

Часть 1

Коллоквиум № 1.

Часть 1

Продолжение курса.

Поддержка пользователя:

- хорошие (и своевременные) *сообщения об ошибках*; не «назидательные»/грубые; с информацией о том, что нужно/можно сделать;
- хорошо структурированная и полная *справочная информация* и встроенные системные утилиты -«мастера» (подсистема помощи; мастер установки оборудования и др.);
- хорошо структурированная и полная (при этом – обозримая) *внешняя документация*.

Пример:

В книге [2, с. 197-200] приводится пример мучительного для пользователя диалога с системой электронной почты. Пользователь правильно ввел свой запрос только с 7 попытки.

При этом он ошибся 4 раза:

- простая ошибка ввода слова с клавиатуры (**haeders** вместо **headers**),
- второстепенная ошибка в наименовании (**intersection** вместо **intersect**),
- простая синтаксическая ошибка (пользователь не понял, что значения параметра следует заключать в кавычки),
- грубая синтаксическая ошибка (пользователь не понимает, что два ключевых параметра должны быть связаны логическим оператором).

Пользователь сразу же заметил первую ошибку, но при этом повторял ее несколько раз (когда терял сосредоточенность); он так и не заметил в сообщениях системы, что это слово можно было сократить до одной буквы (**h**).

Он сразу же исправил вторую и третью ошибки. Скорее из соображений здравого смысла, чем с помощью системы.

Исправить четвертую ошибку без помощи системы пользователь, скорее всего, не смог бы.

Вывод: Причина ошибок – диалог ведется на основе командного языка & поддержка пользователя в системе явно неудачна (недостаток системы).

Ошибки

Последствия ошибок:

- а) система выходит из строя;
- б) система обнаруживает невозможность дальнейшей обработки полученных данных и просит пользователя скорректировать их;
- с) система выполняет задачу, но результат отличается от того, который ожидал пользователь (например, пользователь по ошибке выполнил команду **Move** вместо команды **Copy**).

Основные причины ошибочных действий системы – неверный ввод информации пользователем.

В случае а) проверка входных данных на корректность не предохраняет систему от невыполнимых команд.

В случае б) такая проверка показывает неосуществимость требований пользователя.

В случае с) такая проверка неполна (например, проверяется формат входных данных, а правильность самих данных не проверяется).

Основные причины ошибок пользователей:

- неправильное понимание сути требования (например, неправильное понимание подсказки);
- неправильное действие пользователя (в том числе случайная ошибка, например, нажатие соседней клавиши);
- недоработки авторов системы:

в одной из ОС были такие команды:

copy <ToFilename> = <FromFilename> - копирование файла

ren <FromFilename> <ToFilename> - переименование файла.

Процесс ввода данных в общем случае не должен быть сложным. Избежать ошибок ввода могут помочь: опережающий ввод, автозаполнение форм, максимально широкое использование **меню**; минимизация объема клавиатурного ввода, минимизация действий "по умолчанию", запрос и ожидание подтверждения.

Диалог не должен способствовать появлению ошибок восприятия или понимания.

Все системы нуждаются в механизме проверки входных данных для прекращения ошибочного ввода и информирования пользователя об этом (а, в лучшем случае, исправления ошибки ввода, если это возможно и допустимо с учетом полномочий системы – автоматическая замена **haeders** на **headers**).

Классический пример плохого сообщения об ошибках:

В одной из известных и распространенных в свое время крупных компьютерных систем при сбое центрального процессора в верхней части экрана каждого терминала появлялось сообщение:

ОЖИДАЮ ФУНКЦИЮ ЧЕТЫРЕ

Еще один пример плохого сообщения об ошибках:

ДИСК ЗАПОЛНЕН – НАЖМИТЕ КЛАВИШУ ESCape

Пользователь взволнован: пропадут ли 10 страниц текста, который он перед этим вводил? Можно ли (и как) сохранить результаты ввода?

В подобных ситуациях более удачно сообщение такого вида:

ДИСК ЗАПОЛНЕН – НЕВОЗМОЖНО СОХРАНИТЬ ТЕКУЩИЙ ФАЙЛ

НАЖМИТЕ КЛАВИШУ ESCape И ВСТАВЬТЕ НОВЫЙ ДИСК ДЛЯ СОХРАНЕНИЯ ФАЙЛА

Сообщение об ошибке должно точно объяснить, в чем заключается ошибка и какие действия следует предпринять, чтобы ее устранить.

Введенные пользователем данные, которые расценены системой как некорректные, могут на самом деле быть правильными (появление незнакомого текстовому процессору слова). Можно пометить такие данные флажком **"новое"**, а в дальнейшем уточнить ситуацию (например, с помощью пользователя).

К проблеме субъективного аспекта правильности данных мы вернемся на Лекциях № 7 и № 9.

Справочная информация.

В диалогах, управляемых системой, справочная информация вызывается с помощью стандартного ответа (**Help** и т.п.) на любой запрос системы. В диалогах, управляемых пользователем, может быть предусмотрена специальная команда (типа **Help**); возможно, с уточняющими параметрами.

Справочная информация (как и сообщения об ошибках) должна быть: информативна;

своевременна (появляется на экране тогда и только тогда, когда она требуется);

обозрима;

сформулирована в понятных пользователю терминах;

доступна в любой момент;

отвечать контексту (соответствовать конкретной проблеме пользователя).

Целесообразно использовать иерархическую систему справочной информации.

Внешняя документация.

Помощь пользователю преследует три главные цели:

- описание возможностей и принципов работы программной системы;
- снабжение пользователя инструкцией по решению конкретной задачи;
- помощь в работе с системой и выдача справочной информации.

Вопрос – всю ли справочную информацию следует "встраивать" в систему или же часть ее можно оформить в виде внешнего документа?

Учитывать: пользователь не может эффективно использовать объемное руководство во время работы с системой.