

Операционные системы

Лабораторная работа № 1

Основные консольные команды

Приведем основные команды, необходимые для работы. В квадратных скобках [...] указываются не обязательные параметры команды, в треугольных <...> – обязательные.

`cd [диск:] <путь>` - изменение текущего каталога.

`dir [диск:] [путь\] [имя-файла] [параметры]` - просмотр каталога.

`md [диск:] <путь>` - создание каталога.

`rd [диск:] <путь>` - уничтожение каталога.

`copy con <имя файла>` - создание текстовых файлов. После ввода этой команды нужно будет поочередно вводить строки файла. В конце каждой строки нужно нажимать клавишу Enter, а после ввода последней – нажать клавишу Ctrl+Z или F6 и затем Enter.

`del <имя файла>` - удаление файлов.

`ren <имя файла1> <имя файла2>` - переименование файлов.

`copy <имя файла1> <имя файла2>` или

`copy <имя-файла1> [имя-каталога]` - копирование файлов.

`copy <имя файла1> [+ имя-файла2] <имя файла3>` - соединение (конкатенация) файлов.

`type <имя файла>` - вывод файла на экран.

`cls` - очистка экрана монитора

`vol` - вывод метки диска

`tree [диск:] [путь] [/F]` – выводит дерево каталогов. Параметр [путь] задает каталог, для которого вы хотите вывести структуру. Параметр /F выводит имена файлов в каждом каталоге.

`color [цвета]` – изменение цвета экрана и текста.

`edit` – вызов текстового редактора.

`ключ /?` – выводит справку о команде.

`Команда /? help` – вызов справки.

Перенаправление вывода `имя_команды > имя_файла.txt` – передает результат выполнения команды `имя_команды` в файл `имя_файла.txt` (перенаправляет вывод из консоли в файл). Например, команда `dir>dir.txt` позволяет сохранить оглавление текущего каталога в виде текстового файла `dir.txt`.

Пакетные файлы

Пакетные (командные) файлы - средство, позволяющее автоматизировать часто выполняемые действия пользователя. Пакетные файлы могут выполнять довольно сложную последовательность действий. Пакетный файл представляет собой обычный текстовый файл, расширение которого меняется на `.BAT` (batch file – пакетный файл). Для запуска командного файла достаточно набрать в командной строке его имя, расширение можно не указывать.

Командный файл состоит из последовательности строк, в каждой из которых, может находиться либо вызов программы, либо вспомогательные команды. Большинство строк командного файла обрабатываются так же, как если бы они вводились пользователем в командной строке. Вспомогательные команды служат для управления ходом работы командного файла. Наиболее распространенные вспомогательные команды:

ECHO

При формате вызова `ECHO OFF` отменяет вывод строк командного файла на экран при выполнении. Команда `ECHO ON` возобновляет их вывод на экран. Большинство командных файлов начинаются со строки `@ECHO OFF`. Символ "@" служит для отмены вывода на экран строки, следующей непосредственно за ним. В остальных случаях команда `ECHO` выводит на экран текст, следующий за ней в строке.

GOTO

Применяется для перехода к определенной строке командного файла. При выполнении команды `GOTO LABEL` происходит переход к строке, начинающейся с текста `:LABEL`.

IF

Служит для проверки условия во время выполнения командного файла. Формат команды:

`IF УСЛОВИЕ КОМАНДА.`

Команда будет выполнена в том случае, если условие будет истинно. Однако допустим формат `IF NOT УСЛОВИЕ КОМАНДА`. При этом команда выполнится, если условие ложно. В качестве команды часто используется `GOTO`.

PAUSE

Приостанавливает выполнение командного файла до нажатия какой-либо клавиши. Если нажать клавиши `Ctrl+Break`, выполнение командного файла будет прервано.

REM

Строки, начинающиеся с REM, при выполнении командного файла игнорируются. В них можно записывать комментарии.

Пример. Пакетный файл создает папку и копирует туда содержимое текущей папки.

```
echo off

cls

echo start backup
mkdir backup
copy *.* backup
echo end backup
```

Переменные

Переменные в пакетных файлах задаются при помощи команды SET. При использовании значения переменной, ее имя необходимо заключать в значки «%» с обеих сторон.

Пример. Хранение параметра в переменной

```
set file=dostoansi

C:\MASM32\BIN\Ml.exe /c /coff %file%.asm

C:\MASM32\BIN\Link.exe /SUBSYSTEM:WINDOWS %file%.obj /RELEASE
```

В переменных можно хранить не только строковые, но и числовые значения. Вычисление осуществляется командой SET, если она выполняется с ключом /A. При этом поддерживается практически полный набор операторов языка C, включая шестнадцатеричный модификатор 0x.

Пример. Сложение двух чисел

```
@echo off

set a=2

set b=2

set /a c=%a%+%b%

echo %c%

pause
```

Чтобы организовать ввод значения переменной с клавиатуры, можно воспользоваться командой SET с параметром /P. Модифицируем предыдущий пример так, чтобы значения переменных a и b можно было вводить с клавиатуры.

Пример. Ввод значений переменных с клавиатуры

```
@echo off

set /p a="Input a "

set /p b="Input b "
```

```
set /a c=%a%+%b%  
echo %c%  
pause
```

Параметры командной строки

Иногда удобней вводить значения переменных не во время исполнения программы, а сразу передавать их на вход программы как параметр. Так например, при выполнении команды `DIR /P`, ключ `/P` является параметром и указывает что, необходимо выводить содержимое каталога постранично. В командном файле параметры указываются с помощью выражений `%x`, где `x` - цифра от 1 до 9. `%0` означает имя самого командного файла. Разберем применение параметров командной строки на примере сложения двух чисел.

Пример. Параметры командной строки

```
@echo off  
set /a c=%1+%2  
echo %c%  
Pause
```

Ветвления в пакетных файлах

Ветвления в пакетных файлах организуются при помощи команды `IF`. Оператор `IF` проверяет истинность какого-либо `<условия>` и в зависимости от этого выполняет или не выполняет команду `<следствие>`.

```
IF EXIST TEST.BAT GOTO lab1
```

если существует файл TEST.BAT, то перейти на метку lab1.

```
IF NOT EXIST TEST.BAT GOTO exit
```

если файл TEST.BAT не существует, то следует перейти на метку exit.

```
IF %1==ASDF GOTO asdflabel
```

если переменная %1 равна ASDF, то осуществляется переход на метку asdflabel

```
IF ERRORLEVEL 1 GOTO err1
```

если ошибка (код возврата) выполнения предыдущей команды равно 1, то перейти на метку err1.

Если после проверки условия необходимо выполнить несколько команд, то их необходимо заключить в круглые скобки

```
if %1 == 0 (  
set RESULT=1  
exit /b
```

```
)
```

Также можно организовать ветвления с двумя и более направлениями

```
if condition (
```

```
ветвь 1
```

```
) else (
```

```
ветвь 2
```

```
)
```

Пример. Изменение цвета интерфейса в зависимости от параметра.

```
@echo off
```

```
if %1==r (
```

```
color 04
```

```
goto ext
```

```
) else (
```

```
color 40
```

```
goto ext
```

```
)
```

```
:ext
```

```
Pause
```

Циклы в пакетных файлах

Команда FOR позволяет организовать выполнение повторяющихся однотипных действий. Можно использовать ее для того, чтобы вывести на экран числа от одного до десяти.

```
for /l %%i in (1,1,10) do echo %%i
```

Переменная *i* называется счетчиком цикла. В силу своеобразия синтаксиса команды FOR, имя счетчика цикла должно состоять из одной буквы. Причем, если мы пишем командный файл, то перед именем счетчика цикла надо поставить удвоенный знак процента, если же мы просто набираем команду в командной строке, то одиночный. Логика работы этой команды такова. После слова *in* указан диапазон изменения счетчика цикла. В данном варианте команды это тройка чисел: начальное значение счетчика, шаг счета, предельное значение счетчика. При выполнении команды командный процессор сначала присвоит переменной *i* значение 1, а потом на каждом шаге цикла будет увеличивать его на 1, пока оно не превысит 10. Очевидно, таких шагов получится десять. Если бы в качестве шага счета мы указали число 2, то цикл выполнялся бы пять раз. На каждом шаге цикла выполняется тело цикла, написанное после слова *do*. В приведенном примере это команда *echo*, которая выводит на экран текущее значение счетчика цикла.

Обычно команда FOR используется для перебора и обработки файлов. Надо сказать, что в достаточно простых случаях массовая обработка файлов выполняется с помощью подстановочных символов. Если, мы хотим всем файлам в текущем каталоге заменить расширение .htm на .html, мы вводим команду `ren *.htm *.html`. Но если то же самое надо сделать не в одном каталоге, а в дереве каталогов, то без команды FOR не обойтись.

Приведенный ниже командный файл выполняет эту операцию для всех htm-файлов в подкаталоге website текущего каталога. Точнее, во всем дереве каталогов, которое находится внутри website.

```
for /r website %%i in (*.htm) do ren %%i %%~ni.html
```

Ключ `/r` указывает на необходимость обхода каталога website и всех его внутренностей. Если его не указать (но тогда и каталог указывать не разрешается), то обработаны будут только файлы в текущем каталоге. Диапазоном значений счетчика цикла в данном варианте команды является множество всех файлов с расширением .htm, находящихся внутри каталога (точнее, дерева) website. Запись `~ni` означает, что из значения переменной `i` требуется выделить только имя файла. В языке команд предусмотрено несколько таких модификаторов, например, запись `~xi` обозначает расширение файла. Все модификаторы описаны в справке по команде FOR.

Тело цикла может состоять из нескольких команд, заключенных в скобки.

```
@echo off
for /r website %%i in (*.htm) do (
rem Выводим имя файла
echo %%i
rem Переименовываем файл
ren %%i %%~ni.html
)
```

Задания к лабораторной работе

Задание 1.

1. Вызовите командный процессор cmd.exe (Пуск-Выполнить-cmd или Пуск-Все программы-Командная строка).
2. Создайте на диске D: каталог своей группы
3. В своем каталоге создать каталоги First и Second.
4. Работа в каталоге First:
 - Создать текстовый файл – содержащий описание способов создания файлов.
 - Создать текстовый файл содержащий справочную информацию о текстовом редакторе.
5. В каталоге Second:
 - Скопировать все файлы из каталога First
 - Скопировать все файлы начинающиеся на A (латинскую) из каталога C:\WINDOWS.
 - Объединить файлы, скопированные из каталога First в один файл.
 - Создать текстовый файл, отображающий содержимое каталога.
 - Удалить все файлы, начинающиеся на A.
6. Очистить экран.
7. Выведите на экран дерево каталогов своей папки.

Задание 2.

1. Создайте командный файл COMM.BAT, выполняющий действия:
 - Отключение режима отображения на экране выполняемой команды.
 - Вывод на экран: "Копирование и удаление файла".
 - Создание на диске каталога DIR1, и в нем создание каталога DIR2.
 - Копирование файла с именем TEXT1.TXT, предварительно созданного в папке группы, в файл с именем TEXTNEW.TXT в каталоге \DIR1\DIR2.
 - Удаление исходного файла.
 - Вывод на экран: "Файл скопирован и удален".
 - Пауза до нажатия клавиши.
2. Создайте командный файл с именем SUMMA.BAT, выполняющий действия:
 - Вывод на экран «Объединение и переименование файлов».
 - Объединение содержимого файлов A.TXT и B.TXT, предварительно созданных в папке своей группы, под именем C.TXT.
 - Вывод содержимого объединенного файла на экран.

- Ожидание нажатия клавиши.

- Переименование файлов А.TXT и В.TXT в FINA.TXT и FINB.TXT соответственно.

- Вывод на экран: "Задание выполнено".

3. Предъявите работу преподавателю, после чего удалите все созданные вами файлы и каталоги.

Задание 3. Работа с переменными

Напишите пакетный файл, сравнивающий два числа. Числа вводятся пользователем с клавиатуры. В результате сравнения выводится одно из следующих сообщений: «Первое число больше», «Второе число больше», «Числа равны».

Задание 4. Параметры командной строки

Напишите пакетный файл для, определяющий наличие файла в директории с вводимым пользователем именем. Имя искомого файла передается как параметр командной строки. В случае если файл присутствует в директории, то ему устанавливается атрибут «системный», иначе выводится сообщение «файл не найден». Для установления атрибутов файла воспользуйтесь командой ATTRIB, синтаксис команды разберите самостоятельно.

Задание 5. Организация цикла

Создайте пакетный файл, выводящий имя и содержимое всех файлов с расширением .txt в задаваемой пользователем директории. Каталог задается из командной строки. Перед выводом содержимого каждого файла выводится полное имя файла (включая каталог).

Задание 6. Организация меню

Создайте пакетный файл, выполняющий резервное копирование каталога. Имя резервного каталога задается пользователем, если пользователь не задает имя каталога, то копия помещается в том же разделе, где исходный каталог. Пакетный файл должен предоставлять пользователю меню, позволяющее выбрать один из следующих режимов работы:

- 1) замещение файлов (файлы из исходного каталога замещают файлы резервного);
- 2) добавление файлов (если в резервном каталоге файла не существует, то он добавляется из исходного, прочие файлы не заменяются);
- 3) удаление файлов (если в резервном каталоге существует файл, которого нет в исходном, то он удаляется).

Предусмотрите возможность вызова справки по работе пакетного файла.