распознавания по отпечаткам пальцев занимают более половины биометрического рынка. Множество российских и зарубежных компаний занимаются производством систем управления доступом, основанных на методе дактилоскопической идентификации. По причине того, что это направление является одним из самых давнишних, оно получило наибольшее распространение и является на сегодняшний день самым разработанным. Сканеры отпечатков пальцев прошли действительно длинный путь к улучшению. Современные системы оснащены различными датчиками (температуры, силы нажатия и т.п.), которые повышают степень защиты от подделок. С каждым днем системы становятся все более удобными и компактными. По сути, разработчики достигли уже некоего предела в данной области, и развивать метод дальше некуда. Кроме того, большинство компаний производят готовые системы, которые оснащены всем необходимым, включая программное обеспечение. Интеграторам в этой области просто нет необходимости собирать систему самостоятельно, так как это невыгодно и займет больше времени и сил, чем купить готовую и уже недорогую при этом систему, тем более выбор будет действительно широк.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- An overview of possible threats and attacks.
- Technical trends affecting software security.
- Security goals.
- Preventive measures are the key point in the provision of software security.
- Identity theft.

CRYPTOGRAPHY AND DATA ENCRYPTION

TERMINOLOGY

Read the following words and word combinations and use them for understanding and translation of the text:

thereof - соответственно
thorough - тщательный
undecipherable code — код, не поддающийся расшифровке
forgery – подделка
cipher - шифр
to derive from ... – происходить из...

Cryptography and encryption have been used for secure communication for thousands of years.

Throughout history, military communication has had the greatest influence on encryption and the advancements thereof. The need for secure commercial and private communication has been led by the Information Age, which began in the 1980s. Although the Internet had been invented in the late 1960s, it did not gain a public face until the World Wide Web was invented in 1989. This new method of information exchange has caused a tremendous need for information security. A thorough understanding of cryptography and encryption will help people develop better ways to protect valuable information as technology becomes faster and more efficient.

Cryptography is the science or study of techniques of secret writing and message hiding. Cryptography constitutes any method in which someone attempts to hide a message, or the meaning thereof, in some medium.

Encryption is one specific element of cryptography in which one hides data or information by transforming it into an undecipherable code. Encryption typically uses a specified parameter or key to perform the data transformation. Some encryption algorithms require the key to be the same length as the message to be encoded, yet other encryption algorithms can operate on much smaller keys relative to the message. **Decryption** is often classified along with encryption as its opposite. Decryption of encrypted data results in the original data.

Encryption is used in everyday modern life. Encryption is most used among transactions over insecure channels of communication, such as the Internet. Encryption is also used to protect data being transferred between devices such as automatic teller machines (ATMs), mobile telephones, and many more. Encryption can be used to create digital signatures, which allow a message to be authenticated. When properly implemented, a digital signature gives the recipient of a message reason to believe the message was sent by the claimed sender. Digital signatures are very useful when sending sensitive email and other types of digital communication. This is relatively equivalent to traditional handwritten signatures, in that, a more complex signature carries a more complex method of forgery.

A cipher is an algorithm, process, or method for performing encryption and decryption. A cipher has a set of well-defined steps that can be followed to encrypt and decrypt messages. The operation of a cipher usually depends largely on the use of an encryption key. The key may be any auxiliary information added to the cipher to produce certain outputs.

Plaintext and ciphertext are typically opposites of each other. Plaintext is any information before it has been encrypted. Ciphertext is the output information of an encryption cipher. Many encryption systems carry many layers of encryption, in which the ciphertext output becomes the plaintext input to another encryption layer. The process of decryption takes ciphertext and transforms it back into the original plaintext.

In efforts to remain secure, governments have employed staff for studying encryption and the breaking thereof. **Cryptanalysis** is the procedures, processes, and methods used

to translate or interpret secret writings or communication as codes and ciphers for which the key is unknown.
Even though the goal has been the same, the methods and techniques of cryptanalysis have changed drastically through time. These changes derive from an attempt to adapt to the increasing complexity of cryptography. Due to the tremendous advantage of knowing the enemy's thoughts, war is the main driving force of cryptanalysis. Throughout history many governments have employed divisions solely for cryptanalysis during war time. Within the last century, governments have employed permanent divisions for this purpose.

Notes:
ATM (Automatic Teller Machine) – банкомат

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Throughout history, military communication has had the greatest influence on encryption and the advancements thereof.
2. Although the Internet had been invented in the late 1960s, it did not gain a public face until the World Wide Web was invented in 1989.
3. A thorough understanding of cryptography and encryption will help people develop better ways to protect valuable information as technology becomes faster and more efficient.

4. Cryptography constitutes any method in which someone attempts to hide a message, or the meaning thereof, in some medium.
5. When properly implemented, a digital signature gives the recipient of a message reason to believe the message was sent by the claimed sender. Digital signatures are very useful when sending sensitive email and other types of digital communication.
6. Plaintext is any information before it has been encrypted.
7. Ciphertext is the output information of an encryption cipher.
8. Due to the tremendous advantage of knowing the enemy's thoughts, war is the main driving force of cryptanalysis.
9. Throughout history many governments have employed divisions solely for cryptanalysis during war time.

2. Answer the following questions:

1. What is the difference between cryptography and cryptanalysis?
2. How can encryption be used in everyday life?
3. What does the operation of the cipher depend on?
4. What are the interrelations of a plaintext and ciphertext?
5. What is the main driving force of cryptanalysis?

3. Translate into English:

Криптография — наука о математических методах обеспечения конфиденциальности и аутентичности информации. Изначально криптография изучала методы шифрования информации — обратимого преобразования открытого (исходного) текста на основе секретного алгоритма и/или ключа в шифрованный текст (шифртекст). Традиционная криптография образует раздел симметричных криптосистем, в которых зашифрование и расшифрование проводится с использованием одного и того же секретного ключа. Помимо этого раздела современная криптография включает в себя асимметричные криптосистемы, системы электронной цифровой подписи, хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию.

Криптоанализ — наука о методах получения исходного значения зашифрованной информации, не имея доступа к секретной информации (ключу), необходимой для этого. В большинстве случаев под этим подразумевается нахождение ключа. В нетехнических терминах, криптоанализ есть взлом шифра (кода).

Под термином «криптоанализ» также понимается попытка найти уязвимость в криптографическом алгоритме или протоколе. Результаты криптоанализа конкретного шифра называют криптографической атакой на этот шифр. Успешную криптографическую атаку, полностью дискредитирующую атакуемый шифр, называют взломом или вскрытием.

4. Give the summary of the text using the key terms.

HISTORICAL CRYPTOGRAPHY

Read the following words and word combinations and use them for understanding and translation of the text:

scribe - переписчик
inscription - надпись
substitution cipher - подстановочный шифр
parchment - пергамент
to wind - наматывать
length-wise - по длине

a transposition cipher - перестановочный шифр
to overtake - опережать
grid - решетка
offset - сдвинутый
gibberish - бессмысленный
to intercept - перехватывать
brute force method - подбор методом грубой силы
overconfidence - самонадеянность
to map - преобразовать
stepping switch - шаговый переключатель
renowned - прославленный

Ancient Egypt

The earliest known text containing components of cryptography originates in the Egyptian town Menet Khufu on the tomb of nobleman Khnumhotep II nearly 4,000 years ago. In about 1900 B.C. Khnumhotep's scribe drew his master's life in his tomb. As he drew the hieroglyphics he used a number of unusual symbols to obscure the meaning of the inscriptions. This method of encryption is an example of a substitution cipher, which is any cipher system which substitutes one symbol or character for another.

As the Egyptian culture evolved, hieroglyphic substitution became more common. This method of encryption was relatively easy to break for those who could read and write. There are several possibilities why the Egyptians would use the sacred nature of their religious rituals from common cryptography is that the scribes wanted to give a formal appearance to their writings. This seems to be very similar to formal complicated language used in any modern legal document. Egyptian cryptography could also have been a way for a scribe to impress others by showing that he could write at a higher level.

Greece

In about 500 B.C. the Spartans developed a device called Scytale, which was used to send and receive secret messages. The device was a cylinder in which a narrow strip of parchment was wound. The message was then written length-wise on the parchment. Once it was unwound the message on the strip of parchment became unreadable. To receive the message an identical cylinder was needed. It was only then that the letters would line up resulting in the original message.

The Scytale is an example of a transposition cipher, which is any cipher system that changes the order of the characters rather than changing the characters themselves. In today's standards, the Scytale would be very easy to decipher, however, 2,500 years ago the percent of people that could read and write was relatively small. The Scytale provided the Spartans a secure method of communication.

Rome

The earliest recorded military use of cryptography comes from Julius Caesar 2,000 years ago. Caesar, being commander of the Roman army, solved the problem of secure communication with his troops. The problem was that messengers of secret military messages were often overtaken by the enemy. Caesar developed a substitution cipher method in which he would substitute letters for different letters. Only those who knew the substitution used could decipher the secret messages. Now when the messengers were overtaken the secret messages were not exposed. This gave the Roman army a huge advantage during war.

Caesar typically just shifted his letters by some predetermined number. This number was the cipher key of his algorithm. A randomized order of substitution yields a much larger amount of security due to the larger amount of possible orderings.

Alberti-Vigenere Cipher

During the mid 1400s a man named Leon Battista Alberti invented an encryption system using a cipher disk. This was a mechanical device with sliding disks that allowed for many different methods of substitution. This is the base concept of a polyalphabetic cipher, which is an encryption method which switches through several substitution ciphers throughout encryption. Alberti never developed his cipher disk concept.

In the 1500s Blaise De Vigenere, following Alberti's polyalphabetic cipher style, created a cipher that came to be known as the Vigenere Cipher. The Vigenere Cipher works exactly like the Caesar except that it changes the key throughout the encryption process. The Vigenere Cipher uses a grid of letters that give the method of substitution. This grid is called a Vigenere Square or a Vigenere Table. The grid is made up of 26 alphabets offset from each other by one letter.

The method of changing from one key to another follows one simple pattern. The encryption key was chosen as a special secret word. The corresponding letter is then substituted for the plaintext character. This method is repeated through all characters of the key word. After all characters of the key word are used, the word is just repeated.

Jefferson Wheel Cipher

In the late 1700s, Thomas Jefferson came up with a cipher system very similar to the Vigenere Cipher except with higher security. His invention was 26 wheels with the alphabet randomly scattered on each wheel. The wheels were numbered and ordered with a specified order. This order is the key to the encryption algorithm.

To message to be encrypted on the wheels lining up is made on the wheels such that the message is present. The ciphertext is any other line besides the line containing the original message. The person decrypting the ciphertext must have the wheels in the proper order. As the ciphertext is made on the wheels, the plaintext is lined up somewhere else on the wheels. A visual scan can quickly result in finding the original text. There is an extremely small chance that two non-gibberish messages will emerge on the disk during decryption.

Zimmerman Telegram

In early 1917, during the early stages of World War I, British cryptographers encountered a German encoded telegram. This telegram is often referred to as the Zimmerman Telegram. These cryptographers were able to decipher the telegram, and in doing so they changed cryptanalysis history. Using this deciphered message, they were able to convince the United States to join the war.

The Zimmerman Telegram was a secret communication between the Foreign Secretary of the German Empire, Arthur Zimmerman, and the German ambassador in Mexico, Heinrich von Eckardt. The telegram contained an offer for Mexico to reclaim its territory of New Mexico, Texas, and Arizona if it joined the German cause.

Choctaw Codetalkers

As WWI went on, the United States had the continuing problem of the lack of secure communication. Almost every phone call made was intercepted by the Germans, leaving every move made by the allies known to the Germans. Army commander, Captain Lewis devised a plan that utilized American Indian languages. He found eight Choctaw men in the battalion and used them to talk to each other over radio and phone lines. Their language was valuable because ordinary codes and ciphers of a shared language can be broken, whereas codes based on a unique language must be studied extensively before beginning to decode them. Within 24 hours of using the Choctaw language as encryption, the advantage fell in favor of the United States. Within 72 hours, the Germans were retreating and the allies were in full attack.

Enigma Encryption Machine

At the end of World War I, Arthur Scherbius invented the Enigma, an electro-mechanical machine that was used for encryption and decryption of secret messages. Because of the numerous configurations, the Enigma was virtually unbreakable with brute force methods.

It wasn't until World War II that the Enigma gained its fame. Due to the Enigma's statistical security, Nazi Germany became overconfident about their ability to encrypt secret messages. This overconfidence caused the downfall of the Enigma. Along with numerous German operator errors, the Enigma had several built-in weaknesses that Allied cryptographers exploited. The major weakness was that its substitution algorithm did not allow any letter to be mapped to itself. This allowed the Allied cryptographers to decrypt a vast number of ciphered messages sent by Nazi Germans.

Purple

While the Allied forces were focusing on cracking the German Enigma, the Japanese developed an encryption machine called Purple. In contrast to the Enigma's rotors, Purple was made using stepping switches commonly used for routing telephone signals. During the war, the Japanese were most efficient in destroying their encryption machines. Currently, not one complete Purple machine has been discovered.

Because the Japanese were so good at keeping their encryption methods secret, the United States cryptographers had a hard time decrypting their messages. William Friedman, a renowned cryptographer, and his team built a replica of Purple based only on the encrypted messages recovered. Because they had never seen a Purple machine and didn't know how it worked, this proved to be very difficult. Eventually the team figured out the encryption method used by Purple, and were able to build a different machine for the decryption of it. This advancement allowed the United States to access the Japanese diplomatic secrets in World War II.

Notes:

Khnumhotep – древнеегипетский высокопоставленный придворный вельможа.

B.C. (Before Christ) – до нашей эры.

Scytale – шифр Древней Спарты, прибор для перестановочного шифрования.

Choctaw – коренной народ США, проживавший изначально на юго-востоке.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. In about 1900 B.C. Khnumhotep's scribe drew his master's life in his tomb. As he drew the hieroglyphics he used a number of unusual symbols to obscure the meaning of the inscriptions.
2. The device was a cylinder in which a narrow strip of parchment was wound. The message was then written length-wise on the parchment.
3. The Scytale is an example of a transposition cipher, which is any cipher system that changes the order of the characters rather than changing the characters themselves
4. Only those who knew the substitution used could decipher the secret messages. Now when the messengers were overtaken the secret messages were not exposed.
5. A randomized order of substitution yields a much larger amount of security due to the larger amount of possible orderings.

6. There is an extremely small chance that two non-gibberish messages will emerge on the disk during decryption.
7. Because of the numerous configurations, the Enigma was virtually unbreakable with brute force methods.
8. The major weakness was that its substitution algorithm did not allow any letter to be mapped to itself.

2. Answer the following questions:

1. How does the substitution cipher work?
2. What is a Scytale?
3. What is a transposition cipher?
4. What idea underlies the Vegenere Cipher?
5. Why is it more difficult to break the code based on a unique language?
6. What is the Enigma known for?

3. Translate into English:

Криптография – тайнопись. Термин ввел Д. Валлис. Потребность шифровать и передавать шифрованные сообщения возникла очень давно. Так, еще в V-IV вв. до н. э. греки применяли специальное шифрующее устройство. По описанию Плутарха, оно состояло из двух палок одинаковой длины и толщины. Одну оставляли себе, а другую отдавали отъезжающему. Эти палки называли скиталами. Когда правителям нужно было сообщить какую-нибудь важную тайну, тогда вырезали длинную и узкую, вроде ремня, полосу папируса, наматывали ее на свою скиталу, не оставляя на ней никакого промежутка, так чтобы вся поверхность палки была охвачена этой полосой. Затем, оставляя папирус на скитале в том виде, как он есть, писали на нем все, что нужно, а написав, снимали полосу и без палки отправляли адресату. Так как буквы на ней разбросаны в беспорядке, то прочитать написанное он мог, только взяв свою скиталу и намотав на нее без пропусков эту полосу.

Аристотелю принадлежит способ дешифрования этого шифра. Надо изготовить длинный конус и, начиная с основания, обертывать его лентой с шифрованным сообщением, постепенно сдвигая ее к вершине. В какой-то момент начнут просматриваться куски сообщения. Так можно определить диаметр скиталы.

4. Give the summary of the text using the key terms.

MODERN ENCRYPTION

Read the following words and word combinations and use them for understanding and translation of the text:

one-time pad – блокнот одноразового использования
to withstand - противостоять
prior to... - перед чем-либо
to be referred to as... - называться
flaw - ошибка
staggering - ошеломляющий
power (зд.) - степень
conventional computers — традиционные компьютеры
to listen in - подслушать
key distribution problem — задача распределения ключей

Until the 1990s, cryptology was based on **algorithms** -- a mathematical process or procedure. These algorithms are used in conjunction with a **key**, a collection of bits (usually numbers). Without the proper key, it's virtually impossible to decipher an encoded message, even if you know what algorithm to use.

The **"one-time pad"** encryption algorithm was invented in the early 1900s, and has since been proven as unbreakable. The one-time pad algorithm is derived from a previous cipher called Vernam Cipher, named after Gilbert Vernam. The Vernam Cipher was a cipher that combined a message with a key read from a paper tape or pad. The Vernam Cipher was not unbreakable until Joseph Mauborgne recognized that if the key was completely random the cryptanalytic difficultly would be equal to attempting every possible key. Even when trying every possible key, one would still have to review each attempt at decipherment to see if the proper key was used. The unbreakable aspect of the one-time pad comes from two assumptions: the key used is completely random; and the key cannot be used more than once. The security of the one-time pad relies on keeping the key 100% secret. The one-time pad is typically implemented by using a modular addition (XOR) to combine plaintext elements with key elements. The key used for encryption is also used for decryption. Applying the same key to the ciphertext results back to the plaintext.

If any non-randomness occurs in the key of a one-time pad, the security is decreased and thus no more unbreakable. Numerous attempts have been made to create seemingly random numbers from a designated key. These number generators are called **Pseudo-Random Number Generators (PRNGs)** because they cannot give a completely random number stream. Even though the security of a PRNG is not 100% unbreakable, it can provide sufficient security when implemented correctly. PRNGs that have been designated secure for cryptographic use are called Cryptographically Secure Pseudo-Random Number Generators (CSPRNGs). CSPRNGs have qualities that other PRNGs do not. CSPRNGs must pass the "next-bit test" in that given the first k bits, there is no polynomial-time algorithm that can predict the (k+1)th bit with probability of success higher than 50%. CSPRNGs must also withstand "state compromises." In the event that part or all of its state is revealed, it should be impossible to reconstruct the stream of random numbers prior to the revelation.

There are limitless possibilities for keys used in cryptology. But there are only two widely used methods of employing keys: public-key cryptology and secret-key cryptology. In both of these methods (and in all cryptology), the sender (point A) is referred to as Alice. Point B is known as Bob.

In the **public-key cryptology (PKC)** method, a user chooses two interrelated keys. He lets anyone who wants to send him a message know how to encode it using one key. He makes this key public. The other key he keeps to himself. In this manner, anyone can send the user an encoded message, but only the recipient of the encoded message knows how to decode it. Even the person sending the message doesn't know what code the user employs to decode it.

PKC is often compared to a mailbox that uses two keys. One unlocks the front of the mailbox, allowing anyone with a key to deposit mail. But only the recipient holds the key that unlocks the back of the mailbox, allowing only him to retrieve the messages.

The other usual method of traditional cryptology is **secret-key cryptology (SKC)**. In this method, only one key is used by both Bob and Alice. The same key is used to both encode and decode the plaintext. Even the algorithm used in the encoding and decoding process can be announced over an unsecured channel. The code will remain uncracked as long as the key used remains secret.

SKC is similar to feeding a message into a special mailbox that grinds it together with the key. Anyone can reach inside and grab the cipher, but without the key, he won't be able to decipher it. The same key used to encode the message is also the only one that can decode it, separating the key from the message.

Both the secret-key and public-key methods of cryptology have unique flaws. The problem with public-key cryptology is that it's based on the staggering size of the numbers created by the combination of the key and the algorithm used to encode the message. These numbers can reach unbelievable proportions.

What's more, they can be made so that in order to understand each bit of output data, you have to also understand every other bit as well. This means that to crack a 128-bit key, the possible numbers used can reach upward to the 1038 power. That's a lot of possible numbers for the correct combination to the key.

The keys used in modern cryptography are so large, in fact, that a billion computers working in conjunction with each processing a billion calculations per second would still take a trillion years to definitively crack a key. This isn't a problem now, but it soon will be. Current computers will be replaced in the near future with quantum computers, which exploit the properties of physics on the immensely small quantum scale. Since they can operate on the quantum level, these computers are expected to be able to perform calculations and operate at speeds no computer in use now could possibly achieve. So the codes that would take a trillion years to break with conventional computers could possibly be cracked in much less time with quantum computers. This means that secret-key cryptology (SKC) looks to be the preferred method of transferring ciphers in the future.

But SKC has its problems as well. The chief problem with SKC is how the two users agree on what secret key to use. It's possible to send a message concerning which key a user would like to use, but shouldn't that message be encoded, too? And how do the users agree on what secret key to use to encode the message about what secret key to use for the original message? There's almost always a place for an unwanted third party to listen in and gain information the users don't want that person to have. This is known in cryptology as the **key** distribution problem.

RSA encryption, named for the surnames of the inventors, relies on multiplication and exponentiation being much faster than prime factorization. The entire protocol is built from two large prime numbers. These prime numbers are manipulated to give a public key and private key. Once these keys are generated they can be used many times. Typically one keeps the private key and publishes the public key. Anyone can then encrypt a message using the public key and send it to the creator of the keys. This person then uses the private key to decrypt the message. Only the one possessing the private key can decrypt the message. One of the special numbers generated and used in RSA encryption is the modulus, which is the product of the two large primes. In order to break this system, one must compute the prime factorization of the modulus, which results in the two primes. The strength of RSA encryption depends on the difficultly to produce this prime factorization. RSA Encryption is the most widely used **asymmetric key encryption** system used for electronic commerce protocols.

Notes:

One-time pad (другое название **Vernam Cipher**) – система симметричного шифрования. Является единственной системой шифрования, для которой доказана абсолютная криптографическая стойкость.

XOR (Exclusive or) – сложение по модулю 2, логическая и битовая операция.

Pseudo-Random Number Generator – генератор псевдослучайных чисел.

Cryptographically Secure Pseudo-Random Number Generator – криптографически безопасный генератор псевдослучайных чисел.

RSA encryption (аббревиатура от имен Rivest, Shamir, and Adleman) – криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The "one-time pad" encryption algorithm was invented in the early 1900s, and has since been proven as unbreakable.
2. Even when trying every possible key, one would still have to review each attempt at decipherment to see if the proper key was used.
3. The one-time pad is typically implemented by using a modular addition (XOR) to combine plaintext elements with key elements.
4. Numerous attempts have been made to create seemingly random numbers from a designated key.
5. CSPRNGs must pass the "next-bit test" in that given the first k bits, there is no polynomial-time algorithm that can predict the (k+1)th bit with probability of success higher than 50%.
6. SKC is similar to feeding a message into a special mailbox that grinds it together with the key.
7. RSA encryption, named for the surnames of the inventors, relies on multiplication and exponentiation being much faster than prime factorization.
8. One of the special numbers generated and used in RSA encryption is the modulus, which is the product of the two large primes. In order to break this system, one must compute the prime factorization of the modulus, which results in the two primes.

2. Answer the following questions:

1. Is the one-time pad an unbreakable means of encryption?
2. What two assumptions does the unbreakable aspect of the one-time pad come from?
3. What is the difference between PRNG and CSPRNG?
4. What is safer: PKC or SKC?
5. What can the strength of RSA encryption depend on?

3. Translate into English:

Как бы ни были сложны и надежны криптографические системы, их слабое место при практической реализации - проблема распределения ключей. Для того, чтобы был возможен обмен конфиденциальной информацией между двумя субъектами ИС, ключ должен быть сгенерирован одним из них, а затем каким-то образом опять же в конфиденциальном порядке передан другому. Т.е. в общем случае для передачи ключа опять же требуется использование какой-то криптосистемы.

Для решения этой проблемы на основе результатов, полученных классической и современной алгеброй, были предложены системы с открытым ключом.

Суть их состоит в том, что каждым адресатом ИС генерируются два ключа, связанные между собой по определенному правилу. Один ключ объявляется открытым, а другой закрытым. Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Секретный ключ сохраняется в тайне.

Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст в принципе не может быть расшифрован тем же открытым

4. Give the summary of the text using the key terms.

FUTURE METHODS OF ENCRYPTION

Read the following words and word combinations and use them for understanding and translation of the text:

eavesdropper - перехватчик
to harness — приспособить, поставить на службу
spin - вращение
binary code – двоичный код
coherent - понятный
to accomplish - выполнять
to discard - отбрасывать
discrepancy - несоответствие
parity check — контроль четности
to bounce - отскакивать
spooky — жуткий, зловещий
entanglement - переплетение
fiber optic cable – оптоволоконный кабель

Quantum Cryptology

One of the great challenges of cryptology is to keep unwanted parties – or **eavesdroppers** - from learning of sensitive information. Quantum physics has provided a way around this problem. By harnessing the unpredictable nature of matter at the quantum level, physicists have figured out a way to exchange information on secret keys.

Quantum cryptography uses photons to transmit a key. Once the key is transmitted, coding and encoding using the normal secret-key method can take place. But how does a photon become a key? How do you attach information to a photon's spin?

This is where **binary code** comes into play. Each type of a photon's spin represents one piece of information - usually a 1 or a 0, for binary code. This code uses strings of 1s and 0s to create a coherent message. So a binary code can be assigned to each photon. Alice can send her photons through randomly chosen filters and record the polarization of each photon. She will then know what photon polarizations Bob should receive. Bob has no idea what filter to use for each photon, he's guessing for each one. After the entire transmission, Bob and Alice have a non-encrypted discussion about the transmission.

The reason this conversation can be public is because of the way it's carried out. Bob calls Alice and tells her which filter he used for each photon, and she tells him whether it was the correct or incorrect filter to use. Since Bob isn't saying what his measurements are - only the type of filter he used - a third party listening in on their conversation can't determine what the actual photon sequence is.

In modern cryptology, Eve (E – an eavesdropper) can **passively intercept** Alice and Bob's encrypted message - she can get her hands on the encrypted message and work to decode it without Bob and Alice knowing she has their message. Eve can accomplish this in different ways, such as wiretapping Bob or Alice's phone or reading their secure e-mails.

Quantum cryptology is the first cryptology that safeguards against passive interception. Here's an example. If Alice sends Bob a series of polarized photons, and Eve has set up a filter of her own to intercept the photons, Eve is in the same boat as Bob: Neither has any idea what the polarizations of the photons Alice sent are. Like Bob, Eve can only guess which filter orientation she should use to measure the photons.

After Eve has measured the photons by randomly selecting filters to determine their spin, she will pass them down the line to Bob. She does to cover up her presence and the fact that she intercepted the photon message. But Eve's presence will be detected. By measuring the photons, Eve inevitably altered some of them.

Alice and Bob can further protect their transmission by discussing some of the exact correct results after they've discarded the incorrect measurements. This is called a **parity check**. If the chosen examples of Bob's measurements are all correct - meaning the pairs of Alice's transmitted photons and Bob's received photons all match up - then their message is secure.

Bob and Alice can then discard these discussed measurements and use the remaining secret measurements as their key. If discrepancies are found, they should occur in 50 percent of the parity checks. Since Eve will have altered about 25 percent of the photons through her measurements, Bob and Alice can reduce the likelihood that Eve has the remaining correct information down to a one-in-a-million chance by conducting 20 parity checks.

Quantum Cryptology Problems

Despite all of the security it offers, quantum cryptology also has a few fundamental flaws. Chief among these flaws is the length under which the system will work: It's too short.

The original quantum cryptography system, built in 1989 by Charles Bennett, Gilles Brassard and John Smolin, sent a key over a distance of 36 centimeters. Since then, newer models have reached a distance of 150 kilometers (about 93 miles). But this is still far short of the distance requirements needed to transmit information with modern computer and telecommunication systems.

The reason why the length of quantum cryptology capability is so short is because of interference. A photon's spin can be changed when it bounces off other particles, and so when it's received, it may no longer be polarized the way it was originally intended to be. As the distance a photon must travel to carry its binary message is increased, so, too, is the chance that it will meet other particles and be influenced by them.

One group of Austrian researchers may have solved this problem. This team used what Albert Einstein called "spooky action at a distance." This observation of quantum physics is based on the **entanglement** of photons. At the quantum level, photons can come to depend on one another after undergoing some particle reactions, and their states become entangled. This entanglement doesn’t mean that the two photons are physically connected, but they become connected in a way that physicists still don't understand. In entangled pairs, each photon has the opposite spin of the other. If the spin of one is measured, the spin of the other can be deduced. What’s strange (or “spooky”) about the entangled pairs is that they remain entangled, even when they’re separated at a distance.

The Austrian team put a photon from an entangled pair at each end of a fiber optic cable. When one photon was measured in one polarization, its entangled counterpart took the opposite polarization, meaning the polarization the other photon would take could be predicted. It transmitted its information to its entangled partner. This could solve the distance problem of quantum cryptography, since there is now a method to help predict the actions of entangled photons.

Even though it’s existed just a few years so far, quantum cryptography may have already been cracked. A group of researchers from Massachusetts Institute of Technology took advantage of another property of entanglement. In this form, two states of a single photon become related, rather than the properties of two separate photons. By entangling the photons the team intercepted, they were able to measure one property of the photon and make an educated guess of what the measurement of another property - like its spin - would be. By not measuring the photon’s spin, they were able to identify its direction without affecting it. So the photon traveled down the line to its intended recipient none the wiser.

The MIT researchers admit that their eavesdropping method may not hold up to other systems, but that with a little more research, it could be perfected. Hopefully, quantum cryptology will be able to stay one step ahead as decoding methods continue to advance.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. By harnessing the unpredictable nature of matter at the quantum level, physicists have figured out a way to exchange information on secret keys.
2. After the entire transmission, Bob and Alice have a non-encrypted discussion about the transmission.
3. Since Bob isn't saying what his measurements are -- only the type of filter he used -- a third party listening in on their conversation can't determine what the actual photon sequence is.
4. One of the great challenges of cryptology is to keep unwanted parties – or eavesdroppers -- from learning of sensitive information.
5. In modern cryptology, Eve (E – an eavesdropper) can passively intercept Alice and Bob's encrypted message -- she can get her hands on the encrypted message and work to decode it without Bob and Alice knowing she has their message.
6. After Eve has measured the photons by randomly selecting filters to determine their spin, she will pass them down the line to Bob.
7. If discrepancies are found, they should occur in 50 percent of the parity checks. Since Eve will have altered about 25 percent of the photons through her measurements, Bob and Alice can reduce the likelihood that Eve has the remaining correct information down to a one-in-a-million chance by conducting 20 parity checks.
8. As the distance a photon must travel to carry its binary message is increased, so, too, is the chance that it will meet other particles and be influenced by them.

9. At the quantum level, photons can come to depend on one another after undergoing some particle reactions, and their states become entangled.

2. Answer the following questions:

1. How can quantum physics help users to exchange information securely?
2. How does a photon become a key?
3. Can the users communicate openly using photons for encryption?
4. How can quantum cryptology safeguard against passive interception?
5. How does the parity check work?
6. What are the main flaws of quantum cryptology?
7. Is it possible to increase the quantum cryptology capability?

3. Translate into English:

Наибольшее практическое применение КК находит сегодня в сфере защиты информации, передаваемой по *волоконно-оптическим линиям связи (ВОЛС*). Это объясняется тем, что оптические волокна ВОЛС позволяют обеспечить передачу фотонов на большие расстояния с минимальными искажениями. В качестве источников фотонов применяются лазерные диоды передающих модулей ВОЛС; далее происходит существенное ослабление мощности светового сигнала - до уровня, когда среднее число фотонов на один импульс становится много меньше единицы. Системы передачи информации по ВОЛС, в приемном модуле которых применяются лавинные фотодиоды в режиме счета фотонов, называются *квантовыми оптическими каналами связи (КОКС).*

Понятно, что вследствие малой энергетики сигналов скорости передачи информации в КОКС по сравнению с возможностями современных ВОЛС не слишком высоки (от килобит до мегабит в секунду, в зависимости от реализации). Поэтому в большинстве случаев *квантовые криптографические системы (ККС)* применяются для распределения ключей, которые затем используются средствами шифрования высокоскоростного потока данных. Необходимо отметить, что квантово-криптографическое оборудование пока серийно не выпускается. Однако по мере совершенствования и удешевления применяемой элементной базы можно ожидать появления ККС на рынке телекоммуникаций в качестве, например, дополнительной услуги при построении корпоративных волоконно-оптических сетей.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- The Information Age and the increased vulnerability of sensitive data.
- Cryptography, cryptology, cryptanalysis, and their interrelation.
- Historical insight into the development of cryptography and cryptology.
- The evolution of modern encryption.
- The challenges of quantum cryptology.

COMPUTER GRAPHICS

HISTORY AND DEVELOPMENT

Read the following words and word combinations and use them for understanding and translation of the text:

in a sense - в широком смысле
to be widespread - быть распространенным
to improve - улучшать
to emerge - возникать, появляться
visual content - визуальное содержание
rendering - передача
to be coined by - быть созданным
amendable - поддающийся улучшению
to be hooked up to - быть подключенным к чему-либо
forbidding - запрещающий
non-obvious uses - неявные, неосновные использования

The term computer graphics has been used in a broad sense to describe "almost everything on computers that is not text or sound".

Computer graphics is widespread today. The computer imagery is found on television, in newspapers, for example in weather reports, or for example in all kinds of medical investigation and surgical procedures. A well-constructed graph can present complex statistics in a form that is easier to understand and interpret. In the media "such graphs are used to illustrate papers, reports, thesis", and other presentation material.

Many powerful tools have been developed to visualize data. Computer generated imagery can be categorized into several different types: 2D, 3D, and animated graphics. As technology has improved, 3D computer graphics have become more common, but 2D computer graphics are still widely used.

Computer graphics has emerged as a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Over the past decade, other specialized fields have been developed like information visualization, and scientific visualization more concerned with "the visualization of three dimensional phenomena (architectural, meteorological, medical, biological, etc.), where the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component".

The phrase "Computer Graphics" was coined in 1960 by William Fetter, a graphic designer for Boeing. The field of computer graphics developed with the emergence of computer graphics hardware. Early projects like the Whirlwind and SAGE Projects introduced the CRT as a viable display and interaction interface and introduced the light pen as an input device.

Most early mainframe business computers produced out- put only in the form of punched cards, paper tape, or text printouts. However, system designers realized that some kinds of data were particularly amenable to a graphical representation. In the early 1950s, the first systems using the cathode ray tube (CRT) for graphics output found specialized application. For example, the MIT Whirlwind and the Air Force's SAGE air defense system used a CRT to display information such as the location and heading of radar targets. By the late 1970s, the microcomputers from Apple, Radio Shack, Commodore, and others either included CRT monitors or had adapters that allowed them to be hooked up to regular television sets. These machines generally came with a version of the BASIC language that included commands for plotting lines and points and filling enclosed figures with color. While crude by modern standards, these graphics capabilities meant that spreadsheet programs could provide charts while games and simulations could show moving, interacting objects. Desktop computers showed pictures on television-like screens. Research at the

Xerox PARC laboratory in the 1970s demonstrated the advantages of a graphical user interface based on visual objects, including menus, windows, dialog boxes, and icons.

The Apple Macintosh, introduced in 1984, was the first commercially viable computer in which everything displayed on the screen (including text) consisted of bitmapped graphics. Microsoft's similar Windows operating environment became dominant on IBM architecture PCs during the 1990s.

Today Apple, Microsoft, and UNIX-based operating systems include extensive graphics functions. Game and multimedia developers can call upon such facilities as Apple QuickDraw and Microsoft Directx to create high resolution, realistic graphics.

What is computer graphics used for?

Obvious uses of computer graphics include computer art, CGI films, architectural drawings, and graphic design — but there are many non-obvious uses as well and not all of them are "artistic." Scientific visualization is a way of producing graphic output from computer models so it's easier for people to understand. Computerized models of global warming produce vast tables of numbers as their output, which only a PhD in climate science could figure out; but if you produce a speeded-up animated visualization — with the Earth getting bluer as it gets colder and redder as it gets hotter — anyone can understand what's going on. Medical imaging is another good example of how graphics make computer data more meaningful. When doctors show you a brain or body scan, you're looking at a computer graphic representation drawn using vast amounts of data produced from thousands or perhaps even millions of measurements. The jaw-dropping photos beamed back from space by amazing devices like the Hubble Space Telescope are usually enhanced with the help of a type of computer graphics called image processing; that might sound complex, but it's not so very different from using a graphics package like Google Picasa or PhotoShop to touch up your holiday snaps).

And that's really the key point about computer graphics: they turn complex computer science into everyday art we can all grasp, instantly and intuitively. Back in the 1980s, when programming a Commodore PET, the only way to get it to do anything was to type meaningless little words like PEEK and POKE onto a horribly unfriendly green and black screen. Virtually every modern computer now has what's called a GUI (graphical user interface), which means you operate the machine by pointing at things you want, clicking on them with your mouse or your finger, or dragging them around your "desktop." It makes so much more sense because we're visual creatures: something like a third of our cortex (higher brain) is given over to processing information that enters our heads through our eyes. That's why a picture really is worth a thousand words (sometimes many more) and why computers that help us visualize things with computer graphics have truly revolutionized the way we see the world.

Notes:

CRT - мониторы (Cathode Ray Tube) - самый распространенный тип. Как видно из названия, в основе всех подобных мониторов лежит катодно-лучевая трубка, или, как принято говорить в отечественной литературе, электронно-лучевая трубка (ЭЛТ).

Microsoft Directx - Microsoft DirectX - это ряд технологий, благодаря которым компьютеры на основе Windows становятся идеальной средой для запуска и отображения приложений, богатых элементами мультимедиа, такими как цветная графика, видео, трехмерная анимация и стереозвук. DirectX включает обновления, повышающие безопасность и производительность, а также новые функции, относящиеся к различным технологиям, к которым приложение может обращаться с помощью DirectX API.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. A well-constructed graph can present complex statistics in a form that is easier to understand and interpret.
2. Many powerful tools have been developed to visualize data.
3. Over the past decade, other specialized fields have been developed like information visualization.
4. The phrase "Computer Graphics" was coined in 1960 by William Fetter.
5. The field of computer graphics developed with the emergence of computer graphics hardware.
6. By the late 1970s, the microcomputers from Apple, Radio Shack, Commodore, and others either included CRT monitors or had adapters that allowed them to be hooked up to regular television sets.

2. Answer the following questions:

1. Where can the computer imagery be seen?
2. From what sub-field of computer science did the computer graphics emerge?
3. Who coined the term "computer graphics"?
4. What are the main steps of computer graphics development?
5. What are the main fields of computer graphics application?

3. Translate into English:

Область применения компьютерной графики не ограничивается одними художественными эффектами. Во всех отраслях науки, техники, медицины, в коммерческой и управленческой деятельности используются построенные с помощью компьютера схемы, графики, диаграммы, предназначенные для наглядного отображения разнообразной информации. Конструкторы, разрабатывая новые модели автомобилей и самолетов, используют трехмерные графические объекты, чтобы представить окончательный вид изделия. Архитекторы создают на экране монитора объемное изображение здания, и это позволяет им увидеть, как оно впишется в ландшафт.

4. Give the summary of the text using the key terms.

CONCEPTS AND PRINCIPLES

Read the following words and word combinations and use them for understanding and translation of the text:

binary format - двоичный формат

sequence raster images of ones and zeros - последовательность растровых изображений из единиц и нулей

arranged in - расположены в

grid - сетка, решетка

represented using dots or squares - представлены с использованием точек или квадратов

variable - переменная

deliberate - намеренный, обдуманный

distinctive style – отличительный (характерный) стиль

to vary - меняться

predictable - предсказуемый

ray tracing - трассировка лучей

shading - затемнение

to depict - отражать

accurate and smooth surface patches - точные и гладкие участки поверхности
polygonal mesh modeling - моделирование многоугольной сетки

Images are typically produced by optical devices; such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

A digital image is a representation of a two-dimensional image in binary format as a sequence of ones and zeros. Digital images include both vector images and raster images, but raster images are more commonly used.

Pixel

In the enlarged portion of the image individual pixels are rendered as squares and can be easily seen.

In digital imaging, a pixel (or picture element) is a single point in a raster image. Pixels are normally arranged in a regular 2-dimensional grid, and are often represented using dots or squares. Each pixel is a sample of an original image, where more samples typically provide a more accurate representation of the original. The intensity of each pixel is variable; in color systems, each pixel has typically three components such as red, green, and blue.

Graphics

Graphics are visual presentations on some surface, such as a wall, canvas, computer screen, paper, or stone to brand, inform, illustrate, or entertain. Examples are photographs, drawings, line art, graphs, diagrams, typography, numbers, symbols, geometric designs, maps, engineering drawings, or other images. Graphics often combine text, illustration, and color. Graphic design may consist of the deliberate selection, creation, or arrangement of typography alone, as in a brochure, flier, poster, web site, or book without any other element. Clarity or effective communication may be the objective, association with other cultural elements may be sought, or merely, the creation of a distinctive style.

Rendering

Rendering is the process of generating an image from a model (or models in what collectively could be called a scene file), by means of computer programs. A scene file contains objects in a strictly defined language or data structure; it would contain geometry, viewpoint, texture, lighting, and shading information as a description of the virtual scene. The data contained in the scene file is then passed to a rendering program to be processed and output to a digital image or raster graphics image file. The rendering program is usually built into the computer graphics software, though others are available as plug-ins or entirely separate programs. The term "rendering" may be by analogy with an "artist's rendering" of a scene. Though the technical details of rendering methods vary, the general challenges to overcome in producing a 2D image from a 3D representation stored in a scene file are outlined as the graphics pipeline along a rendering device, such as a GPU. A GPU is a purpose-built device able to assist a CPU in performing complex rendering calculations. If a scene is to look relatively realistic and predictable under virtual lighting, the rendering software should solve the rendering equation. The rendering equation does not account for all lighting phenomena, but is a general lighting model for computer-generated imagery. 'Rendering' is also used to describe the process of calculating effects in a video editing file to produce final video output.

3D projection

3D projection is a method of mapping three dimensional points to a two dimensional plane. As most current methods for displaying graphical data are based on planar two dimensional media, the use of this type of projection is widespread, especially in computer graphics, engineering and drafting.

Ray tracing

Ray tracing is a technique for generating an image by tracing the path of light through pixels in an image plane. The technique is capable of producing a very high degree of photorealism; usually higher than that of typical scanline rendering methods, but at a greater computational cost.

Shading

Shading refers to depicting depth in 3D models or illustrations by varying levels of darkness. It is a process used in drawing for depicting levels of darkness on paper by applying media more densely or with a darker shade for darker areas, and less densely or with a lighter shade for lighter areas. There are various techniques of shading including cross hatching where perpendicular lines of varying closeness are drawn in a grid pattern to shade an area. The closer the lines are together, the darker the area appears. Likewise, the farther apart the lines are, the lighter the area appears. The term has been recently generalized to mean that shaders are applied.

Texture mapping

Texture mapping is a method for adding detail, surface texture, or colour to a computer-generated graphic or 3D model. Its application to 3D graphics was pioneered by Dr. Edwin Catmull in 1974. A texture map is applied (mapped) to the surface of a shape, or polygon. This process is akin to applying patterned paper to a plain white box. Multitexturing is the use of more than one texture at a time on a polygon. Procedural textures (created from adjusting parameters of an underlying algorithm that produces an output texture), and bitmap textures (created in an image editing application or imported from a digital camera) are, generally speaking, common methods of implementing texture definition on 3D models in computer graphics software, while intended placement of textures onto a model's surface often requires a technique known as UV mapping (arbitrary, manual layout of texture coordinates) for polygon surfaces, while NURBS surfaces have their own intrinsic parameterization used as texture coordinates.

3D modeling

3D modeling is the process of developing a mathematical, wireframe representation of any three-dimensional object, called a "3D model", via specialized software. Models may be created automatically or manually; the manual modeling process of preparing geometric data for 3D computer graphics is similar to plastic arts such as sculpting. 3D models may be created using multiple approaches: use of NURBS curves to generate accurate and smooth surface patches, polygonal mesh modeling (manipulation of faceted geometry), or polygonal mesh subdivision (advanced tessellation of polygons, resulting in smooth surfaces similar to NURBS models). A 3D model can be displayed as a two-dimensional image through a process called 3D rendering, used in a computer simulation of physical phenomena, or animated directly for other purposes. The model can also be physically created using 3D Printing devices.

Notes:

Pixel- Пиксель (пиксел) – наименьший элемент изображения или экрана в виде квадратика (квадратной точки), который может иметь индивидуальные параметры: яркость, цвет и др. Размер пикселя может быть разным в зависимости от величины изображения и его разрешения, т. е. количества пикселов из которых оно состоит.

GPU (Graphics Processing Unit) - графический процессор. Он являет собой отдельное устройство игровой приставки, компьютера, фотоаппарата. Отвечает за рендеринг графики, выполняет его.

NURBS models - Non-Uniform Rational B-Spline - неоднородные рациональные B-сплайны. NURBS-кривые обладают одной особенностью: они всегда имеют гладкую форму.

Rendering - ре́ндеринг (англ. rendering — «визуализация») — термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. A digital image is a representation of a two-dimensional image in binary format as a sequence of ones and zeros.
2. Digital images include both vector images and raster images, but raster images are more commonly used.
3. Graphic design may consist of the deliberate selection, creation, or arrangement of typography alone, as in a brochure, flier, poster, web site, or book without any other element.
4. 3D modeling is the process of developing a mathematical, wireframe representation of any three-dimensional object, called a "3D model", via specialized software.
5. Models may be created automatically or manually; the manual modeling process of preparing geometric data for 3D computer graphics is similar to plastic arts such as sculpting.

2. Answer the following questions:

1. What is the pixel`s representation?
2. What is the method of 3D projection?
3. Who was the pioneer of texture mapping?

4. What are the techniques of shading?
5. Give the description of 3D modeling process.

3. Translate into English:

Трёхмерная графика обычно имеет дело с виртуальным, воображаемым трёхмерным пространством, которое отображается на плоской, двухмерной поверхности дисплея или листа бумаги. В настоящее время известно несколько способов отображения трехмерной информации в объемном виде, хотя большинство из них представляет объёмные характеристики весьма условно, поскольку работают со стереоизображением. Из этой области можно отметить стереоочки, виртуальные шлемы, 3D-дисплеи, способные демонстрировать трехмерное изображение. Несколько производителей продемонстрировали готовые к серийному производству трёхмерные дисплеи. Однако и 3D-дисплеи по-прежнему не позволяют создавать полноценной физической, осязаемой копии математической модели, создаваемой методами трехмерной графики.

4. Give the summary of the text using the key terms.

IMAGE TYPES

Read the following words and word combinations and use them for understanding and translation of the text:

added semantic value - добавленное семантическое значение
slight shift - небольшое изменение
to edit on the pixel level - редактировать на пиксельном уровне
to be complementary to - дополнять

to be akin to – быть сродни чему-то

relatively limited - относительно ограничены

instance - случай, пример

to rely on - полагаться на что-то, зависеть от чего-то

distinction - различие, отличие

keyframe - ключевой кадр

low bandwidth - медленный канал

virtual entities - виртуальные объекты

to be mapped - отображаться

Two-dimensional

2D computer graphics are the computer-based generation of digital images — mostly from two-dimensional models, such as 2D geometric models, text, and digital images, and by techniques specific to them.

2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawing, advertising, etc. In those applications, the two-dimensional image is not just a representation of a real-world object, but an independent artifact with added semantic value; two-dimensional models are therefore preferred, because they give more direct control of the image than 3D computer graphics, whose approach is more akin to photography than to typography.

Pixel art

Pixel art is a form of digital art, created through the use of raster graphics software, where images are edited on the pixel level. Graphics in most old (or relatively limited) computer and video games, graphing calculator games, and many mobile phone games are mostly pixel art.

The term pixel art was first published by Adele Goldberg and Robert Flegal of Xerox Palo Alto Research Center in 1982. The concept, however, goes back about 10 years before that, for example in Richard Shoup's SuperPaint system in 1972, also at Xerox PARC.

Some traditional art forms, such as counted-thread embroidery (including cross-stitch) and some kinds of mosaic and beadwork, are very similar to pixel art. These art forms construct pictures out of small colored units similar to the pixels of modern digital computing. A similar concept on a much bigger scale can be seen in the North Korean Arirang Festival.

Vector graphics

Vector graphics formats are complementary to raster graphics. Raster graphics is the representation of images as an array of pixels and is typically used for the representation of photographic images. Vector graphics consists in encoding information about shapes and colors that comprise the image, which can allow for more flexibility in rendering. There are instances when working with vector tools and formats is best practice, and instances when working with raster tools and formats is best practice. There are times when both formats come together. An understanding of the advantages and limitations of each technology and the relationship between them is most likely to result in efficient and effective use of tools.

Three-dimensional

3D computer graphics in contrast to 2D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. Such images may be for later display or for real-time viewing.

Despite these differences, 3D computer graphics rely on many of the same algorithms as 2D computer vector graphics in the wire frame model and 2D computer raster graphics in the final rendered display. In computer graphics software, the distinction between 2D and 3D is occasionally blurred; 2D applications may use 3D techniques to achieve effects such as lighting, and primarily 3D may use 2D rendering techniques.

3D computer graphics are often referred to as 3D models. Apart from the rendered graphic, the model is contained within the graphical data file. However, there are differences. A 3D model is the mathematical representation of any three-dimensional object. A model is not technically a graphic until it is visually displayed. Due to 3D printing, 3D models are not confined to virtual space. A model can be displayed visually as a two-dimensional image through a process called 3D rendering, or used in non-graphical computer simulations and calculations. There are some 3D computer graphics software for users to create 3D images e.g. Autocad, Photoshop, Solidwork, Google sketchup etc.

Computer animation

Computer animation is the art of creating moving images via the use of computers. It is a subfield of computer graphics and animation. Increasingly it is created by means of 3D computer graphics, though 2D computer graphics are still widely used for stylistic, low bandwidth, and faster real-time rendering needs. Sometimes the target of the animation is the computer itself, but sometimes the target is another medium, such as film. It is also referred to as CGI (Computer-generated imagery or computer-generated imaging), especially when used in films.

Virtual entities may contain and be controlled by assorted attributes, such as transform values (location, orientation, and scale) stored in an object's transformation matrix. Animation is the change of an attribute over time. Multiple methods of achieving animation exist; the rudimentary form is based on the creation and editing of keyframes, each storing a value at a given time, per attribute to be animated. The 2D/3D graphics software will interpolate between keyframes, creating an editable curve of a value mapped over time, resulting in animation. Other methods of animation include procedural and expression-based techniques: the former consolidates related

elements of animated entities into sets of attributes, useful for creating particle effects and crowd simulations; the latter allows an evaluated result returned from a user-defined logical expression, coupled with mathematics, to automate animation in a predictable way (convenient for controlling bone behavior beyond what a hierarchy offers in skeletal system set up).
To create the illusion of movement, an image is displayed on the computer screen then quickly replaced by a new image that is similar to the previous image, but shifted slightly. This technique is identical to the illusion of movement in television and motion pictures.

Notes:
Raster Graphics - Компьютерное растровое изображение представляется в виде прямоугольной матрицы, каждая ячейка которой задана цветной точкой — вместе они формируют целостную картинку. Пиксели подобны зернам фотографии и при значительном увеличении становятся заметными. Растровые изображения используются чаще векторных, так как они более просты в получении и допечатной подготовке.
CGI (Computer-generated imagery or computer-generated imaging) - стандарт интерфейса, используемого для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».
CGI также может означать «Computer-generated imagery» — компьютерные спецэффекты.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. 2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawings.
2. Vector graphics formats are complementary to raster graphics.
3. Raster graphics is the representation of images as an array of pixels and is typically used for the representation of photographic images. g, advertising, etc.
4. Despite these differences, 3D computer graphics rely on many of the same algorithms as 2D computer vector graphics in the wire frame model and 2D computer raster graphics in the final rendered display.

2. Answer the following questions:

1. Where is 2D graphics traditionally used?
2. What do you know about pixel art?
3. What does vector graphics consist in?
4. What are the main differences between 2D and 3D graphics?
5. What methods of animation do you know?

3. Translate into English:

Компьютерная графика - технология создания и обработки графических изображений средствами вычислительной техники.

Компьютерная графика изучает методы получения изображений полученных на основании невизуальных данных или данных, созданных непосредственно пользователем.

Растровая графика (raster graphics) — вид компьютерной графики, используемой в приложениях, в частности, для рисования, близкого по технике к традиционному процессу (на бумаге или холсте). Данные в памяти ЭВМ хранятся в виде «карты» яркости и цвета для каждого элемента изображения (пикселя) или прямоугольной матрицы пикселей (bitmap), дополненной данными о цвете и яркости каждого из них, а также способе сжатия записи и другими сведениями которые могут содержаться в «заголовке» и «концовке» файла.

Векторная графика (vector graphics) — вид компьютерной графики, используемой в приложениях для рисования. В отличие от растровой графики позволяет пользователю создавать и модифицировать исходные изобразительные образы при подготовке рисунков, технических чертежей и диаграмм путем их вращения, увеличения или уменьшения, растягивания. Графические образы создаются и хранятся в памяти ЭВМ в виде формул, описывающих различные геометрические фигуры, которые являются компонентами изображения.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- History and development of computer graphics
- 3D modeling
- Computer animation

ARTIFICIAL INTELLIGENCE: OVERVIEW

DEFINITIONS

Read the following words and word combinations and use them for understanding and translation of the text:

to impart - наделять
attribute - определение
highest good – высшее благо
offhand – импровизированный, сделанный на скорую руку
enclosed surface – замкнутое пространство
behavior pattern – модель поведения
observable – наблюдаемый
insufficient definition – неполное определение
tersely and concisely – сжато и кратко
to lose relevance – терять актуальность
human reasoning – мышление человека, человеческое мышление
to adjust – приспосабливаться
productive approach - плодотворный подход
to a limited extent – в определенных пределах
chatterbot – чатбот, программа «виртуальный собеседник»

The term *artificial intelligence* stirs emotions. For one thing there is our fascination with *intelligence*, which seemingly imparts to us humans a special place among life forms. Questions arise such as “What is intelligence?”, “How can one measure intelligence?” or “How does the brain work?” All these questions are meaningful when trying to understand artificial intelligence. However, the central question for the engineer, especially for the computer scientist, is the question of the intelligent machine that behaves like a person, showing intelligent behavior. The attribute *artificial* might awaken much different associations. It brings up fears of intelligent cyborgs. It recalls images from science fiction novels. It raises the question of whether our highest good, the soul, is something we should try to understand, model, or even reconstruct. With such different offhand interpretations, it becomes difficult to define the term *artificial intelligence* or *AI* simply and robustly.

In 1955, John McCarthy, one of the pioneers of AI, was the first to define the term *artificial intelligence*, roughly as follows: The goal of AI is to develop machines that behave as though they were intelligent.

To test this definition, imagine the following scenario. Fifteen or so small robotic vehicles are moving on an enclosed square surface. One can observe various behavior patterns. Some vehicles form small groups with relatively little movement. Others move peacefully through the space and gracefully avoid any collision. Still others appear to follow a leader. Aggressive behaviors are also observable. Is what we are seeing intelligent behavior? According to McCarthy's definition these robots can be described as intelligent, thus it is clear that this definition is insufficient.

In the Encyclopedia Britannica one finds a definition that goes like: AI is the ability of digital computers or computer controlled robots to solve problems that are normally associated with the higher intellectual processing capabilities of humans . . . But this definition also has weaknesses. It would admit, for example, that a computer that can save a long text and retrieve it on demand displays intelligent capabilities, for memorization of long texts can certainly be considered a higher intellectual processing capability of humans, as can, for example, the quick multiplication of two 20-digit numbers. According to this definition, then, every computer is an AI system. This dilemma is solved elegantly by the following definition by Elaine Rich: Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better.

Rich, tersely and concisely, characterizes what AI researchers have been doing for the last 50 years. Even in the year 2050, this definition will be up to date.

Tasks such as the execution of many computations in a short amount of time are the strong points of digital computers. In this regard they outperform humans by many multiples. In many other areas, however, humans are far superior to machines. For instance, a person entering an unfamiliar room will recognize the surroundings within fractions of a second and, if necessary, just as swiftly make decisions and plan actions. To date, this task is too demanding for autonomous robots. According to Rich's definition, this is, therefore, a task for AI. In fact, research on autonomous robots is an important, current theme in AI. Construction of chess computers, on the other hand, has lost relevance because they already play at or above the level of grandmasters.

It would be dangerous, however, to conclude from Rich's definition that AI is only concerned with the pragmatic implementation of intelligent processes. Intelligent systems, in the sense of Rich's definition, cannot be built without a deep understanding of human reasoning and intelligent action in general, because of which neuroscience is of great importance to AI. This also shows that the other cited definitions reflect important aspects of AI. A particular strength of human intelligence is adaptivity. We are capable of adjusting to various environmental conditions and change our behavior accordingly through *learning*. Precisely because our learning ability is so vastly superior to that of computers, *machine learning* is, according to Rich's definition, a central subfield of AI.

In 1950, computer pioneer Alan M. Turing suggested a productive approach to evaluating claims of artificial intelligence in what became known as the Turing test. He gave a definition of an intelligent machine, in which the machine in question must pass the following test. The test person Alice sits in a locked room with two computer terminals. One terminal is connected to a machine, the other with a non-malicious person

Bob. Alice can type questions into both terminals. She is given the task of deciding, after five minutes, which terminal belongs to the machine. The machine passes the test if it can trick Alice at least 30% of the time.

Computer programs have been able to pass the Turing test to a limited extent. The AI pioneer and social critic JosephWeizenbaum developed a program named Eliza, which is meant to answer a test subject's questions like a human psychologist. He was in fact able to demonstrate success in many cases. Supposedly his secretary often had long discussions with the program. Today in the internet there are many so-called chatterbots, some of whose initial responses are quite impressive. After a certain amount of time, however, their artificial nature becomes apparent.

Notes:

John McCarthy (1927 - 2011) was a legendary computer scientist at Stanford University who developed time-sharing, invented LISP, and founded the field of Artificial Intelligence.

Elaine Rich works as Distinguished Senior Lecturer at the University of Texas at Austin. Books: Automata, Computability and Complexity: Theory and Applications (author), Artificial Intelligence (co-author).

Joseph Weizenbaum (1923 - 2008) was a German-American computer scientist who is famous for his development of the Eliza program in 1966 and for his views on the ethics of artificial intelligence. He became sceptical of artificial intelligence and a leading critic of the AI field following the response of users to the Eliza program.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The term *artificial intelligence* stirs emotions. For one thing there is our fascination with *intelligence*, which seemingly imparts to us humans a special place among life forms.
2. With such different offhand interpretations, it becomes difficult to define the term *artificial intelligence* or *AI* simply and robustly.
3. AI is the ability of digital computers or computer controlled robots to solve problems that are normally associated with the higher intellectual processing capabilities of humans.
4. In this regard they outperform humans by many multiples. In many other areas, however, humans are far superior to machines.
5. It would be dangerous, however, to conclude from Rich's definition that AI is only concerned with the pragmatic implementation of intelligent processes.

2. Answer the following questions:

1. What is the key AI problem to be addressed by computer scientists?
2. Why is McCarthy's definition called "insufficient"?
3. What is wrong with the definition of AI in the Encyclopedia Britannica?
4. Where do machines outperform humans? Where do people win?
5. What is the essence of the Turing test?

3. Translate into English:

Эрик Браун, 45-летний исследователь из IBM, отвечает за мозг суперкомпьютера *Ватсон*, который в 2011 г. получил известность победами над людьми в популярной телевикторине. Самая большая трудность для Брауна, как наставника машины, не в том, чтобы впихнуть в Ватсона как можно больше знаний, но в том, чтобы придать тонкость его пониманию языка. Например, научить слэнгу.

Как проверить, может ли компьютер «мыслить»? Классический тест — так называемый тест Тьюринга — прост: он предполагает способность вести светскую беседу. Если бы компьютер сумел бы не выдать свою двоичную сущность в непринужденном разговоре, он бы доказал свое интеллектуальное превосходство. Но пока ни одной машине это не удалось.

Два года назад Браун попытался натаскать Ватсона с помощью популярного веб-сайта Urban Dictionary. Словарные статьи на сайте составляются обычными пользователями и редактируются добровольными редакторами по достаточно произвольным правилам. Тут есть всевозможные актуальные аббревиатуры как bb (англ. bye bye) — пока, hf (англ. have fun) — отлично повеселиться, w8 (англ. wait) — жди. В том числе и огромное количество всяких слэнговых конструкций, таких, как hot mess — «горячая штучка».

Но Ватсон не мог различить салонную лексику и слэнговую — которой в Urban Dictionary хватает. Кроме того, из-за чтения Википедии Ватсон приобрел некоторые дурные привычки. В ответах на вопросы исследователя в тестах он использовал малоцензурные словечки.

В конечном счете команда Брауна разработала фильтр, чтобы отцеживать брань Ватсона, и выскребла Urban Dictionary из его памяти. Это испытание доказывает, насколько тернист будет путь любого железного интеллектуала к “лёгкой болтовне”. Теперь Браун подготавливает Ватсона к использованию в качестве диагностического инструмента в больнице: там знание всяких модных аббревиатур не потребуется.

4. Give the summary of the text using the key terms.

APPROACHES AND TECHNIQUES

Read the following words and word combinations and use them for understanding and translation of the text:

conventional - общепринятый, традиционный

computational intelligence - вычислительный интеллект

machine learning - машинное обучение

case-based reasoning - вывод (рассуждения), основанные на прецедентах

behavior-based AI - поведенческий ИИ

referred to as - под названием, именуемый

neural networks - нейронные сети

fuzzy logic - нечеткая логика

neats versus scruffies - чистюли против нерях

ad hoc rules - ситуативные правила

inference engine - механизм логического вывода

forward chaining - прямой логический вывод

backward chaining - обратный логический вывод

directed acyclic graph - ориентированный циклический граф

arc - дуга

conditional dependence - условная зависимость

to be subject to controversy - вызывать споры

track record - послужной список, достижения

The artificial intelligence community can be roughly divided into two schools of thought: conventional AI and computational intelligence. *Conventional AI* is based on machine learning, which is the development of the techniques and algorithms that allow machines to "learn" or at least simulate learning. Machine learning attempts to use computer programs to generate patterns or rules from large data sets. This problem is similar to the data-mining problem (and data mining is one area where AI has found commercial success). Machine learning makes heavy use of symbolic formalism and logic, as well as statistics. Key areas in conventional AI include case-based reasoning, behavior-based AI, Bayesian networks, and expert systems. *Computational intelligence*, in contrast, relies more on clever algorithms (heuristics) and computation and less on formal logical systems. Computational intelligence is sometimes referred to as soft computing. It often involves iterative methods using computation to generate intelligent agents. Whereas conventional AI is considered to be a *top-down approach*, with the structure of solutions imposed from above, computational intelligence is more *bottom-up*, where solutions emerge from an unstructured initial state. Two areas of computational intelligence will be discussed further: neural networks and fuzzy logic. Hybrid intelligent systems attempt to combine the two approaches. Some proponents claim that this is appropriate, because the human mind uses multiple techniques to develop and verify results, and hybrid systems show some promise.

Another distinction within the artificial intelligence community is *weak AI* versus *strong AI*. Weak AI refers to using software to solve particular problems or reasoning tasks that do not encompass fully human intelligence. Strong AI implies creating artificial systems that are fully self-aware, the systems that can reason and independently solve problems. Current research is nowhere near creating strong AI, and a lively debate is ongoing as to whether this is even possible.

Another division in the artificial intelligence community is over the best way to design an intelligent system (*Neats* versus *Scruffies*). The Neats maintain that the solution should be elegant, obvious, and based on formal logic. The Scruffies hold that intelligence is too messy and complicated to be solved under the limitations the Neats propose. Interestingly, some good results have come from hybrid approaches, such as putting ad hoc rules (Scruffy style) into a formal (Neat) system. Not surprisingly, the Neats are often associated with conventional artificial intelligence, whereas the Scruffies are usually associated with computational intelligence.

Conventional AI has achieved success in several areas. **Expert systems**, or knowledge-based systems, attempt to capture the domain expertise of one or more humans and apply that knowledge. Most commonly, this is done by developing a set of rules that analyze information about a problem and recommend a course of action. Expert systems demonstrate behavior that appears to show reasoning. Expert systems work best in organizations with high levels of know-how and expertise that are difficult to transfer among staff. The simpler expert systems are all based on binary logic, but more sophisticated systems can include methods such as fuzzy logic. At the heart of an expert system is an *inference engin*e, a program that attempts to create answers from the *knowledge base* of rules provided by the expert. Knowledge engineers convert a human expert's "rules-of-thumb" into inference rules, which are if-then statements that provide an action or a suggestion if a particular statement is true. The inference engine then uses these inference rules to reason out a solution. *Forward chaining* starts with the available information and tries to use the inference rules to generate more data until a solution is reached. *Backward-chaining* starts with a list of solutions and works backward to see if data exists that will allow it to conclude that any of the solutions are true. Expert systems are used in many fields, including finance, medicine, and automated manufacturing.

Another approach from conventional AI that has achieved some commercial success is **case-based reasoning**, or CBR, which attempts to solve new problems based on past solutions of similar problems. Proponents argue that case-based reasoning is a critical element in human problem solving. As formalized in computer reasoning, CBR is composed of four steps: retrieve, reuse, revise, retain. First, access the available information about the problem (Retrieve). Second, try to extend a previous solution to the current problem (Reuse). Next, test the refactored solution and revise it if necessary (Revise). Finally, store the new experience into the knowledge base (Retain).

Behavior-based artificial intelligence (BBAI) attempts to decompose intelligence into a set of distinct, semi-autonomous modules. BBAI is popular in the robotics field and is the basis for many Robocup robotic soccer teams, as well as the Sony Aibo. A BBAI system is composed of numerous simple behavior modules, which are organized into layers. Each layer represents a particular goal of the system, and the layers are organized hierarchically. A low layer might have a goal of "avoid falling," whereas the layer above it might be "move forward." The move forward layer might be one component of a larger "walk to the store" goal. The layers can access sensor data and send commands to the robot's motors. The lower layers tend to function as reflexes, whereas the higher layers control more complex goal-directed behavior.

Bayesian networks are another tool in the conventional AI approach. They are heavily based upon probability theory. The problem domain is represented as a network. This network is a directed acyclic graph where the nodes represent variables, and the arcs represent conditional dependences between the variables. Graphs are easy to work with, so Bayesian networks can be used to produce models that are simple for humans to understand, as well as effective algorithms for inference and learning. Bayesian networks have been successfully applied to numerous areas, including medicine, decision support systems, and text analysis, including optical character recognition.

There is no widespread agreement yet on exactly what Computational intelligence (CI) is, but it is agreed that it includes **neural networks** and **fuzzy computing**. A **neural network** consists of many nodes that cooperate to produce an output. The system is trained by supplying input on the solution of known problems, which changes the weighting between the nodes. After training has tuned the parameters between the connections, neural networks can solve difficult problems in machine vision and other areas. Also known as neurocomputing, or parallel distributed processing, neural networks loosely model structures in the human brain. Neural network outputs rely on the cooperation of individual nodes. Data processing in neural networks is typically done in parallel, rather than sequentially as is the standard for nearly all modern computers. Neural nets can generalize from their training, and solve new problems, so they are self-adaptive systems. Neural networks have been criticized as "bad science" because it is difficult to explain exactly how they work. Nonetheless, neural networks have been successfully applied in areas as diverse as credit card fraud detection, machine vision, chess, and vehicle control.

Fuzzy logic, fuzzy systems, and fuzzy set theory are all ways to refer to reasoning that is based upon approximate values, rather than precise quantities. Modern computers are built upon binary, or Boolean, logic that is based on ones and zeros. The bit is zero or one, yes or no, with no middle ground. Fuzzy systems provide for a broader range of possible values. Consider the question, "Are the books in the study?" Well, yes, there are books in the study. There are also books in the office, books in the bedroom, and a pile of books in the doorway to the study. Fuzzy logic provides for an answer of 72%, meaning that 72% of the books are in the study. Fuzzy sets are based on vague definitions of sets. They are not random. Fuzzy logic is not imprecise; rather, it is a formal mathematical technique for handling imprecise data. Like neural networks, fuzzy logic is subject to controversy and criticism. But systems based on fuzzy logic have an excellent track record at certain types of problems. Antilock braking systems are based on fuzzy logic, and many appliances incorporate fuzzy logic.

Notes:

Bayesian network (Bayesian network, Bayes network, belief network, Bayes(ian) model or probabilistic directed acyclic graphical model) is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

Robocup is an international robotics competition that aims to develop autonomous robots with the intention of developing research and education in the field of artificial intelligence. The best universities in the world compete in several leagues.

AIBO (Artificial Intelligence robot) is a robotic project from Sony. In Japanese, AIBO means pal or partner. AIBO was one of several types of robotic pets that were designed and manufactured by Sony. **Sony Aibo** is basically a robotic dog that that is able to walk and "see" its environment using the on board cameras. It is even able to recognize spoken commands in languages including Spanish and English.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. Machine learning makes heavy use of symbolic formalism and logic, as well as statistics.
2. Whereas conventional AI is considered to be a *top-down approach*, with the structure of solutions imposed from above, computational intelligence is more *bottom-up*, where solutions emerge from an unstructured initial state.
3. Some proponents claim that this is appropriate, because the human mind uses multiple techniques to develop and verify results, and hybrid systems show some promise.

4. Weak AI refers to using software to solve particular problems or reasoning tasks that do not encompass fully human intelligence.
5. Current research is nowhere near creating strong AI, and a lively debate is ongoing as to whether this is even possible.
6. The Neats maintain that the solution should be elegant, obvious, and based on formal logic. The Scruffies hold that intelligence is too messy and complicated to be solved under the limitations the Neats propose.
7. Expert systems, or knowledge-based systems, attempt to capture the domain expertise of one or more humans and apply that knowledge.
8. There is no widespread agreement yet on exactly what Computational intelligence (CI) is, but it is agreed that it includes neural networks and fuzzy computing.

2. Answer the following questions:

1. What are the ways to classify Artificial Intelligence?
2. How does an expert system work?
3. What are the four steps in case-based reasoning?
4. What tasks do the layers of a BBAI system perform?
5. Where are Bayesian networks applied?
6. What are the working principles of a neural network?
7. How does fuzzy logic differ from Boolean logic?

3. Translate into English:

Эпименид Кносский с острова Крит – полумифический поэт и философ, живший в VI в. до н.э., однажды заявил: «Все критяне – лжецы!». Так как он и сам был критянином, то его помнят как изобретателя так называемого критского парадокса.

В терминах аристотелевой логики, в которой утверждение не может быть одновременно истинным и ложным, и подобные самоотрицания не имеют смысла. Если они истинны, то они ложны, но если они ложны, то они истинны.

И здесь на сцену выходит нечеткая логика, где переменные могут быть частичными членами множеств. Истинность или ложность перестают быть абсолютными – утверждения могут быть частично истинными и частично ложными. Использование подобного подхода позволяет строго математически доказать, что парадокс Эпименида ровно на 50% истинен и на 50% ложен.

Таким образом, нечеткая логика в самой своей основе несовместима с аристотелевой логикой, особенно в отношении закона Tertium non datur («Третьего не дано» – лат.), который также называют *законом исключения среднего*. Если сформулировать его кратко, то звучит он так: если утверждение не является истинным, то оно является ложным. Эти постулаты настолько базовые, что их часто просто принимают на веру.

Более банальный пример пользы нечеткой логики можно привести в контексте концепции холода. Большинство людей способно ответить на вопрос: «Холодно ли вам сейчас?». В большинстве случаев люди понимают, что речь не идет об абсолютной температуре по шкале Кельвина. Хотя температуру в 0 К можно, без сомнения, назвать холодом, но температуру в +15 С многие холодом считать не будут.

Но машины не способны проводить такую тонкую градацию. Если стандартом определения холода будет «температура ниже +15 С», то +14,99 С будет расцениваться как холод, а +15 С – не будет!

4. Give the summary of the text using the key terms.

CURRENT TRENDS

Read the following words and word combinations and use them for understanding and translation of the text:

hype - навязчивая (агрессивная) реклама, рекламная шумиха, раскрутка

acquisitions - приобретение

buzz word - модное словцо

cognitive computing - когнитивные вычисления, познавательные вычисления

predictive analytics - прогнозная аналитика

two-voice counterpoint - двухголосая полифония (контрапункт)

visual cue - визуальная подсказка

to underpin - лежать в основе

big data - большие данные, супермассив данных

to sift through - перелопатить

evidence - реальные факты, полученные сведения

to leverage - выгодно использовать, по-новому применять

relevant information - необходимая (актуальная) информация

It looks like the beginning of a new technology hype for artificial intelligence (AI). The media has started flooding the news with product announcements, acquisitions, and investments. AI is capturing the attention of tech firm and investor giants such as Google, Microsoft, and IBM. The buzz words are great too: cognitive computing, deep learning, AI2.

For those who started their careers in AI and left in disillusionment or data scientists today, the consensus is often that artificial intelligence is just a new fancy marketing term for good old predictive analytics. They point to the reality of Apple's Siri to listen and respond to requests as adequate but more often frustrating. Or, IBM Watson's win on Jeopardy as data loading and brute force programming.

But, is this fair? No. New AI breaks the current rule that machines must be better than humans: they must be smarter, faster analysts, or manufacture things better and cheaper.
New AI says:

- *The question is sometimes more important than the answer*. Suggestions don't always need to be answers, they can be questions. Eric Horvitz of Microsoft told MIT Technology Review, "…Another possibility is to build systems that understand the value of information, meaning they can automatically compute what the next best question to ask is...."
- *Improvisation is the true meaning of adaptation.* Search on 'artificial intelligence' and 'improvisation' and you get a lot of examples of AI being linked to music. The head of Facebook's AI lab and musician, Yan Lecun, says,"I have always been interested in Jazz because I have always been intrigued by the intellectual challenge of improvising music in real time". Linking the two, he wrote a program that automatically composed two-voice counterpoint for a college artificial intelligence project.
- *Collaboration produces better results.* Guy Hoffman at the Media Innovation Lab, School of Communication, IDC Herzliya introduced a robot that could not only compose music independently, but also collaborate with another musician (Guy himself) to create a new piece of music. The robot provided visual cues, reacting and communicating the effect of the music and creative process for lifelike interaction between robot and composer.

This is game changing, both in how organizations operate and strategize as well as the impact on customer experience. These three principles are the foundation for customer and organizational engagement. Today AI is like a super smart magic eight ball. Tomorrow AI supports and creates a dialog between companies and customers, managers and employees, and business to business.

We're now seeing the emergence of **cognitive computing**, a new era of computing systems that will understand the world in the way that humans do: through senses, learning, and experience.

These new cognitive systems will help us think. They will relate to us and answer questions using the language we humans use every day. They will learn from their interactions with data and with us, basically adapting their behavior automatically based on new knowledge.

That's what makes this third major era of computing such a huge leap forward. The first era was made up of tabulating machines and the second of programmable computers. While the programmable era will continue perhaps indefinitely and certainly underpin the next era of computing, cognitive systems represent a whole new approach to solving complex data and information analysis problems that goes beyond just computing.

Data is available everywhere, all the time. It's piling up, simply waiting to be used. Which is why we need computing systems that we can interact with using human language, rather than programming language. We need computers that can dish up advice, rather than waiting for commands.

How will these systems work? IBM Watson, one of the first systems built as a cognitive computing system, applies deep analytics to text and other unstructured big data sources to pull meaning out of the data by using inference, probability, and reasoning to solve complex problems. Watson is a first step toward cognitive computing, expanding the reaches of human understanding by helping us quickly and efficiently sift through massive amounts of data, pinpointing the information and insights that are now trapped within these sources.

Watson does this by using hundreds of analytics, which provide it with capabilities such as natural language processing, text analysis, and knowledge representation and reasoning to make sense of huge amounts of complex information in split seconds, rank answers based on evidence and confidence, and learn from its mistakes. And, of course, this capability is deployed in the cloud and made available to applications as a cognitive service.

One of the first domains for Watson is healthcare. Cleveland Clinic is working to explore how Watson can be used to better leverage valuable information trapped in large electronic health records. Watson's analytics can sift through unstructured clinical notes in a patient's health record, reason over that information, and connect it with other structured information in the health record to produce summaries, deeper insights, and faster access to relevant information.

A new application of Watson, called WatsonPaths, is able to analyze complex medical scenarios and propose relationships and connections to possible diagnoses extracted from the underlying medical literature. Medical students can interact with WatsonPaths to both learn from Watson and teach Watson by grading Watson's recommendations.

"Right now the science of cognitive computing is in the formative stages," says IBM Research's Ton Engbersen. "To become machines that can learn, computers must be able to process sensory as well as transactional input, handle uncertainty, draw inferences from their experience, modify conclusions according to feedback, and interact with people in a natural, human-like way."

Watson is a first step, but it points to what will be possible and how the age of cognitive computing will transform how we work with computers and what we expect out of them, helping remake our industries, economies and societies.

Notes:

deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions in data by using architectures composed of multiple non-linear transformations.[

brute force programming - программирование методом "грубой силы", неэффективный с точки зрения расходования вычислительных ресурсов стиль программирования, решение "в лоб", когда программист полагается только на производительность компьютера, вместо того чтобы попытаться упростить задачу, - поэтому программы получаются громоздкими, тяжеловесными, неэлегантными. В ряде случаев такой подход оправдан, например, когда решение разовой задачи нужно получить любой ценой

magic eight ball также **mystic 8 ball**, **шар судьбы**, **шар вопросов и ответов**, **шар предсказаний** — игрушка, шуточный способ предсказывать будущее. Это шар, сделанный из пластмассы, обычно диаметром 10-11 см, внутри которого есть емкость с тёмной жидкостью, в которой плавает фигура с 20 поверхностями — икосаэдр, на которых нанесены ответы. Ответы (20 вариантов) нанесены в формате «да», «нет», «абсолютно точно», «плохие шансы», «вопрос не ясен», и т. д.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

- For those who started their careers in AI and left in disillusionment or data scientists today, the consensus is often that artificial intelligence is just a new fancy marketing term for good old predictive analytics.
- This is game changing, both in how organizations operate and strategize as well as the impact on customer experience. These three principles are the foundation for customer and organizational engagement.

- While the programmable era will continue perhaps indefinitely and certainly underpin the next era of computing, cognitive systems represent a whole new approach to solving complex data and information analysis problems that goes beyond just computing.
- Watson is a first step toward cognitive computing, expanding the reaches of human understanding by helping us quickly and efficiently sift through massive amounts of data, pinpointing the information and insights that are now trapped within these sources.

2. Answer the following questions:

1. What is the current situation in the field of AI?
2. What are the three basic principles of new AI?
3. Why is AI (in its current state) called "a super smart magic eight ball"?
4. How does Watson address complex problems?
5. Where can cognitive computing systems be applied?

3. Translate into English:

До сих пор все, что было в кибернетике и вычислительной технике, базировалось, так или иначе, на моделях фон Неймана и Тьюринга. Сегодня IBM Research исследует следующее поколение вычислительных систем — *когнитивных* — они, по сути, отходят от модели Тьюринга, которая говорит о том, что любое вычисление может быть представлено в виде бесконечной ленты ячеек, в каждой из которых находится одна простая команда.

Человеческий мозг так не работает. У нейронов, во-первых, гораздо больше связей; во-вторых, у нервной клетки больше состояний, чем 0 и 1. А в-третьих, и это самое важное, у традиционных кибернетических устройств есть 3 принципиально разные функции, разделенные между разными модулями. Одна функция — это память, где хранится информация, вторая функция — это устройство ввода-вывода и третья — это, собственно говоря, функция вычисления. Нейрон – устройство универсальное: он получает информацию, хранит и перерабатывает ее.

Таким образом, когнитивные машины, конечно, должны не заново создавать мозг (природа один раз уже это сделала), а на основе тех знаний о физических и химических процессах, которые происходят в чипе, попытаться воспроизвести этот единый, параллельный по природе, растянутый во времени процесс познания, мышления, восприятия и осмысления реальности, и на этом основании выдать решение.

Пока что живые системы гораздо более эффективны, прежде всего, энергетически. Суперкомпьютер Watson, обыгравший участников телевикторины Jeopardy!, потребляет 80 кВт энергии, а человеческий мозг — 20 Вт. То есть Watson в 4 тыс. раз менее энергоэффективен, чем мозг: на одинаковое по результату действие у него уходит гораздо больше энергии, чем у человека.

4. Give the summary of the text using the key terms.

Topics for essays (you might need additional information):

- The origins of AI
- Expert systems
- AI2
- Cognitive computing

ARTIFICIAL INTELLIGENCE: ROBOTICS AND A-LIFE

ROBOTICS

Read the following words and word combinations and use them for understanding and translation of the text:

intricate - мудреный, замысловатый, сложный

gears and cams - шестеренки и эксцентрики

tentative - пробный, экспериментальный

serf - крепостной

end effector - рабочий орган, захватное устройство, кольцевой захват

welding - сварка, сварочные работы

automatic guided vehicle - робокар, автоматически управляемая тележка

scouting - разведка, дозор

rover - самоходная машина

law enforcement - правоохранительные органы, охрана правопорядка

market penetration - внедрение (выход) на рынок

low-wage - низкооплачиваемый

hazardous materials - опасные вещества

viability - жизнеспособность, живучесть

swarm intelligence - роевой интеллект

swarm robotics - групповая робототехника

fault tolerance - отказоустойчивость

malfunction - сбой, неисправность

scalable - масштабируемый, размерно варьируемый

mapping - картографирование

senior lecturer - старший преподаватель

search-and-rescue - поисково-спасательный

contaminated environment - зараженная окружающая среда
envision - предвидеть, представлять себе, воображать

The idea of the automaton — the lifelike machine that performs intricate tasks by itself — is very old. Simple automatons were known to the ancient world. By the 18th century, royal courts were being entertained by intricate humanlike automatons that could play music, draw pictures, or dance. A little later came the "Turk," a chess-playing automaton that could beat most human players.

However, things are not always what they seem. The true automatons, controlled by gears and cams, could play only whatever actions had been designed into them. They could not be reprogrammed and did not respond to changes in their environment. The chess-playing automaton held a concealed human player. True robotics began in the mid-20th century and has continued to move between two poles: the pedestrian but useful industrial robots and the intriguing but tentative creations of the artificial intelligence laboratories.

Industrial Robots

In 1921, the Czech playwright Karel Capek wrote a play called *R.U.R.* or *Rossum's Universal Robots.* Robot is a Czech word that has been translated as work(er), serf, or slave. In the play the robots, which are built by factories to work in other factories, eventually revolt against their human masters. During the 1960s, real robots began to appear in factory settings.

An industrial robot is basically a movable arm that ends in a "hand" called an end effector. The arm and hand can be moved by some combination of hydraulic, pneumatic, electrical, or mechanical means. Typical applications include assembling parts, welding, and painting. The robot is programmed for a task either by giving it a detailed set of commands to move to, grasp, and manipulate objects, or by "training" the robot by moving its arm, hand, and effectors through the required motions, which are then stored in the robot's memory. The early industrial robots had very little ability to respond to variations in the environment, such as the "work piece" that the robot was supposed to grasp being slightly out of position. However, later models have more sophisticated sensors to enable them to adjust to variations and still accomplish the task.

Mobile Robots and Service Robots

Industrial robots work in an extremely restricted environment, so their world representation can be quite simple. However, robots that can move about in the environment have also been developed. Military programs have developed automatic guided vehicles (AGVs) with wheels or tracks, capable of navigating a battlefield and scouting or attacking the enemy. Space-going robots including the Sojourner Mars rover also have considerable onboard "intelligence," although their overall tasks are programmed by remote commands.

Indeed, the extent to which mobile robots are truly autonomous varies considerably. At one end is the "robot" that is steered and otherwise controlled by its human operator, such as law enforcement robots that can be sent into dangerous hostage situations.

Moving toward greater autonomy, we have the "service robots" that have begun to show up in some institutions such as hospitals and laboratories. These mobile robots are often used to deliver supplies. For example, the Help-Mate robot can travel around a hospital by itself, navigating using an internal map. It can even take an elevator to go to another floor.

Service robots have had only modest market penetration, however. They are relatively expensive and limited in function, and if relatively low-wage more versatile human labor is available, it is generally preferred. For now mobile robots and service robots are most likely to turn up in specialized applications in environments too dangerous for human workers, such as in the military, law enforcement, handling of hazardous materials, and so on.

Smart Robots

Robotics has always had great fascination for artificial intelligence researchers. After all, the ability to function convincingly in a real-world environment would go a long way toward demonstrating the viability of true artificial intelligence. Building a smart, more humanlike robot involves several interrelated challenges, all quite difficult. These include developing a system for seeing and interpreting the environment (computer vision) as well a way to represent the environment internally so as to be able to navigate around obstacles and perform tasks.

One of the earliest AI robots was "Shakey," built at the Stanford Research Institute (SRI) in 1969. Shakey could navigate only in a rather simplified environment. However, the "Stanford Cart," built by Hans Moravec in the late 1970s could navigate around the nearby campus without getting into too much trouble.

An innovative line of research began in the 1990s at MIT. Instead of a "top down" approach of programming robots with explicit logical rules, so-called behavior-based robotics works from the bottom up, coupling systems of sensors and actuators that each have their own simple rules, from which can emerge surprisingly complex behavior. The MIT "sociable robots" Cog and Kismet were able to explore the world and learn to interact with people in somewhat the way a human toddler might.

Swarm robotics is an approach to robotics that emphasizes many simple robots instead of a single complex robot. A robot swarm has much in common with an ant colony or swarm of bees. No individual in the group is very intelligent or complex, but combined, they can perform difficult tasks. Swarm robotics has been an experimental field, but many practical applications have been proposed.

A traditional robot often needs complex components and significant computer processing power to accomplish its assigned tasks. In swarm robotics, each robot is relatively simple and inexpensive. As a group, these simple machines cooperate to perform advanced tasks that otherwise would require a more powerful, more expensive robot.

Using many simple robots has other advantages as well. Robot swarms have high fault tolerance, meaning that they still will perform well if some of the individual units malfunction or are destroyed. Swarms also are scalable, so the size of the swarm can be increased or decreased as needed.

One use that researchers have demonstrated for swarm robotics is mapping. A single robot would constantly need to keep track of its location, remember where it had been and figure out how to avoid obstacles while still exploring the entire area. A swarm of robots could be programmed simply to avoid obstacles while keeping in contact with other members of the swarm. The data from all of the robots in the swarm is then combined into a single map.

Swarm robotics has been an emerging field, and it has presented unique challenges to researchers. Programming a swarm of robots is unlike other types of programming. The model of distributed computing — using many computers to work on a single large task — is somewhat similar. Unlike distributed computing, however, each individual in swarm-style robotics deals with unique stimuli. Each robot, for example, is in a different location at any given time.

Some approaches to swarm robotics use a control unit that coordinates other robots. Other approaches use techniques borrowed from nature to give the swarm itself a type of collective intelligence. Much of the current research in the field focuses on finding the most efficient way to use a swarm.

Swarm robotics is a concept that's buzzed around since the 1980s, but now the technology is starting to fly. The environmental applications being explored range from coral restoration and oil spill clean-ups to precision farming – even the creation of artificial bees to pollinate crops.

Dr Roderich Gross, senior lecturer in robotics and computational intelligence, explains the concept: "In a swarm system there is no single point of failure – if a unit fails, the whole system keeps on going. Wherever you have a very heavy load that a human cannot manipulate, using a swarm of robots to do the job would be very sensible. That could be in a factory, transporting boxes. Or it could be a search-and-rescue scenario – maybe a collapsed building and you need to remove a very heavy part, or working in contaminated environments."

Scientists and designers at Heriot-Watt University have been looking at using a swarm of "coral bots" to restore ocean habitats. Dr Lea-Anne Henry of the university's school of life sciences believes that swarm robotics can "revolutionise conservation". Agriculture is looking into the potential for using swarms too. Professor Simon Blackmore, head of engineering at Harper Adams University works on larger robots that can work in fleets, able to identify weeds and administer microdots of chemicals with the result of using 99.9% less herbicide than traditional methods. He believes that, though the technology may appear an expensive luxury, it may have a wider appeal than the latest generation of conventional farm machinery such as expensive tractors and harvesters.

Perhaps the most famous – and controversial – swarm project to date is Harvard University's "Robobees", aiming to find an artificial solution to pollination to address the current decline in the global bee population. Here the robotic swarm is attempting to replicate one of nature's greatest swarms. But even setting aside the ethics of attempting to replace nature's pollinators, the idea may remain impossible.

The problems of organizing a swarm haven’t kept people from imagining what swarm robotics could offer some day. Some scientists envision a swarm of very small microbots being used to explore other planets. Other proposed uses include search-and-rescue missions, mining and even firefighting. When used with nanobots — microscopic-size robots — swarm robotics could even be used in human medicine.

Future Applications

A true humanoid robot with the kind of capabilities written about by Isaac Asimov and other science fiction writers is not in sight yet. However, there are many interesting applications of robots that are being explored today. These include the use of remote robots for such tasks as performing surgery (telepresence) and the application of robotics principles to the design of better prosthetic arms and legs for humans (bionics). Farther afield is the possibility of creating artificial robotic "life" that can self-reproduce.

Notes:

The Turk, also known as the **Mechanical Turk** or **Automaton Chess Player** was a fake chess-playing machine constructed in the late 18th century. From 1770 until its destruction by fire in 1854, it was exhibited by various owners as an automaton, though it was exposed in the early 1820s as an elaborate hoax.

Karel Čapek (1890 – 1938) was a Czech writer of the early 20th century best known for his science fiction, including his novel *War with the Newts* and the play *R.U.R.* that introduced the word *robot*.

Sojourner was the Mars Pathfinder robotic Mars rover that landed on July 4, 1997 and explored Mars for around three months.

Shakey the robot was the first general-purpose mobile robot to be able to reason about its own actions. While other robots would have to be instructed on each individual step of completing a larger task, Shakey could analyze the command and break it down into basic chunks by itself. It was developed from approximately 1966 through 1972 at the Artificial Intelligence Center of Stanford Research Institute

MIT (Massachusetts Institute of Technology) is a private research university in Cambridge, Massachusetts, founded in 1861 in response to the increasing industrialization of the United States. The institute adopted apolytechnic university model and stressed laboratory instruction.

RoboBee is a tiny robot capable of tethered flight, developed by a research robotics team at Harvard University. The 3-centimeter (1.2 in) wingspan of RoboBee makes it the smallest man-made device modeled on an insect to achieve flight.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. True robotics began in the mid-20th century and has continued to move between two poles: the pedestrian but useful industrial robots and the intriguing but tentative creations of the artificial intelligence laboratories.
2. The early industrial robots had very little ability to respond to variations in the environment, such as the "work piece" that the robot was supposed to grasp being slightly out of position.
3. At one end is the "robot" that is steered and otherwise controlled by its human operator, such as law enforcement robots that can be sent into dangerous hostage situations.
4. Service robots have had only modest market penetration, however.
5. After all, the ability to function convincingly in a real-world environment would go a long way toward demonstrating the viability of true artificial intelligence.
6. Swarm robotics has been an emerging field, and it has presented unique challenges to researchers.
7. Other approaches use techniques borrowed from nature to give the swarm itself a type of collective intelligence.
8. Swarm robotics is a concept that's buzzed around since the 1980s, but now the technology is starting to fly.

9. He believes that, though the technology may appear an expensive luxury, it may have a wider appeal than the latest generation of conventional farm machinery such as expensive tractors and harvesters.
10. Perhaps the most famous – and controversial – swarm project to date is Harvard University's "Robobees", aiming to find an artificial solution to pollination to address the current decline in the global bee population.

2. Answer the following questions:

1. How was the term "robot" coined?
2. What are the limitations of industrial robots?
3. Where are mobile robots being used?
4. What approaches does the development of smart robots call for?
5. What are the advantages of swarm robotics over conventional approaches?
6. What are the major challenges posed by swarm robotics?
7. Where can swarm intelligence be of practical assistance?

3. Translate into English:

Проект TERMES, реализуемый в течение четырех лет исследовательской группой самоорганизующихся систем Гарвардского университета, в основе которого лежит моделирование поведения колонии термитов, имеет конечную цель в создании масштабируемой системы искусственного интеллекта, в основе которой лежат простейшие роботы, способные уже сейчас совместными усилиями строить башни, пирамиды и другие сооружения, возводя даже дополнительные элементы, позволяющие роботам подниматься выше.

Данный проект имеет абсолютно другой подход к организации работ, нежели традиционная иерархическая система, в которой основной план движется, дробясь на множество мелких задач, от руководителей высшего звена через череду менеджеров и специалистов к непосредственным исполнителям. Вместо этого, модель колонии термитов предусматривает выполнение работ каждым роботом обособленно, без всякого централизованного руководства. Исследователи объясняют, что роботы действуют при помощи принципа стигмергии (stigmergy), принципа неявных коммуникаций, когда каждый индивидуум распознает изменения окружающей его среды и корректирует свои собственные планы в соответствии с этими изменениями.

Благодаря использованию принципа стигмергии, роботы TERMES могут работать группами от нескольких экземпляров до нескольких тысяч, выполняя единую задачу, но абсолютно не общаясь друг с другом. Отсутствие централизованного управления означает, что у системы в целом имеется крайне высокий уровень надежности, выход из строя одного экземпляра робота не приводит к неработоспособности системы, а оставшиеся работы продолжают работу, не замечая этого факта. Такой подход позволяет сделать роботов максимально простыми, ведь им не требуется наличия радио- или другого коммуникационного канала, работающего на иных принципах. Роботы TERMES, созданные гарвардскими исследователями, имеют всего по четыре датчика, по три независимых привода и несложный механизм, позволяющий брать, переносить и укладывать строительные блоки.

4. Give the summary of the text using the key terms.

ARTIFICIAL LIFE

Read the following words and word combinations and use them for understanding and translation of the text:

manifold - разнообразный, многообразный
blanket term - общий термин
design space - пространство проектных решений (параметров)
generalize - обобщать
to conceive - задумать, замыслить, разработать
typified - на примере
crossover - кроссинговер (перекрест хромосом)
to intervene - вмешиваться
full-blown design - полнофункциональная модель (образец)
laypeople - непрофессионалы
carbon chemistry - химия углеродных соединений
species of prey - хищный вид
predator - хищник
to validate - подтверждать, проверять правильность
conversely - наоборот, напротив, с другой стороны
to remedy - лечить, исправлять
commitment to the idea - приверженность идее
autopoiesis - самосоздание, самовоспроизводство

The historical and theoretical roots of the field are manifold. These roots include:

- early attempts to imitate the behavior of humans and animals by the invention of mechanical automata in the sixteenth century;
- cybernetics as the study of general principles of informational control in machines and animals;
- computer science as theory and the idea of abstract equivalence between various ways to express the notion

of computation, including physical instantiations of systems performing computations;

- John von Neumann's so-called self-reproducing Cellular Automata;
- computer science as a set of technical practices and computational architectures;
- artificial intelligence (AI)
- robotics;
- philosophy and system science notions of levels of organization, hierarchies, and emergence of new properties;
- non-linear science, such as the physics of complex systems and chaos theory; theoretical biology, including abstract theories of life processes; and
- evolutionary biology.

Artificial life is a blanket term used to refer to human attempts at setting up systems with lifelike properties all biological organisms possess, such as self-reproduction, homeostasis, adaptability, mutational variation, optimization of external states, and so on. The term is commonly associated with computer simulation-based artificial life, preferred heavily to robotics because of its ease of reprogramming, inexpensive hardware, and greater design space to explore. Artificial life projects can be thought of as attempts to generalize the phenomenon of life, asking questions like, "what would life have looked like if it evolved under radically different physical conditions?", "what is the logical form of all living systems?", or "what is the simplest possible living system?"

The term "artificial life", often shortened to "alife" or "A-Life", was coined in the late 1980s by researcher Christopher Langton, who defined it as "the study of artificial systems that exhibit behavior characteristic of natural living systems. It is the quest to explain life in any of its possible manifestations, without restriction to the particular examples that have evolved on earth... the ultimate goal is to extract the logical form of living systems."

Probably the first person to actively study and write on topics related to A-Life was the noted mathematician John Von Neumann, who was also an early figure in the field of game theory. In the middle of the 20th century, Von Neumann delivered a paper entitled "The General and Logical Theory of Automata," in which he discussed the concept of a machine that follows simple rules and reacts to information in its environment. Von Neumann proposed that living organisms are just such machines. He also studied the concept of machine self-replication, and conceived the idea that a self-replicating machine, or organism, must contain within itself a list of instructions for producing a copy of itself. This was several years before James Watson and Francis Crick, with the help of Rosalind Franklin and Maurice Wilkins, discovered the structure of DNA.

The field was expanded by the development of cellular automata as typified in John Conway's Game of Life in the 1970s, which demonstrated how simple components interacting according to a few specific rules could generate complex emergent patterns. This principle is used to model the flocking behavior of simulated birds, called "boids".

The development of genetic algorithms by John Holland added selection and evolution to the act of reproduction. This approach typically involves the setting up of numerous small programs with slightly varying code, and having them attempt a task such as sorting data or recognizing patterns. Those programs that prove most "fit" at accomplishing the task are allowed to survive and reproduce. In the act of reproduction, biological mechanisms such as genetic mutation and crossover are allowed to intervene. A rather similar approach is found in the neural network, where those nodes that succeed better at the task are given greater "weight" in creating a composite solution to the problem.

A more challenging but interesting approach to AL is to create actual robotic "organisms" that navigate in the physical rather than the virtual world. Roboticist Hans Moravec of the Stanford AI Laboratory and other researchers have built robots that can deal with unexpected obstacles by improvisation, much as people do, thanks to layers of software that process perceptions, fit them to a model of the world, and make plans based on goals. But such robots, built as full-blown designs, share few of the characteristics of artificial life. As with AI, the bottom-up approach offers a different strategy that has been called "fast, cheap, and out of control"—the production of numerous small, simple, insectlike robots that have only simple behaviors, but are potentially capable of interacting in surprising ways. If a meaningful genetic and reproductive mechanism can be included in such robots, the result would be much closer to true artificial life.

Artificial life is still a very new discipline, having been founded only in the late 1980s, and is still very much under development. Like other new fields, it has been the subject of some criticism. Based on its abstract nature, artificial life has taken time to be understood and accepted by the mainstream; papers on the topic have only recently been put into prominent scientific publications like *Nature* and *Science*. As with any new discipline, researchers need time to select the most fruitful research paths and translate their findings into terms other scientists and laypeople can understand and appreciate. The field of artificial life is one that seems poised to grow as the cost of computing power continues to drop.

Artificial life may be labeled **software**, **hardware**, or **wetware**, depending on the type of media researchers work with.

Software artificial life_is rooted in computer science and represents the idea that life is characterized by form, or forms of organization, rather than by its constituent material. Thus, "life" may be realized in some form (or media) other than carbon chemistry, such as in a computer's central processing unit, or in a network of computers, or as computer viruses spreading through the Internet. One can build a virtual ecosystem and let small component programs represent species of prey and predator organisms competing or cooperating for resources like food.

The difference between this type of artificial life and ordinary scientific use of computer simulations is that, with the latter, the researcher attempts to create a model of a real biological system (e.g., fish populations of the Atlantic Ocean) and to base the description upon real data and established biological principles. The researcher tries to validate the model to make sure that it represents aspects of the real world. Conversely, an artificial life model represents biology in a more abstract sense; it is not a real system, but a virtual one, constructed for a specific purpose, such as investigating the efficiency of an evolutionary process of a Lamarckian type (based upon the inheritance of acquired characters) as opposed to Darwinian evolution (based upon natural selection among randomly produced variants). Such a biological system may not exist anywhere in the real universe. As Langton emphasized, artificial life investigates "the biology of the possible" to remedy one of the inadequacies of traditional biology, which is bound to investigate how life actually evolved on Earth, but cannot describe the borders between possible and impossible forms of biological processes. For example, an artificial life system might be used to determine whether it is only by historical accident that organisms on Earth have the universal genetic code that they have, or whether the code could have been different.

It has been much debated whether virtual life in computers is nothing but a model on a higher level of abstraction, or whether it is a form of genuine life, as some artificial life researchers maintain. In its computational version, this claim implies a form of Platonism whereby life is regarded as a radically medium-independent form of existence similar to futuristic scenarios of disembodied forms of cognition and AI that may be downloaded to robots. In this debate, classical philosophical issues about dualism, monism, materialism, and the nature of information are at stake, and there is no clear-cut demarcation between science, metaphysics, and issues of religion and ethics.

Hardware artificial life refers to small animal-like robots, usually called animats, that researchers build and use to study the design principles of autonomous systems or agents. The functionality of an agent (a collection of modules, each with its own domain of interaction or competence) is an emergent property of the intensive interaction of the system with its dynamic environment. The modules operate quasi-autonomously and are solely responsible for the sensing, modeling, computing or reasoning, and motor control that is necessary to achieve their specific competence. Direct coupling of perception to action is facilitated by the use of reasoning methods, which operate on representations that are close to the information of the sensors.

This approach states that to build a system that is intelligent it is necessary to have its representations grounded in the physical world. Representations do not need to be explicit and stable, but must be situated and "embodied." The robots are thus situated in a world; they do not deal with abstract descriptions, but with the environment that directly influences the behavior of the system. In addition, the robots have "bodies" and experience the world directly, so that their actions have an immediate feedback upon the robot's own sensations. Computer-simulated robots, on the other hand, may be "situated" in a virtual environment, but they are not embodied. Hardware artificial life has many industrial and military technological applications.

Wetware artificial life comes closest to real biology. The scientific approach involves conducting experiments with populations of real organic macromolecules (combined in a liquid medium) in order to study their emergent self-organizing properties. An example is the artificial evolution of ribonucleic acid molecules (RNA) with specific catalytic properties. (This research may be useful in a medical context or may help shed light on the origin of life on Earth.) Research into RNA and similar scientific programs, however, often take place in the areas of molecular biology, biochemistry and combinatorial chemistry, and other carbon-based chemistries. Such wetware research does not necessarily have a commitment to the idea, often assumed by researchers in software artificial life, that life is a composed of medium-in-dependent forms of existence.

Thus wetware artificial life is concerned with the study of self-organizing principles in "real chemistries." In theoretical biology, autopoiesis is a term for the specific kind of self-maintenance produced by networks of components producing their own components and the boundaries of the network in processes that resemble organizationally closed loops. Such systems have been created artificially by chemical components not known in living organisms.

The philosophical implications arising from the possible development of true artificial life are similar to those involved with “strong AI.” Human beings are used to viewing themselves as the pinnacle of a hierarchy of intelligence and creativity. However, artificial life with the capability of rapid evolution might quickly outstrip human capabilities, perhaps leading to a world like that portrayed by science fiction writers where flesh-and-blood humans become a marginalized remnant population.

Notes:

Cellular Automaton is a collection of "colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. The rules are then applied iteratively for as many time steps as desired.

homeostasis is the ability to maintain a constant internal environment in response to environmental changes.

DNA or deoxyribonucleic acid is the hereditary material in humans and almost all other organisms that encodes the genetic instructions used in the development and functioning of all known living organisms and many viruses.

Game of Life, also known simply as **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. The "game" is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

Hans Moravec (born November 30, 1948, Kautzen, Austria) is an adjunct faculty member at the Robotics Institute of Carnegie Mellon University. He is known for his work on robotics, artificial intelligence, and writings on the impact of technology. Moravec also is a futurist with many of his publications and predictions focusing on transhumanism. Moravec developed techniques in computer vision for determining the region of interest (ROI) in a scene.

animats are artificial animals, a contraction of animal-materials. The term includes physical robots and virtual simulations.

ribonucleic acid (RNA) is a ubiquitous family of large biological molecules that perform multiple vital roles in the coding, decoding, regulation, and expression of genes. Together with DNA, RNA comprises the nucleic acids, which, along with proteins, constitute the three major macromolecules essential for all known forms of life.

autopoiesis (from Greek αὐτο- *(auto-)*, meaning "self", and ποίησις *(poiesis)*, meaning "creation, production") refers to a system capable of reproducing and maintaining itself.

Assignments

1. Translate the sentences from the texts into Russian in writing paying attention to the underlined words and phrases:

1. The term is commonly associated with computer simulation-based artificial life, preferred heavily to robotics because of its ease of reprogramming, inexpensive hardware, and greater design space to explore.
2. It is the quest to explain life in any of its possible manifestations, without restriction to the particular examples that have evolved on earth... the ultimate goal is to extract the logical form of living systems.
3. This principle is used to model the flocking behavior of simulated birds, called “boids”.
4. This approach typically involves the setting up of numerous small programs with slightly varying code, and having them attempt a task such as sorting data or recognizing patterns.
5. As Langton emphasized, artificial life investigates "the biology of the possible" to remedy one of the inadequacies of traditional biology, which is bound to investigate how life actually evolved on Earth, but cannot describe the borders between possible and impossible forms of biological processes.
6. The functionality of an agent (a collection of modules, each with its own domain of interaction or competence) is an emergent property of the intensive interaction of the system with its dynamic environment.
7. This approach states that to build a system that is intelligent it is necessary to have its representations grounded in the physical world.

2. Answer the following questions:

1. What are the origins of A-life as a discipline?
2. What questions are believed to be central to the field of A-life?
3. What does the concept of Cellular Automata involve?

4. Which of the three types of A-life seems to be most promising?
5. What are the distinguishing features of each type?
6. What kind of ethic issues might arise concerning A-life?

3. Translate into English:

Искусственная жизнь создана! Возможно ли такое?

24 мая 2010 года на пресс-конференции известный и талантливый американский биолог и бизнесмен Вентер, первый в мире расшифровавший геном человека, объявил общественности, что под его руководством институтом его же имени создана искусственная жизнь.

Впервые в истории создана искусственная живая клетка, которая всецело управляется рукотворным геномом. Ранее ученые лишь редактировали ДНК по кусочкам, получая генномодифицированные растения и животных.

Это достижение, несомненно, подогреет споры об этичности создания искусственной жизни, а также о юридически-правовых моментах и общественной опасности таких работ. "Это поворотный момент в отношениях человека с природой: впервые создана целая искусственная клетка с заранее заданными свойствами", - пояснил молекулярный биолог Ричард Эбрайт из Университета Рутджерса. По мнению экспертов, вскоре метод будет использоваться в коммерческих целях: некоторые компании уже разрабатывают живые организмы, способные синтезировать топливо, вакцины и др. Компания Synthetic Genomics Inc., основанная Вентером, заключила контракт на 600 млн. долларов на разработку водорослей, способных поглощать углекислый газ и производить топливо.

Ученые фактически претворили компьютерную программу в новое живое существо. Взяв за основу одну из бактерий, они внесли в компьютер полную расшифровку ее генома, заменили некоторые фрагменты в этом "тексте" своими собственными "сочинениями" и получили модифицированный вариант бактерии другого реально существующего вида. "Мы изготавливаем геном из четырех пузырьков химикатов, вносим искусственный геном в клетку, и наш искусственный геном подчиняет клетку себе", - разъяснил один из руководителей проекта Дэниел Гибсон. Чтобы обособить эту новую бактерию и всех ее потомков от творений природы, Вентер и его коллеги вставили в геном свои имена, а также три цитаты из Джеймса Джойса и других авторов. Эти "генетические водяные знаки" помогут ученым предъявить право собственности на клетки.

Topics for essays (you might need additional information):

- Early automatons
- Famous robotics projects
- Swarm intelligence: pros and cons
- Chemically Synthesized Genome

FUTURE COMPUTING

QUANTUM COMPUTING

Read the following words and word combinations and use them for understanding and translation of the text:

property - свойство, качество
quantum - квант. квантовый
spin - вращение
superposition - суперпозиция, наложение, совмещение
to flesh out - конкретизировать, изложить в деталях
to spur - побуждать, стимулировать
in part - частично
to outline - намечать, изложить вкратце
to factor - факторизовать, разложить (на множители)
integer - целое число
to be of great interest (to) - представлять большой интерес (для)
to tackle - заниматься
entanglement - перепутывание (квантовых состояний)
to crack - раскалывать(ся), ломаться
civilian - гражданский

The fundamental basis of electronic digital computing is the ability to store a binary value (1 or 0) using an electromagnetic property such as electrical charge or magnetic field.

However, during the first part of the 20th century, physicists discovered the laws of quantum mechanics that apply to the behavior of subatomic particles. An electron or photon, for example, can be said to be in any one of several "quantum states" depending on such characteristics as spin. In 1981, physicist Richard Feynman came up with the provocative idea that if quantum properties could be "read" and set, a computer could use an electron, photon, or other particle to store not just a single 1 or 0, but a number of values simultaneously. This ability of a quantum system to be in multiple states at the same time is called *superposition*. The simplest case, storing two values at once, is called a “qubit” (short for “quantum bit”). In 1985, David Deutsch at Oxford University fleshed out Feynman’s ideas by creating an actual design for a “quantum computer”, including an algorithm to be run on it.

At the time of Feynman’s proposal, the techniques for manipulating individual atoms or even particles had not yet been developed, so a practical quantum computer could not be built. However, during the 1990s, considerable progress was made, spurred in part by the suggestion of Bell Labs researcher Peter Shor, who outlined a quantum algorithm that might be used for rapid factoring of extremely large integers. Since the security of modern public key cryptography depends on the difficulty of such factoring, a working quantum computer would be of great interest to spy agencies.

The reason for the tremendous potential power of quantum computing is that if each qubit can store two values simultaneously, a register with three qubits can store eight values, and in general, for n qubits one can operate on 2^n values simultaneously. This means that a single quantum processor might be the equivalent of a huge number of separate processors. Clearly many problems that have been considered not practical to solve might be tackled with quantum computers.

Quantum computers also utilize another aspect of quantum mechanics known as *entanglement.* Unfortunately, quantum particles cannot be observed without being altered. Scientists use their knowledge of entanglement to indirectly observe the value of a qubit. When two subatomic particles become entangled, one particle adopts the properties of the other. Without looking at the qubit itself, scientists can read its value by observing the behavior of a particle with which it is entangled.

There are many potential applications for quantum computing. While the technology could be used to crack conventional cryptographic keys, researchers have suggested that it could also be used to generate unbreakable keys that depend on the “entanglement” of observers and what they observe. The sheer computational power of a quantum computer might make it possible to develop much better computer models of complex phenomena such as weather, climate, and the economy – or of quantum behavior itself.

As of 2014 quantum computing is still in its infancy but experiments have been carried out in which quantum computational operations were executed on a very small number of qubits. Both practical and theoretical research continues, and many national governments and military funding agencies support quantum computing research to develop quantum computers for both civilian and national security purposes, such as cryptanalysis.

Notes:

Bell Labs (Bell Laboratories) - бывшая американская, а ныне франко-американская корпорация, крупный исследовательский центр в области телекоммуникаций, электронных и компьютерных систем. Штаб-квартира Bell Labs расположена в Мюррей Хилл (Нью-Джерси, США)

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. However, during the 1990s, considerable progress was made, spurred in part by the suggestion of Bell Labs researcher Peter Shor, who outlined a quantum

algorithm that might be used for rapid factoring of extremely large integers.

2. Since the security of modern public key cryptography depends on the difficulty of such factoring, a working quantum computer would be of great interest to spy agencies.
3. Unfortunately, quantum particles cannot be observed without being altered.
4. As of 2014 quantum computing is still in its infancy but experiments have been carried out in which quantum computational operations were executed on a very small number of qubits.
5. Both practical and theoretical research continues.

2. Answer the following questions:

1. What is the basis of electronic digital computing?
2. What provocative idea did physicist Richard Feynman come up with?
3. Why could a practical quantum computer not be built at the time of Feynman's proposal?
4. Describe the reason for a huge potential power of quantum computing.
5. What aspects of quantum mechanics do quantum computers utilize?
6. How can quantum computing be applied?

3. Translate into English:

Современные компьютерные чипы могут содержать до нескольких миллиардов транзисторов на одном квадратном сантиметре кремния, а в будущем подобные элементы не будут превышать размера молекулы. Устройства с такими чипами будут существенно отличаться от классических компьютеров. Это обусловлено тем, что принципы их работы будут основаны на квантовой механике, физических законах, объясняющих поведение атомов и субатомных частиц. Ученые надеются, что квантовые компьютеры смогут решать ряд специфических задач гораздо быстрее, чем их классические собратья.

В действительности создать квантовый компьютер непросто. Основные его элементы - атомы, фотоны или специально созданные микроструктуры, хранящие данные в так называемых кубитах (квантовых битах), особенность которых заключается в том, что они должны отвечать двум противоречивым требованиям. С одной стороны они должны быть достаточно изолированы от любых внешних воздействий, которые могут нарушить вычислительный процесс, а с другой - иметь возможность взаимодействовать с другими кубитами. Кроме того необходимо иметь возможность измерить окончательное состояние кубитов и отобразить результаты вычислений.

Ученые во всем мире используют несколько подходов для создания первых прототипов квантовых компьютеров.

4. Give the summary of the text using the key terms.

BIOINFORMATICS

Read the following words and word combinations and use them for understanding and translation of the text:

undertaking - предприятие, начинание, дело
inherently - по существу
gene - ген
intricate - сложный, запутанный
protein folding - сворачивание белков
unlikely - невероятный, малообещающий
to harness - использовать

simulation - моделирование
makeup - состав, структура, строение
advent - приход, появление
predator - хищник
to devise - разрабатывать, придумывать, изобретать
sophisticated - сложный
feasible- возможный, осуществимый
to probe - исследовать, прозондировать
to bridge the gap - устранить разрыв
emergent behavior - непредсказуемое поведение
to inspire - вдохновлять, воодушевлять

Broadly speaking, bioinformatics (and the related field of computational biology) is the application of mathematical and information-science techniques to biology. This undertaking is inherently difficult because a living organism represents such a complex interaction of chemical processes. As more has been learned about the genome of humans and other organisms, it has become increasingly clear that the ″programs″ represented by gene sequences are "interpreted" through complex interactions of genes and the environment. Given this complexity, the great strides that have been made in genetics and the detailed study of metabolic and other biological processes would have been impossible without advances in computing and computer science.

Application to genetics.
Since information in the form of DNA sequences is the heart of genetics, information science plays a key role in understanding its significance and expression. The sequences of genes that determine the makeup and behavior of organisms can be represented and manipulated as strings of symbols using, for example, indexing and search algorithms. It is thus natural that the advent of powerful computer workstations and automated lab equipment would lead to the automation of gene sequencing, comparing or determining the relationship

between corresponding sequences. The completion of the sequencing of the human genome well ahead of schedule was thus a triumph of computer science as well as biology.

From genes to protein.

Gene sequences are only half of many problems in biology. Computational techniques are also being increasingly applied to the analysis and simulation of the many intricate chemical steps that link genetic information to expression in the form of particular protein and its three-dimensional structure in the process known as protein folding. The development of better algorithms and more powerful computing architectures for such analysis can further speed up research, avoid wasteful "dead ends", and bring effective treatments for cancer and other serious diseases to market sooner. The unlikely platform of a Sony PlayStation 3 and its powerful processor has been harnessed to turn gamers' idle time to the processing of protein-folding data in the Folding@Home project.

Simulation

A variety of other types of biological computer simulation have been employed. Examples include the chemical components that are responsible for metabolic activity in organisms, the structure of the nervous system and the brain (neural network), and the interaction of multiple predators and food sources in an ecosystem. Simulations can also incorporate algorithms first devised by artificial intelligence researchers (genetic algorithms). Simulations are combined with sophisticated graphics to enable researchers to visualize structure. Visualization algorithms developed for biomedical research can also be applied to the development of advanced MRI and other scans for use in diagnosis and therapy.

A fruitful relationship

Bioinformatics has been one of the "hottest" areas in computing in recent years, often following trends in the broader "biotech" sector. This challenging field involves such diverse subjects as genetics, biochemistry, physiology, mathematics (structural and statistical), database analysis and search techniques, simulation, modeling, graphics and image analysis. Major projects often involve close cooperation between bioinformatics specialists and other researchers. Researchers must also consider how the availability of ever-increasing computing power might make previously impossible projects feasible.

The relationship between biology and computer science seems destined to be even more fruitful in coming years. As software tools allow researchers to probe ever more deeply into biological processes and to bridge the gap between physics, biochemistry, and the emergent behavior of the living organisms, understanding of those processes may in turn inspire the creation of new architectures and algorithms in areas such as artificial intelligence and robotics.

Notes:

DNA (Deoxyribonucleic acid) - дезоксирибонуклеиновая кислота (ДНК)- макромолекула, обеспечивающая хранение, передачу из поколения в поколение и реализацию генетической программы развития и функционирования живых организмов.

Folding@Home - проект распределенных вычислений для проведения компьютерного моделирования свертывания молекул белка

MRI (Magnetic Resonance Imaging) – магнитно-резонансная томография

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases.

1. Broadly speaking, bioinformatics (and the related field of computational biology) is the application of mathematical and information-science techniques to biology.
2. Given this complexity, the great strides that have been made in genetics and the detailed study of metabolic and other biological processes would have been impossible without advances in computing and computer science.
3. The completion of the sequencing of the human genome well ahead of schedule was thus a triumph of computer science as well as biology.
4. Understanding of those processes may in turn inspire the creation of new architectures and algorithms in areas such as artificial intelligence and robotics.
5. The relationship between biology and computer science seems destined to be even more fruitful in coming years.

2. Answer the following questions:

1. What is bioinformatics?
2. How did advances in computing and computer science affect the development of genetics?
3. Why is information science so important for understanding genetics?
4. Describe the applications of computational techniques in the research of genetics.
5. How is the relationship between biology and computer science supposed to develop in future?

3. Translate into English:

В настоящее время, когда каждый новый шаг в совершенствовании полупроводниковых технологий дается со все большим трудом, ученые ищут альтернативные возможности развития вычислительных систем.

Естественный интерес ряда исследовательских групп вызвали природные способы хранения и обработки информации в биологических системах. Итогом их изысканий явился гибрид информационных и молекулярных технологий и биохимии - биокомпьютер.

Потенциал биокомпьютеров очень велик. По сравнению с обычными вычислительными устройствами они имеют ряд уникальных особенностей. Во-первых, они используют не бинарный, а тернарный код (так как информация в них кодируется тройками нуклеотидов). Во-вторых, поскольку вычисления производятся путем одновременного вступления в реакцию триллионов молекул ДНК, они могут выполнять до 1014 операций в секунду. В-третьих, вычислительные устройства на основе хранят данные с плотностью, в триллионы раз превышающей показатели оптических дисков. И наконец, ДНК-компьютеры имеют исключительно низкое энергопотребление.

Другим перспективным направлением замены полупроводниковых компьютеров является создание клеточных (бактериальных) компьютеров. Они представляют собой самоорганизующиеся колонии различных «умных» микроорганизмов. С помощью клеточных компьютеров станет возможным непосредственное объединение информационной технологии и биотехнологии.

Биокомпьютеры не рассчитаны на широкие массы пользователей. Но ученые надеются, что они найдут свое место в медицине и фармации. Глава израильской исследовательской группы профессор Эхуд Шапиро уверен, что в перспективе ДНК-наномашины смогут взаимодействовать с клетками человека, осуществлять наблюдение за потенциальными болезнетворными изменениями и синтезировать лекарства для борьбы с ними.

4. Give the summary of the text using the key terms.

NANOTECHNOLOGY

Read the following words and word combinations and use them for understanding and translation of the text:

to space - расставлять
precisely - точно, в точности
implication - значение, роль
density - плотность
to dissipate - рассеивать
to overcome - преодолевать
dormant - дремлющий, находящийся в состоянии покоя
branch - ответвление
assembly - сборка, монтаж
replication - копирование. репродукция
to deposit - поместить. помещать
nanotube - нанотрубка
conductor - проводник
to shrink - сокращаться
counterpart - двойник, аналог
core - ядро, сердечник, стержень
ultimate - конечный, окончательный

In a talk given in 1959, physicist Richard Feynman suggested that it might be possible to manipulate atoms individually, spacing them precisely. As Feynman also pointed out, the implications of computer technology are potentially very impressive. A current commercial DIMM memory module about the size of a person's little finger holds about 250 megabytes worth of data. Feynman calculated that if 100 precisely arranged atoms were used for each bit of information, the contents of all the books that have ever been written could be stored in a cube about 1/200 of an inch wide, just about the smallest object the unaided human eye can see. Further, although the density of computer logic circuits in microprocessors is millions of times greater than it was with the computers of 1959, computers built at the atomic scale would be billions of times smaller still. Indeed, they would be the smallest computers possible short of one that used quantum states within the atoms themselves to store information. “Nanocomputers” could also efficiently dissipate heat energy, overcoming a key problem with today’s increasingly dense microprocessors.

The idea of atomic-level engineering lay largely dormant for about two decades. Starting with a 1981 paper, however, K. Eric Drexler began to flesh out proposed structures and methods for a branch of engineering he termed *nanotechnology.* (The “nano” refers to a nanometer, or one billionth of a meter. The term “nanotechnology had been coined by the Tokyo Science University Professor Norio Taniguchi in 1974, and Drexler unknowingly used a related term to describe what later became known as molecular nanotechnology). Research in nanotechnology focuses on two broad areas: assembly and replication. Assembly is the problem of building tools (called assemblers) that can deposit and position individual atoms. Since such tools would almost certainly be prohibitively expensive to manufacture individually, research has focused on the idea of making tools that can reproduce themselves. This area of research began with John von Neumann’s 1940s concept of self-replicating computers.

There are several potential applications of nanotechnology in the manufacture of computer components. One is the possible use of carbon nanotubes in place of copper wires as conductors in computer chips. As chips continue to shrink, the connectors have also had to get smaller, but this in turn increases electrical resistance and reduces efficiency. Nanotubes, however, are not only superb electrical conductors, they are also far thinner than their copper counterparts. Intel Corporation has conducted promising tests of nanotube conductors, but it will likely be a number of years before they can be manufactured on an industrial scale.

Another alternative is "nanowires". One design consists of a germanium core surrounded by a thin layer of crystalline silicon. Nanowires are easier to manufacture than nanotubes, but their performance and other characteristics may make them less useful for general-purpose computing devices.

The ultimate goal is to make the actual transistors in computer chips out of nanotubes instead of silicon. An important step in this direction was achieved in 2006 by IBM researchers who created a complete electronic circuit using a single carbon nanotube molecule.

Notes:

DIMM (Dual In-line Memory Module) - двусторонний модуль памяти

Intel Corporation - американская корпорация, производящая широкий спектр электронных устройств и компьютерных компонентов, включая микропроцессоры, наборы системной логики (чипсеты) и др. Штаб-квартира - в городе Санта-Клара (Калифорния, США).

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. A current commercial DIMM memory module about the size of a person's little finger holds about 250 megabytes worth of data.
2. Indeed, they would be the smallest computers possible short of one that used quantum states within the atoms themselves to store information.
3. The idea of atomic-level engineering lay largely dormant for about two decades.

4. Starting with a 1981 paper, however, K. Eric Drexler began to flesh out proposed structures and methods for a branch of engineering he termed *nanotechnology.*
5. One is the possible use of carbon nanotubes in place of copper wires as conductors in computer chips.
6. Intel Corporation has conducted promising tests of nanotube conductors, but it will likely be a number of years before they can be manufactured on an industrial scale.

2. Answer the following questions:

1. When was the idea to manipulate atoms individually first suggested?
2. What is nanotechnology?
3. Research in nanotechnology focuses on two broad areas. What are they?
4. Describe the potential applications of nanotechnology in manufacturing computer components.
5. What is the ultimate goal of nanotechnology in the field of manufacturing computers?

3. Translate into English:

В 1959 году Ричард Фейнман в своем выступлении предсказал, что в будущем, научившись манипулировать отдельными атомами, человечество сможет синтезировать все, что угодно. В 1981 году появился первый инструмент для манипуляции атомами - туннельный микроскоп, изобретенный учеными из IBM. Оказалось, что с помощью этого микроскопа можно не только «видеть» отдельные атомы, но и поднимать и перемещать их. Этим была продемонстрирована принципиальная возможность манипулировать атомами, а стало быть, непосредственно собирать из них, словно из кирпичиков, все, что угодно: любой предмет, любое вещество.

Нанотехнологии обычно делят на три направления:

- изготовление электронных схем, элементы которых состоят из нескольких атомов

- создание наномашин, т.е. механизмов и роботов размером с молекулу

- непосредственная манипуляция атомами и молекулами.

Благодаря стремительному прогрессу в таких технологиях, как оптика, нанолитография, механохимия и 3D прототипирование, нанореволюция может произойти уже в течение следующего десятилетия. Когда это случится, нанотехнологии окажут влияние практически на все области промышленности и общества.

4. Give the summary of the text using the key terms.

UBIQUITOUS COMPUTING

Read the following words and word combinations and use them for understanding and translation of the text:

discrete - дискретный
ubiquitous - повсеместный, вездесущий
pervasive - распространенный, всеобъемлющий
to communicate - сообщать, передавать
to tag - добавлять
to track - прослеживать
to predict - прогнозировать
tiny - крошечный
to embed - встраивать
to run out - кончаться, истощаться
ambient - окружающий
to attune (to) – настраивать (на)
dashboard - информационная панель
cue - сигнал, намек

seamlessly - легко, беспрепятственно, без проблем
wearable - носимый

Traditionally people have thought of computers as discrete devices used for specific purposes such as to send e-mail or browse the Web. However, many researchers and futurists are looking toward a new paradigm that is rapidly emerging. Ubiquitous (or pervasive) computing focuses not on individual computers and tasks but on a world where most objects (including furniture and appliances) have the ability to communicate information. Kevin Ashton, a British technologist who created a system for tagging and tracking objects using radio frequencies, has predicted a future where everything is connected to the internet via tiny computer chips embedded within, or as he called it "the Internet of things". A fridge is already available with an on-board computer, allowing it to know its contents, order food when you run out and even suggest suitable recipes, before setting the oven to the right cooking temperature. It is also currently possible to control an entire room- the thermostat, light switch, TV, stereo etc.- all from a tablet or smartphone using wirelessly connected chips in each of the controlled devices.

"The internet of things" can be viewed as the third phase in a process where the emphasis has gradually shifted from individual desktops (1980s) to the network and Internet (1990s) to mobile presence and the ambient environment.

Some examples of ubiquitous computing might include:

- picture frames that display pictures attuned to the user's activities
- "dashboard" devices that can be set to display changing information such as weather and stock quotes
- parking meters that can provide verbal directions to nearby attractions
- kiosks or other facilities to provide verbal cues to guide travelers, such as through airports

- home monitoring systems that can sense and deal with accidents or health emergencies.

Ubiquitous computing greatly increases the ability of people to seamlessly access information for their daily activities. But the fact that the user is in effect "embedded" in the network can also raise issues of privacy and the receiving of unwanted advertising or other information.

An early center of research in ubiquitous computing was Xerox PARC, famous for its development of graphical user interface. Today a major force is MIT, especially its Project Oxygen, which explores networks of embedded computers. This challenging research area brings together aspects of many other fields (artificial intelligence, distributed computing, psychology of computing, smart buildings and homes, touchscreen, user interface, and wearable computers).

Notes:

XeroxPARC (Xerox Palo Alto Research Center) - научно-исследовательский центр, основанный в 1970. В 2002 году PARC стал отдельной компанией (в собственности Xerox)

MIT - Massachusetts Institute of Technology

Project Oxygen- исследовательский проект MIT для разработки вездесущих вычислений

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. "The internet of things" <u>can be viewed as</u> the third phase in a process where the emphasis <u>has gradually shifted from</u> individual desktops (1980s) <u>to</u> the network and Internet (1990s) to mobile presence and the ambient environment.

2. But the fact that the user is in effect "embedded" in the network can also raise issues of privacy.

2. Answer the following questions:

1. What is ubiquitous computing?
2. A British technologist K. Ashton has predicted "the Internet of things". What does this mean?
3. Give examples of ubiquitous computing.
4. What are "the two sides of the coin" when using ubiquitous computing in daily life?
5. Who deals with the research in ubiquitous computing?

3. Translate into English:

Интернет вещей (The Internet of Things, IoT) - концепция вычислительной сети физических объектов («вещей»), оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой. Организация таких сетей способно перестроить экономические и общественные процессы, исключая из части действий и операций необходимость участия человека.

Идея Интернета вещей сама по себе очень проста. Представим, что все окружающие нас предметы и устройства (домашние приборы, одежда, продукты, автомобили, промышленное оборудование и др.) снабжены миниатюрными идентификационными и сенсорными устройствами. Тогда при наличии необходимых каналов связи с ними можно не только отслеживать эти объекты и их параметры в пространстве и во времени, но и управлять ими, а также включать информацию о них в общую «умную планету».

Концепция сформулирована в 1999году как осмысление перспектив широкого применения средств радиочастотной идентификации для взаимодействия физических объектов между собой и с внешним окружением. Внедрение практических решений для ее реализации начиная с 2010 года считается восходящим трендом в информационных технологиях, прежде всего, благодаря повсеместному распространению беспроводных сетей, появлению облачных вычислений, развитию технологии межмашинного взаимодействия и др.

4. Give the summary of the text using the key terms.

SUPERCOMPUTERS

Read the following words and word combinations and use them for understanding and translation of the text:

successive - последующий, преемственный
at a clock speed - с тактовой частотой
to take advantage of - воспользоваться
entire - целый, полный
the ultimate - в конечном итоге
formerly - раньше
to soak up - впитывать, поглощать
protein - белок
cost-effective - рентабельный
distributed - распределенный
to feature - показывать, изображать
synergistic processing element - ядро специального назначения
ad hoc- специальный, на данный случай
to parcel out - выделять, делить на части
extraterrestrial - внеземной
proximity - близость
mesh - сетка, ячейка
to rank - классифицировать

high-performance computing - высокопроизводительные вычисления
benchmark - отметка, стандарт, эталонный тест
to retain - сохранять. удерживать

The term *supercomputer* is not really an absolute term describing a unique type of computer. Rather it has been used through successive generations of computer design to describe the fastest, most powerful computers available at a given time. However, what makes these machines the fastest is usually their adoption of a new technology or computer architecture that later finds its way into standard computers.

The first supercomputer is generally considered to be the Control Data CDC 6600, designed by Seymour Cray in 1964. The speed of this machine came from its use of the new faster silicon (rather than germanium) transistors and its ability to run at a clock speed of 10 MHz (a speed that would be achieved by personal computers by the middle 1980s).

Cray then left CDC to form Cray Research. He designed *the Cray I* in 1976, the first of a highly successful series of supercomputers. The Cray I took advantage of a new technology: integrated circuits, and new architecture: vector processing, in which a single instruction can be applied to an entire series of numbers simultaneously. This innovation marked the use of parallel processing as one of the distinguishing features of supercomputers.

The next generation, the Cray X-MP carried parallelism further by incorporating multiple processors (the successor, Cray Y-MP, had 8 processors which together could perform a billion floating point operations per second (1 gigaflop).

Soon other companies (particularly the Japanese manufactures NEC and Fujitsu) entered the market. The number of processors in supercomputers increased to as many as 1,024, which can exceed 1 trillion floating-point operations per second (1 teraflop)

The ultimate in multiprocessing is the series of Connection Machines built by Thinking Machines Inc. (TMI) and designed by Daniel Hillis. These machines have up to 65,000 very simple processors that run simultaneously and can form connections dynamically, somewhat like the process in the human brain. These "massively parallel" machines are thus attractive for artificial intelligence research.

As the power of standard computers continuous to grow, applications that formerly required a multimillion-dollar supercomputer can now run on a desktop workstation.

On the other hand there are always applications that will soak up whatever computing power can be brought to bear on them. These include: analysis of new aircraft designs, weather and climate models, the study of nuclear reactions, and the creation of models for the synthesis of proteins.

For many applications it may be more cost-effective to build systems with numerous coordinated processors (a sort of successor to the 1980s Connection Machine). For example, the Beowolf architecture involves "clusters" of ordinary PCs coordinated by software running on UNIX or Linux. The use of free software and commodity PCs can make this approach attractive, though application software still has to be rewritten to run on the distributed processors.

A new resource for parallel supercomputing came from an unlikely place: the new generation of cell processors found in game consoles such as the Sony Playstation 3. This architecture features tight integration of a central "power processor element" with multiple "synergistic processing elements".

Finally, an ad hoc "supercomputer" can be created almost for free, using software that parcels out calculation tasks to thousands of computers participating via the Internet, as with SETI@Home (searching for extraterrestrial radio signals) and Folding@Home (for protein-folding analysis). In another approach, a large number of dedicated processors are placed in close proximity to each other (e.g. in a computer cluster); this saves considerable time moving data around and makes it possible for the processors to work together (rather than on separate tasks), for example in mesh and hypercube architecture.

The Top500 project ranks and details the 500 most powerful (non-distributed) computer systems in the world. The project was started in 1993 and publishes an updated list of the supercomputers twice a year. The project aims to provide a reliable basis for tracking and detecting trends in high-performance computing and bases rankings on HPL, a portable implementation of the high-performance LINPACK benchmark written in FORTRAN for distributed memory computers.

According to the 42nd edition (November, 2013) of the Top500 list of the world's most powerful supercomputers, Teanhe-2, a supercomputer developed by China's National University of Defense Technology, retained its position as the world's No.1 system with a performance of 33.86 petaflops/s (quadrillions of calculations per second).

Titan, a Cray XK7 system installed at the Department of Energy's (DOE) Oak Ridge National Laboratory, remains the No.2 system. It achieved 17.59 Pflops/s on the Linpack benchmark. Titan is one of the most energy-efficient systems on the list.

Notes:

NEC (Nippon Electric Corporation) - японская компания, производитель электронной, компьютерной техники

Fujitsu - крупная японская корпорация, производитель электроники

SETI@Home (Search for Extra-Terrestrial Intelligence at Home – поиск внеземного разума на дому) – научный некоммерческий проект добровольных вычислений на платформе BOINC, использующий свободные вычислительные ресурсы на компьютерах добровольцев для поиска радиосигналов внеземных цивилизаций

LINPACK benchmark - тест производительности вычислительных систем, по результатам которого составляется список 500 наиболее высокопроизводительных систем мира

Assignments

1. Translate the sentences from the text into Russian in writing paying attention to the underlined words and phrases:

1. Rather it has been used through successive generations of computer design to describe the fastest, most powerful computers available at a given time.
2. The Cray I took advantage of a new technology: integrated circuits, and new architecture: vector processing, in which a single instruction can be applied to an entire series of numbers simultaneously.
3. The next generation, the Cray X-MP carried parallelism further by incorporating multiple processors.
4. On the other hand there are always applications that will soak up whatever computing power can be brought to bear on them.
5. Finally, an ad hoc "supercomputer" can be created almost for free, using software that parcels out calculation tasks to thousands of computers participating via the Internet.

2. Answer the following questions:

1. What does the term *supercomputer* describe?
2. What is considered to be the first supercomputer?
3. Which innovation marked the use of parallel processing as one of the distinguishing features of supercomputers?
4. Describe resources for parallel "supercomputing".
5. What is Top500 project?

3. Translate into English:

«Элита компьютерного мира»

Суперкомпьютеры находятся на вершине своеобразной пирамиды мира вычислительной техники. Современные машины могут иметь до 100 тысяч процессоров и выполнять 60 000 млрд. операций в секунду. История суперкомпьютеров в СССР началась именно в МГУ. Первая машина «Стрела» была построена в 1956 году легендарным конструктором и основателем советской школы конструкторов вычислительной техники С.А. Лебедевым. «Стрела» выполняла 2000 операций в секунду и занимала 300 кв. м.

Следующая машина «БЭСМ-6», построенная в 1968 году, уже выполняла 1 млн. операций в секунду и являлась на тот момент одной из самых быстродействующих машин в мире.

В 2008 г. На базе НИВЦ МГУ был построен современный суперкомпьютер «Чебышев», названный в честь великого русского математика. Он занимает всего 100 кв. м., весит 30 тонн и способен выполнять уже тысячи миллиардов операций в секунду.

Суперкомпьютер «Ломоносов», построенный компанией «Т-Платформы» для МГУ им. М.В. Ломоносова,- первый гибридный суперкомпьютер такого масштаба в России и Восточной Европе. В нем используется три вида вычислительных узлов и процессоры с различной архитектурой. Предполагается использовать суперкомпьютер для решения ресурсоемких вычислительных задач в рамках фундаментальных научных исследований, а также для проведения научной работы в области разработки алгоритмов и программного обеспечения для мощных вычислительных систем.

Суперкомпьютеры решают огромное количество важных прикладных и фундаментальных задач. Одни из наиболее часто встречающихся - моделирование нефтяных резервуаров, проектирование жилищных застроек и новых материалов, проведение виртуальных краш-тестов в процессе конструирования автомобилей. В МГУ «Чебышев» занимается фундаментальной наукой. На нем, в частности, решается задача исследования магнитного поля Земли. Незаменимы суперкомпьютеры и в криптографии- науке о защите информации. Суперкомпьютер эффективно выполняет параллельные вычисления благодаря большому количеству самостоятельных микропроцессоров.

Не каждую задачу можно решить на суперкомпьютере: сложность задачи должна соответствовать сложности системы. Самым большим суперкомпьютером, как это ни странно, является сеть Интернет, которая соединяет огромное количество вычислительных мощностей по всему миру. В будущем, возможно, удастся скоординировать усилия и использовать невостребованные мощности такой суперсистемы.

Topics for essays (you might need additional information):

- Quantum computers versus traditional computers.
- Nanotechnology in our lives.
- The Internet of things may bring problems.
- Tiny biocomputers move closer to reality.
- Application of supercomputers.

LITERATURE

1. "3D Computer Graphics" by Alan H. Watt, Addison-Wesley, 2000
2. " A Brief History of Programming Languages." http://www.byte.com/art/9509/se7/artl9.htm. Cited, March 25, 2000.
3. "Artificial Intelligence - What You Really Need to Know" by Michele Goetz. Jan 30, 2014 http://www.forbes.com/sites/forrester/2014/01/29/artificial-intelligence-what-you-really-need-to-know/
4. "Artificial Life. Encyclopedia of Science and Religion" by Claus Emmeche, New York University Press, 2003. http://www.encyclopedia.com/topic/Artificial_life.aspx
5. "Best-Kept Secrets: Quantum Cryptography Has Marched from Theory to Laboratory to Real Products" by Stix, Gary. Scientific American, 2005. http://www.sciam.com/article.cfm?chanID=sa006&colID=1&articleID= 000479CD-F58C-11BE-AD0683414B7F0000
6. "Biometric Security Technology" by P Kumbargoudar, PeterIndia , 2008 www.peterindia.net/BiometricsView.html
7. "Cognitive Computing Ushers In New Era of IT" by Eric W. Brown, IBM Smarter Planet. 2/03/2014. http://www.forbes.com/sites/ibm/2014/02/03/cognitive-computing-ushers-in-new-era-of-it/
8. "Computer Graphics: Principles and Practice" by James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes. Addison-Wesley Professional, 1996.
9. "Computer Graphics: Theory into Practice" by Jeffrey J. McConnell. Jones and Bartlett Publishers, 2006.
10. "Computer Science Illuminated" by Nell Dale, John Lewis. Jones and Bartlett Publishers, 2002

11. “Concepts and Terminology for Computer Security” by Donald L. Brinkley and Roger R. Schell, 1995
12. “Concepts in Programming Languages” by John C. Mitchell. Cambridge University Press, 2003
13. “Encyclopedia of Computer Science and Technology” (Revised edition) by Harry Henderson. Infobase Publishing, 2009
14. “Flight of the Robobee: the Rise of Swarm Robotics” by Tim Smedley. http://www.theguardian.com/sustainable-business/swarm-robotics-conservation-coral-reefs-pollination
15. “From ENIAC to Everyone: Talking with J. Presper Eckert” by Alexander Randall. http://www.kurzweilai.net/articles/art0645.html
16. ‘Fuzzy Logic: Четкие решения нечеткой логики.” http://www.bacnet.ru/knowledge-base/articles/index.php?ELEMENT_ID=653
17. “History of Programming Languages-II” by Thomas J. Bergin and Richard G. Gibson. New York: ACM Press, 1996.
18. “How Quantum Cryptology Works” by Josh Clark. science.howstuffworks.com/.../quantum-cryptology.htm
19. http://albatron.ru/27-kompyuter-na-parovoj-tyage-ili-o-tvorenii-britanskix-uchenyx.html
20. http://gimn6.ru/article.asp?id_text=60
21. http://globalfuturist.com/about-igf/top-ten-trends/top-ten-computer-trends-for-the-21st-century.html
22. http://shkolazhizni.ru/archive/0/n-37499/
23. http://software-security.sans.org/resources/paper/cissp/overview-tutorial-artificial-intelligence-systems
24. http://thinkinnovative.ru/experts/blogs/id/78

25. http://www.dataved.ru/2013/01/slang-is-difficult-for-ibm.html
26. “Introduction to Artificial Intelligence” by Wolfgang Ertel. Springer-Verlag London Limited, 2011
27. “Introduction to Information Security” by Linda Pesante. Carnegie Mellon University, 2008
28. “The Art of Computer Programming: Semi-numerical Algorithms” by Donald E. Knuth. Addison-Wesley; 1981
29. Landofcode.com Webhostinggeeks.com Institut fur Theoretische Physik http://www.itp.uni-hannover.de/?lang=en
30. Megaspring.ucoz.ru http://www.ozon.ru/context/detail/id/1421843/
31. “Operating System Concepts” (8th edition) by Silberschats, Galvin, Gagne. John Wiley & Sons Inc., 2008
32. “Organization and Architecture. Designing for Performance” by William Stallings. Pearson Prentice Hall, 2010
33. “Past, Present, and Future Methods of Cryptography and Data Encryption” A Research Review by Nicholas G. McDonald. Department of Electrical and Computer Engineering, University of Utah
34. “Programming Languages”. McGraw-Hill Encyclopedia of Science and Technology. New York: McGraw-Hill, 1997.
35. “Quantum Cryptography” by Alves, Carolina Moura and Kent Adrian. National University of Singapore. http://www.quantumlah.org/?q=tutorial/quantumcrypto
36. “Quantum Cryptography Tutorial” Dartmouth College. http://www.cs.dartmouth.edu/~jford/crypto.html
37. “Quantum Cryptography: Privacy through Uncertainty” by Salvatore Vittorio. Proquest - CSA - October 2002. http://www.csa.com/discoveryguides/crypt/overview.php

38. “Quantum Programming Languages: Survey and Bibliography” by Simon J. Gay. Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK , 2006
39. “Security of the Internet” by James Ellis, Howard F. Lipson, Thomas A. Longstaff, Linda Pesante, Derek Simmel. NEWS AT SEI, 1998.
40. “Tales of the Encrypted” by Brodney A, Asher J. http://library.thinkquest.org/28005/flashed/index2.shtml.
41. “The Codebreakers: The Story of Secret Writing” by D. Kahn. Scribner, 1996.
42. “The ENIAC Story” by Martin H. Weik. Ordnance Ballistic Research Laboratories, Aberdeen Proving Ground, MD ftp.arl.mil/mike/comphist/eniac-story.html
43. to.prabc.ru
44. transhumanism-russia.ru
45. “Understanding Programming Languages” by M. Ben-Ari Weizmann. Institute of Science. Originally published by John Wiley & Sons, Chichester, 1996.
46. “Web Graphics for Dummies” by Linda Richards. Wiley/Dummies, 1997.
47. “What is Artificial Life?” http://www.wisegeek.com/what-is-artificial-life.htm
48. “What is Swarm Robotics? “ http://www.wisegeek.com/what-is-swarm-robotics.htm
49. www.book.kbsu.ru
50. www.ccas.ru
51. www.inf1.info/machineneumann
52. www.osp.ru
53. “Ионы для квантовых компьютеров.” Кристофер Монро, Дэвид Уайнленд, «В мире Науки», ноябрь, 2008
54. “Квантовая криптография.” Курсовая работа по

дисциплине: Криптография. Кафедра Информационной безопасности, Студент группы Зи91 Лазарев Ю.А. МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ (технический университет), МОСКВА 2003.

55. “Легенда о Сетуни” http://habrahabr.ru/post/46688/
56. “Московский университет”, №31, октябрь, 2008
57. “Основные понятия - Криптография” kriptografea.narod.ru/osnponiatia.html
58. “Первая ЭВМ” http://info61.blogspot.ru/p/20-1902-3-1974-1953-12.html
59. “Современные биометрические методы идентификации.” habrahabr.ru/post/126144/11 авг. 2011

CONTENTS