**Имитационное моделирование в исследовании и разработке информационных систем**

Бахмуров Анатолий Геннадиевич (bahmurov@lvk.cs.msu.su)

Все презентации будут выкладываться на сайт.

**Лекция 1 (была пробита за несодержательностью)**

**Лекция 2**

Программные наблюдатели:

В процессорах сейчас есть 4 регистра – hard breakpoint

Встроенные отладочные средства – JTAG (Joint Test Action Group) – разработали аппаратный интерфейс, к которому можно подключиться, чтобы заниматься отладкой микросхемы, все сколь-нибудь серьёзные микросхемы оснащаются этим интерфейсом, чтобы можно было подключиться для отладки.

Можно мерять аппаратуру осциллографом или логическим анализатором (записывает логический сигнал)

Про шину он тут рассказывал, странный дядька. Стандартов – прорва. (VME (для объединения многопроцессорных систем), AGP (видео карта), PCI(управление периферийными устройствами), MIL-STD-1553 (всего 2 линии, 1 мегабит) (для объединения устройств), FiberChannel (пара гигабит))

Шина работает очень быстро и потому. Чтобы её отлаживать, нужно специальные утилиты, которые будут записывать данные, пробегающие по шине. Например это нужно чтобы понять, почему какой-нибудь драйвер упал.

Есть несколько способов отображения – таблица данных (как пакеты в wireshark), временная диаграмма (в какой момент что случилось).

При указании параметров, чего нужно отлавливать – можно указать конечный автомат для указания множества для перехвата.

Можно всё время писать события, и можно сделать буфер скользящим, чтобы момент времени записи был посередине, в начале или в середине окна, чтобы можно было глянуть, что было до и после.

Трассы ещё можно сравнивать. И можно собирать интегральные характеристики – загруженность, время ожидания, время арбитража, …

Важная опция – имитация, т.е. программа может прикинуться каким-нибудь устройством. (программа просто записывает данные по диапазону адресов). Также можно и имитировать ошибки.

Начал рассказывать про linux:

Упомянул strace и ltrace.

Замер времени в ос общего назначения:

Рассказал про библиотеку time.h (gettimeofday, time, times, sleep, …)

Есть деление для подсчёта пользовательского и системного времени.

Вспомнил про команду time в unix, которая выдаст следующие времена – real, user, sys.

Команда rdtsc в x86 – программа, записывающая в пару регистров количество тактов процессора после включения питания.

Чтобы выяснить, что нужно делать, чтобы улучшить ситуацию, где именно тратиться время в программе – для этого используют профилировщики.

gprof – команда в unix (для успешной работы нужно в компиляторе поставить флаг -pg) создаст файл с результатами профилирования (с некоторой частотой (которая вообще говоря настраиваема), может выдать информацию о том, сколько работало, где висели, какие вызовы и сколько было, …)

Другая хорошая утилита – valgrind – он следить за утечками памяти, следит за временем выполнения, …

Вообще говоря valgrind – это некоторая общая среда для запуска разных утилит и проверок, например можно использовать cachegrind для того, чтобы проследить, как программа работает с кешем.

Callgrind – умеет строить графы вызовов.

**Домашняя работа:**

Нужно померять разными способами время выполнения. – подробнее смотри слайды.

Нужно сделать то же самое, но переставив индексы местами.

Результат нужно прислать (что меряли на каком процессоре, …) на почту bahmurov@lvk.cs.msu.su

По итогам будет разное количество вопросов по экзамену.

Приветствуется, если будет выполнено несколько замеров (под linux или ещё что-нибудь).

**Лекция 3**

Рассказывал про разные способы моделирования и их градации, см. слады.

На 10-м слайде он имитировал задачу с 9-го слайда, суть была в том, что если задачу усложнить и предположить, что мы можем промахиваться, то модель будет иметь смысл. (Кажется про теорвер он не слышал)

Концептуальная модель – это частично формализованное описание моделируемого объекта, у которого частично упрощаются многие нюансы.

**Лекция 4**

Рассказывал о подходах к организации моделей:

• Событийно-ориентированный

• Процессо-ориентированный

– Транзакты

– «собственно» процессы и нити (threads)

• Агентно-ориентированный

(развитие «собственно» процессного)

Лучше их просто погуглить

Упоминал GPSS и Anylogic

**Лекция 5**

<http://www.omnetpp.org/> - для анализа сетевых компонентов (это симулятор)

есть среда разработки, библиотека типовых элементов

Задача – нужно поставить себе систему моделирования (либо omnet++ (собирается из исходника (www.omnetpp.org)) или anylogic (есть пробная версия для студентов – бесплатно (anylogic.ru) (вероятно anylogic - проще)))

random – более сложный генератор чисел, чем rand

**Лекция 6**

Системы массового обслуживания