

Содержание

| | |
|---|----|
| Лабораторная работа № 1. Конфигурирование операционной системы и использование командных файлов..... | 4 |
| 1.1. Цель лабораторной работы..... | 4 |
| 1.2. Содержание отчёта..... | 4 |
| 1.3. Краткие теоретические сведения..... | 4 |
| 1.4. Задание на выполнение..... | 16 |
| 1.5. Порядок выполнения работы..... | 16 |
| 1.6. Контрольные вопросы..... | 16 |
| Литература..... | 16 |

ЛАБОРАТОРНАЯ РАБОТА № 1

Конфигурирование операционной системы и использование командных файлов

1.1. Цель лабораторной работы.

Целью лабораторной работы является освоение:

- 1) структуры и состава командных файлов для разных типов ОС;
- 2) приемов работы с командными файлами;
- 3) вызова и настройки командной оболочки;
- 4) работы с глобальными и локальными переменными ОС.

1.2. Содержание отчёта.

Отчет по лабораторной работе должен включать:

- 1) цель лабораторной работы;
- 2) конкретный вариант задания на выполнение;
- 3) схемы алгоритмов командных файлов;
- 4) тексты командных файлов;
- 5) результаты выполнения командных файлов и команд оболочки.

1.3. Краткие теоретические сведения.

1.3.1. Функции операционной системы – пользовательский интерфейс.

Пользователь может обращаться к операционной системе посредством одного или нескольких пользовательских приложений, а также часто посредством специального приложения, называемого оболочкой (shell) или встроенным интерпретатором команд (командная оболочка), т.е. ОС обеспечивает интерфейс не только для прикладных программ, но и для пользователя.

Современные ОС поддерживают функции пользовательского интерфейса двух типов: текстовые и графические.

Графический интерфейс пользователя (Graphical User Interface – GUI) позволяет выполнять нужные действия с помощью мыши, однако достаточно часто в качестве альтернативы ОС дополнительно предлагает и текстовый интерфейс.

Командная оболочка – это интерфейс взаимодействия пользователя с компьютером, в котором команды отдаются путем ввода текстовых строк при помощи клавиатуры. История появления командной строки относится к временам компьютерных терминалов, которые не имели дисплеев, вывод информации производился на печатную ленту. Позже, с появлением мониторов, командная строка приобрела свой современный вид, все вводимые команды и их результат стали отражаться на экране компьютера.

При работе с текстовым интерфейсом пользователь использует систему команд, определяемых данной ОС. Команды могут вводиться не только в интерактивном режиме, но и считываться из командного файла, содержащего последовательность команд. Использование текстового интерфейса на первый

взгляд кажется сложным и устаревшим. Тем не менее, есть целый ряд случаев, когда работа с командной строкой является оправданной и даже необходимой:

- 1) работа с программами, которые не имеют графического интерфейса;
- 2) работа с удаленными устройствами при минимальных затратах трафика;
- 3) работа с большими объемами информации, когда отсутствие элементов графического интерфейса позволяет вместить больше данных на странице;
- 4) работа одновременно с несколькими файлами, в которых необходимо с высокой скоростью выполнять различные команды.

1.3.2. Команды в операционных системах.

Команды - основной способ общения пользователя с компьютером в операционной системе MS-DOS и возможный способ общения пользователя с компьютером в других операционных системах. Они вызывают определенное действие в компьютере: организуют передачу информации; вырабатывают необходимый управляющий сигнал; подключают внешнее устройство для организации процесса ввода-вывода информации и т.д.

Команда технически реализована программой в машинных кодах и хранится либо в файле на диске, либо входит в состав командного процессора операционной системы. Как и любая другая программа, команда имеет уникальное имя и всегда типа .COM или .EXE.

Ввод команды осуществляется в командной строке или через файлы в соответствии с определенными правилами, заданными в виде формата.

Классификация команд.

Команды можно классифицировать по двум признакам - по способу реализации и по функциональному назначению. По способу реализации команды разделяются на две группы (рис. 1.):

- 1) резидентные (внутренние, встроенные);
- 2) транзитные (внешние, утилиты, обслуживающие).



Рис. 1. Классификация команд по способу реализации

Резидентные команды входят в состав командного процессора и после окончания загрузки операционной системы обычно располагаются в оперативной памяти. Резидентные команды доступны в любой момент времени. После ввода пользователем резидентной команды операционная система проводит анализ ее структуры и при отсутствии ошибок вызывает ее на выполнение из оперативной памяти. В системе невозможны замена или

добавление резидентных команд, так как они являются частью командного процессора.

Транзитные команды реализуются в виде файлов типа .COM или . После ввода транзитной команды организация работы будет аналогична обычной процедуре обработки программного файла.

Можно неограниченно расширять возможности операционной системы за счет введения новых транзитных команд. По сравнению с резидентными командами транзитные команды характеризуются большим временем выполнения из-за необходимости обращения к диску. Транзитные команды так же, как и резидентные, задаются именем без указания типа.

По функциональному назначению классификация команд может быть сделана весьма условно вследствие разнообразных возможностей многих команд. Основным функциональным признаком может служить объект, с которым производятся различные действия, например каталог, файл, диск. В этом случае выделяются следующие группы команд: для работы с каталогами, для работы с файлами, для работы с дисками, управления памятью и устройствами, конфигурирования системы и др.

Подробные сведения о каждой команде можно получить двумя способами - введя команду HELP, которая подключает справочную систему по всем командам, или по конкретной команде:

HELP <Имя команды>

Для выдачи текста справки не на экран, а в файл, можно воспользоваться перенаправлением вывода. Стандартным устройством ввода является клавиатура, а устройством вывода – дисплей, но эти устройства можно переназначить с использованием символов перенаправления:

< - перенаправление ввода;

>- перенаправление вывода (или >> - перенаправление в существующий файл, когда выводимые данные дописываются в конец файла).

1.3.3. Командные файлы.

Операционная система в своем арсенале инструментальных программных средств содержит специальный программный механизм для автоматизации работы с командами. Он позволяет не только облегчить работу программиста за счет автоматизации часто повторяющейся совокупности команд, но и освободить пользователя от необходимости знания многих тонкостей форматов команд.

Последовательность автоматически выполняемых операций обработки в операционной системе получила название пакетной обработки (batch processing). Инструментальным средством пакетной обработки является командный (пакетный) файл.

Пакетный файл (англ. batch file) — текстовый файл в MS-DOS, OS/2 или Windows, содержащий последовательность команд, предназначенных для исполнения командным интерпретатором. После запуска пакетного файла, программа - интерпретатор (как правило COMMAND.COM или CMD.EXE)

читает его строка за строкой и последовательно исполняет команды. Пакетный файл — аналог shell script в Unix-подобных операционных системах.

Пакетные файлы полезны для автоматического запуска приложений. Основная область применения - автоматизация наиболее рутинных операций, что регулярно приходится совершать пользователю компьютера. Примерами таких операций могут служить - обработка текстовых файлов; копирование, перемещение, переименование, удаление файлов; работа с папками; архивация; создание резервных копий баз данных и т. п.

Пакетные файлы поддерживают операторы `if` и `goto` (а в системах семейства Windows NT и расширенный оператор `for`), что позволяет обрабатывать результаты выполнения предыдущих команд или приложений, и в зависимости от этого выполнять дальше тот или иной блок команд (как правило, в случае удачного завершения приложение возвращает 0 в переменной `errorlevel`; в случае неудачного — 1 или большее значение).

Основные команды командных файлов.

1. `Assoc` - Выводит или изменяет сопоставления (associations) типов файлов;

2. `break` - Задаёт точки останова при отладке;

3. `cd (chdir)` - Показывает имя текущего каталога или выполняет смену текущего каталога;

4. `cls` - Очищает окно командной строки и буфер экрана,

5. `color` - Задаёт цвета текста и фона окна командной оболочки,

6. `copy` - Копирует файлы или выполняет конкатенацию файлов,

7. `date` - Показывает или устанавливает текущую дату,

8. `del (erase)` - Удаляет заданный файл, группу файлов или каталог,

9. `dir` - Показывает список подкаталогов и файлов в текущем или заданном каталоге,

10. `echo` - Выводит текст в окно командной строки или задаёт, надо ли отображать команды на экране (`on|off`),

11. `endlocal` - Отмечает конец локализации (локальной области видимости) переменных,

12. `exit` - Выход из оболочки командной строки,

13. `ftype` - Выводит или изменяет текущие типы файлов в сопоставлениях расширений файлов с программами,

14. `goto` - Указывает, что интерпретатор команд должен перейти на строку с заданной меткой в пакетном сценарии,

15. `if` - Выполняет команды по условию,

16. `md (mkdir)` - Создает подкаталог в текущем или заданном каталоге,

17. `move` - Перемещает файл или группу файлов из текущего или заданного исходного каталога в указанный каталог. Также может переименовывать каталог,

18. `path` - Показывает или задаёт путь к командам, используемый операционной системой при поиске исполняемых файлов и сценариев,

19. pause - Останавливает выполнение пакетного файла и ожидает ввода с клавиатуры,
20. popd - Делает текущим каталог, имя которого было сохранено командой PUSHd,
21. prompt - Указывает, какой текст должен показываться в строке приглашения,
22. pushd - Сохраняет имя текущего каталога и при необходимости делает текущим заданный каталог,
23. rd (rmdir) - Удаляет каталог или каталог вместе с его подкаталогами,
24. rem - Помечает комментарии в пакетном сценарии или Config.nt,
25. ren (rename) - Переименовывает файл или группу файлов,
26. set - Показывает текущие переменные окружения или задает временные переменные для текущей командной оболочки,
27. setlocal - Отмечает начало локализации (локальной области видимости) переменных в пакетных сценариях,
28. shift - Сдвигает позицию замещаемых параметров в пакетных сценариях,
29. start - Запускает заданную программу или команду в отдельном окне,
30. time - Показывает или устанавливает системное время,
31. title - Задает заголовок окна командной оболочки,
32. type - Показывает содержимое текстового файла,
33. verify - Включает режим проверки файлов после записи на диск,
34. vol - Показывает метку и серийный номер дискового тома.

Синтаксис любой внутренней команды (и большинства внешних) можно получить, введя в командной строке имя команды и /?, например: `copy /?`

Пакетные файлы в MS DOS имеют расширение - .bat; для других операционных систем они могут иметь другие расширения — например, .CMD в Windows NT и OS/2 или .BTM в 4DOS, или подобных оболочках.

Пакетные файлы могут содержать как внутренние команды, обрабатываемые непосредственно COMMAND.COM или CMD.EXE, так и обращения к внешним утилитам, существующим в виде отдельных программ (.EXE файлов). Данные программы значительно расширяют возможности пакетных файлов.

Командный файл создается как текстовый файл в любом текстовом редакторе. Запускается командный файл на выполнение так же, как и команды ОС или файлы запуска прикладных программных систем, имеющие тип .COM или .EXE.

Основные свойства командного файла.

1. Командный файл состоит из команд операционной системы, имен файлов запуска, сервисных программных средств и ряда команд, специально созданных для управления работой командного файла.
2. Каждая команда занимает отдельную строку.
3. Имя командного файла - уникальное в пределах того каталога, где он находится.

4. В конструкции команд могут быть как строчные, так и прописные буквы.

Существует возможность передать командному файлу параметры командной строки и использовать их значения в операторах самого командного файла.

<BAT-файл> <параметр1> <параметр2>... <параметрN>

В самом командном файле первый параметр будет доступен как переменная %1, второй - %2 и т.п. Имя самого командного файла доступно как переменная %0. Команда SHIFT позволяет сдвигать начало отсчета параметров на 1.

Вызов командного файла внутри другого командного файла осуществляется с помощью команды: call имя командного файла. При вызове оператора call программа возвращается в точку, где произошёл вызов.

При прямом вызове командного файла внутри другого командного файла программа не возвращается в командный файл.

1.3.4. Командная оболочка.

Программирование пакетных файлов появилось в MS-DOS и Windows с самого зарождения этих операционных систем. Командные интерпретаторы этих систем предлагают два режима работы: интерактивный режим (когда пользователь непосредственно вводит команды в командной строке и немедленно их исполняет) и пакетный режим (когда пользователь запускает предварительно записанную последовательность команд). Концепция обоих режимов была почерпнута из различных Unix—оболочек, равно как и из других текстовых интерфейсов командной строки начала 1980-х годов, таких как CP/M.

Командный интерпретатор в MS-DOS имеет название COMMAND.COM. ОС Windows включают интерпретатор cmd.exe, который частично совместим с COMMAND.COM. Некоторые старые возможности MS-DOS недоступны, однако вместо них появились дополнительные возможности и команды. COMMAND.COM включается в NT-подобные системы для обеспечения лучшей обратной совместимости.

Существуют и другие командные интерпретаторы, разработанные не компанией Microsoft и предоставляющие расширенный синтаксис команд для пакетного программирования. Примером может служить 4DOS.

Также имеются компиляторы пакетных файлов, превращающие пакетные файлы в исполняемые программы.

Командная оболочка в операционных системах Windows вызывается одинаково. Для этого существует несколько способов:

1. Меню **Пуск** → **Выполнить**. Открывается небольшое окно, в котором нужно ввести команду «cmd» и нажать клавишу **Enter**. В Windows 7 команду «cmd» необходимо вводить в строке поиска в меню Пуск. После этого на экране появится окно командной строки.

2. Для вызова командной строки в операционных системах семейства Windows предусмотрен специальный ярлык. Находится он по адресу

Пуск → Программы → Стандартные → Командная строка.

3. Кроме этого, в операционных системах Windows для вызова командной строки существует специальное сочетание горячих клавиш. Нажмите **Win+R** – после чего откроется окно, в котором необходимо набрать «cmd» и нажать «ОК», после чего будет выполнен вход в командную строку.

В системе имеется возможность использовать командную оболочку для создания и редактирования пакетных файлов (также называемых сценариями), что позволит автоматизировать выполнение обычных задач. Например, можно использовать сценарии для автоматизации управления учетными записями пользователей и ежедневной архивацией в нерабочие часы. Выполнение операций с помощью пакетных файлов является более эффективным, чем с помощью интерфейса пользователя. Пакетные файлы принимают все команды, доступные из командной строки.

При работе с командной оболочкой можно настроить окно командной строки для облегчения просмотра и для увеличения контроля за выполнением программ.

Настройка командной строки через изменение свойств ярлыка:

1. Откройте окно **Командная строка**.
2. Щелкните указателем мышки левый верхний угол окна и выберите команду **Свойства**.
3. Выберите вкладку **Общие**.
4. В области **Запоминание команд** выберите или введите значение 999 в поле **Размер буфера**, а затем выберите или введите значение 5 в поле **Количество буферов**.

Увеличение размера буфера экрана до 999 позволяет выполнять прокрутку в окне командной строки. Увеличение количества буферов до пяти увеличивает число строк в окне командной строки до 5000.

5. В области **Правка** установите флажки **Выделение мышью** и **Быстрая вставка**.

Флажок **Выделение мышью** позволяет копировать и вставлять данные в окне командной строки. Чтобы скопировать, выделите левой кнопкой мыши текст в окне командной строки и нажмите правую кнопку мыши. Чтобы выполнить вставку в командную строку или текстовый файл, щелкните правой кнопкой мыши.

6. Также можно изменить расположение и размер окна командной строки.

7. В диалоговом окне **Изменение свойств** выберите вариант **Сохранить свойства для других окон с тем же именем**.

Изменение внешнего вида и некоторых свойств командной строки можно выполнить не только с помощью определенных ключей при запуске cmd.exe, но и изменив параметры реестра, предназначенные для командного процессора в разделах:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Command Processor – для всех пользователей.

HKEY_CURRENT_USER\Software\Microsoft\Command Processor\ - для текущего пользователя.

HKEY_CURRENT_USER\Console\ - дополнительно для текущего пользователя.

Параметры командной строки при запуске cmd.exe имеют наивысший приоритет, а пользовательские настройки (раздел реестра HKEY_CURRENT_USER) имеют приоритет над настройками для компьютера (раздел реестра HKEY_LOCAL_MACHINE). Одной из наиболее полезных настроек является изменение обработки вводимой командной строки таким образом, чтобы можно было выполнять подстановку имен файлов и каталогов при нажатии специальной клавиши, как это реализовано в командной оболочке Unix-подобных операционных систем.

1.3.5. Переменные окружения.

Окружение в MS-DOS является общим полем памяти для всех программ, с использованием которой они могут взаимодействовать как между собой, так и с пользователем.

Окружение хранит переменные, называемые глобальными, и каждая программа имеет возможность получить их значение, изменить значение, добавить или удалить их из окружения. Возможность выполняемых программ взаимодействовать через окружение с пользователем основано на том, что все права доступа к окружению имеют не только программы, но и пользователь с помощью команд.

Окружение создаётся интерпретатором команд и передаётся каждой запускаемой программе в виде своей копии, которая затем может обновляться. В общем случае в памяти храниться иерархия окружений. Собственником корневого окружения (самого главного окружения-оригинала) является интерпретатор команд. Остальные окружения принадлежат исполняемым программам. Каждая программа имеет доступ только к своему экземпляру окружения. При завершении программы её окружение уничтожается.

При такой организации работы с окружением никакая программа не способна изменить окружение другой программы, в частности, корневого окружения. Такие права предоставлены только пользователю, если он непосредственно взаимодействует с первичной копией интерпретатора команд.

Стандартными глобальными переменными считаются переменные, которые используются для настройки самой DOS.

Они определяют приглашение операционной системы, пути, по которым ищутся вспомогательные файлы и программы, настройки аппаратных средств и опции вывода оглавления каталога.

Переменные окружения используются в любой ОС, где используются командные файлы.

В командных файлах удобно использовать переменные окружения – переменные, значения которых характеризуют среду, в которой выполняется команда или пакетный файл. Значения переменных окружения формируются при загрузке ОС и регистрации пользователя в системе, а также могут быть заданы с помощью команды SET , формат которой:

SET [переменная=[строка]], где

Переменная - Имя переменной среды,

Строка - Строка символов, присваиваемая указанной переменной.

Можно получить значение переменной в программах и командных файлах, используя ее имя, заключенное в знаки процента (%).

Выполнение команды SET без параметров вызывают выдачу текущих значений переменных на экран.

Значение некоторых переменных по команде SET не выдаются. Это переменные, значения которых динамически изменяются :

%CD% - Принимает значение строки текущей директории.

%DATE% - Принимает значение текущей даты.

%TIME% - Принимает значение текущего времени.

%RANDOM% - Принимает значение случайного десятичного числа в диапазоне 1 -32767.

%ERRORLEVEL% - Принимает текущее значение кода завершения задачи ERRORLEVEL

%CMDEXTVERSION% - Принимает значение версии командного процессора CMD.EXE для расширенной обработки команд.

%CMDCMDLINE% - Принимает значение строки, которая вызвала командный процессор.

Для просмотра значения переменной можно использовать команду:

ECHO % переменная %.

Нельзя использовать в команде set никаких лишних пробелов. Указание пробелов после знака "=" при определении переменной в любой операционной системе всегда приводит к очистке этой переменной.

Задание и сброс стандартных глобальных переменных окружения возможно только в файле AUTOEXEC.BAT и в "чистой" командной строке MS-DOS. Задание же переменных окружения в файловых оболочках не приведет к изменению значений переменных окружения. В операционных оболочках (и в графических оболочках MS WINDOWS 95/98/NT) изменение переменных окружения произойдет только на текущий сеанс эмуляции MS-DOS).

1.3.6. Использование переменных среды в Cmd.exe для ОС Windows.

Среда командной оболочки Cmd.exe определяется переменными, задающими поведение командной оболочки и операционной системы. Поведение среды командной оболочки или среды всей операционной системы можно определить с помощью двух типов переменных среды: системных и локальных. Системные переменные среды определяют поведение глобальной среды операционной системы. Локальные переменные среды определяют поведение среды в данном экземпляре Cmd.exe.

Системные переменные среды заданы заранее в операционной системе и доступны для всех процессов Windows. Только пользователи с привилегиями администратора могут изменять эти переменные. Эти переменные наиболее часто используются в сценариях входа в систему.

Локальные переменные среды доступны, только когда пользователь, для которого они были созданы, вошел в систему. Локальные переменные из куста HKEY_CURRENT_USER подходят только для текущего пользователя, но определяют поведение глобальной среды операционной системы.

В следующем списке представлены различные типы переменных в порядке убывания приоритета.

1. Встроенные системные переменные.
 2. Системные переменные куста HKEY_LOCAL_MACHINE
 3. Локальные переменные куста HKEY_CURRENT_USER
 4. Все переменные среды и пути указаны в файле Autoexec.bat.
 5. Все переменные среды и пути указаны в сценарии входа в систему (если он имеется).
 6. Переменные, используемые интерактивно в сценарии или пакетном файле.
- В командной оболочке каждый экземпляр Cmd.exe наследует среду своего родительского приложения. Поэтому можно изменять переменные в новой среде Cmd.exe, что не повлияет на среду родительского приложения.

В табл. 1 приведен список системных и локальных переменных среды для Windows XP.

Таблица 1

Системные и локальные переменные среды для Windows XP

| Переменная | Тип | Описание |
|-------------------|-----------|---|
| %ALLUSERSPROFILE% | Локальная | Возвращает размещение профиля «All Users». |
| %APPDATA% | Локальная | Возвращает используемое по умолчанию размещение данных приложений. |
| %CD% | Локальная | Возвращает путь к текущей папке. |
| %CMDCMDLINE% | Локальная | Возвращает строку команд, с помощью которой был запущен данный экземпляр Cmd.exe. |
| %CMDEXTVERSION% | Системная | Возвращает номер версии текущих расширений обработчика команд. |
| %COMPUTERNAME% | Системная | Возвращает имя компьютера. |
| %COMSPEC% | Системная | Возвращает путь к исполняемой командной оболочке. |
| %DATE% | Системная | Возвращает текущие данные. Использует тот же формат, что и команда date /t. Создается командой Cmd.exe. |
| %ERRORLEVEL% | Системная | Возвращает код ошибки последней использовавшейся команды. Значение, не равное нулю, обычно указывает на наличие ошибки. |

| | | |
|--------------------------|-----------|--|
| %HOMEDRIVE% | Системная | Возвращает имя диска локальной рабочей станции, связанного с основным каталогом пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы». |
| %HOMEPATH% | Системная | Возвращает полный путь к основному каталогу пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы». |
| %HOMESHARE% | Системная | Возвращает сетевой путь к общему основному каталогу пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы». |
| %LOGONSERVER% | Локальная | Возвращает имя контроллера домена, который проверял подлинность текущей сессии. |
| %NUMBER_OF_PROCESSORS% | Системная | Задаёт количество процессоров, установленных на компьютере. |
| %OS% | Системная | Возвращает имя операционной системы. При использовании Windows 2000 имя операционной системы отображается как Windows_NT. |
| %PATH% | Системная | Указывает путь поиска для исполняемых файлов. |
| %PATHEXT% | Системная | Возвращает список расширений файлов, которые рассматриваются операционной системой как исполняемые. |
| %PROCESSOR_ARCHITECTURE% | Системная | Возвращает архитектуру процессора. Значения: x86, IA64. |
| %PROCESSOR_IDENTIFIER% | Системная | Возвращает описание процессора. |
| %PROCESSOR_LEVEL% | Системная | Возвращает номер модели процессора, установленного на компьютере. |

| | | |
|----------------------|------------------------------|---|
| %PROCESSOR_REVISION% | Системная | Возвращает номер модификации процессора. |
| %PROMPT% | Локальная | Возвращает параметры командной строки для текущего интерпретатора. Создается командой Cmd.exe. |
| %RANDOM% | Системная | Возвращает произвольное десятичное число от 0 до 32767. Создается командой Cmd.exe. |
| %SYSTEMDRIVE% | Системная | Возвращает имя диска, содержащего корневой каталог WindowsXP (т. е. системный каталог). |
| %SYSTEMROOT% | Системная | Возвращает размещение системного каталога WindowsXP. |
| %TEMP% и %TMP% | Системная и пользовательская | Возвращает временные папки по умолчанию используемые приложениями, которые доступны пользователям, выполнившим вход в систему. Некоторые приложения требуют переменную TEMP, другие — переменную TMP. |
| %TIME% | Системная | Возвращает текущее время. Использует тот же формат, что и команда time /t. Создается командой Cmd.exe. |
| %USERDOMAIN% | Локальная | Возвращает имя домена, содержащего список учетных записей пользователей. |
| %USERNAME% | Локальная | Возвращает имя пользователя, выполнившего вход в систему. |
| %USERPROFILE% | Локальная | Возвращает размещение профиля для текущего пользователя. |
| %WINDIR% | Системная | Возвращает размещение каталога операционной системы. |

Командная строка позволяет просматривать системную информацию с помощью команды SystemInfo. После ее выполнения отображается полный перечень информации об операционной системе и компонентах компьютера, например, такие детали, как версия операционной системы, состояние памяти и тип процессора. Кроме того, некоторая информация о сети, как IP или MAC-адреса.

Чтобы просмотреть список запущенных процессов, можно использовать команду tasklist – список задач. В результате представляются данные, такие как имя процесса, PID (идентификатор процесса) или памяти.

1.4. Задание на выполнение

1. Просмотреть, какие переменные среды заданы в системе. Прodelать это двумя способами: через окно свойств системы и используя команду SET из окна командной строки.

Создать свою переменную, содержащую, например, путь к некоторой папке. Проверить ее действительность.

2. Продемонстрировать работоспособность всех основных команд встроенного интерпретатора команд системы. В отчете привести примеры набранных команд.

3. Получить информацию об операционной системе и памяти работающих процессов.

4. Разработать схему алгоритма и командный файл для удаления резервных копий из заданного каталога, если он указан, или из рабочего каталога, если аргумент не специфицирован. Осуществить автозапуск командного файла.

5. Разработать схему алгоритма и командный файл, который сообщает день недели для любой приемлемой для MS DOS даты.

1.5. Порядок выполнения работы

1. Создайте рабочий каталог.
2. Создайте командные файлы согласно заданию.
3. Запустите их на исполнение.
4. Распечатайте файлы конфигурации и командные файлы.

1.6. Контрольные вопросы

1. Типы пользовательских интерфейсов различных ОС.
2. Вызов и настройка командной оболочки.
3. Классификация команд.
4. Понятие окружения.
5. Глобальные и локальные переменные системы.
6. Стандартные глобальные переменные.
7. Структура и состав командных файлов.
8. Отличия командных файлов и макросов.

Литература

1. Попов А.В. Командные файлы и сценарии Windows Script Host. - СПб.: БХВ-Петербург, 2002.
2. Таненбаум Э. Современные операционные системы. - СПб.: Питер, 2006.
3. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб.: Питер, 2003.
4. www.microsoft.ru