

ПРИЛОЖЕНИЯ UNIX СИСТЕМ

Лекция №2

Ст. преподаватель
кафедры ПДСиМ
Квиткова Ирина Геннадьевна

Подстановка команд

Подстановка команд заключается в замене имени команды на результат ее выполнения, т.е. можно результат выполнения одной команды подставить в параметр другой команды.

Формы подстановки: **`команда`** или **\$(команда)**
cat `find . -size +4k -type f` - просмотр содержимого файлов текущего каталога размером > 4 кбайт.

cp \$(grep 3 * 2> error| cut -f1 -d:) DIR 2>error
– копирование файлов, в тексте которых есть 3, в папку DIR с исключением вывода потока ошибок.

MYFILES=`ls /etc | grep ^pa`; echo \$MYFILES – запись в переменную списка имён файлов каталога ***/etc***, имена которых начинаются на ***pa***; вывод значения переменной на экран.

Средства группировки команд:

; - последовательное выполнение команд;

& - фоновое выполнение предыдущей команды;

&& - выполнение последующей команды при условии нормального завершения предыдущей;

|| - выполнение последующей команды при ненормальном завершении предыдущей.

Для группировки команд могут использоваться фигурные "{" и круглые "(" скобки.

k1 && k2; k3 - команда "k2" будет выполнена только при успешном завершении "k1"; "k3" - в любом случае.

k1 && (k2; k3) - "k2" и "k3" будут выполнены только при успешном завершении "k1".

{k1; k2} & - в фоновом режиме будет выполняться последовательность команд "k1" и "k2".

Создание сценариев

Командный интерпретатор SHELL

Shell (оболочка) в Unix - это программа, воспринимающая ввод команд от пользователя и выполняющая их.

Скрипт (сценарий) - программа на языке shell (исполняемый файл, содержащий команды).

Создание скрипта (как обычный файл):

1) **cat > имя_скрипта**

2) **vi имя_скрипта** (или **nano**)

✓ Созданный скрипт должен иметь право на исполнение:

chmod a+x имя_скрипта

Запуск скрипта:

1) **sh имя_скрипта**

2) **./имя_скрипта**

Shell-переменная

Имя shell-переменной - это начинающаяся с буквы последовательность букв, цифр и подчеркиваний.

Значение shell-переменной - строка символов.

Для присваивания значений переменным используется оператор **"="**.

При обращении к значению shell-переменной перед именем ставится символ **"\$"**.

Например: **var2=13** (без пробелов)

echo \$var2 - выдаст на экран **13**,

echo var2 - выдаст на экран **var2**.

✓ Имя переменной не должно совпадать с именами команд.

VAR=\$(pwd); echo \$VAR - ?

USER=`who| wc -l`; echo \$USERS - ?

Позиционные параметры и специальные переменные

В скрипт могут быть переданы параметры. Для этого используются **позиционные параметры** (существенна очередность их следования).

Интерпретатор shell автоматически присваивает значения следующим переменным (параметрам):

\$0 – имя скрипта или команды, запустившей сценарий;

\$1..\$9 – позиционные параметры;

\$? – результат предыдущей команды
(**0-успешно, 1-нет**);

\$\$ – номер процесса (запущенного сценария);

\$! – номер фонового процесса;

\$# – число позиционных параметров в скрипте;

\$* – перечень параметров в строке через пробел;

@ – перечень параметров в виде списка.

Пример:

```
vi script_1
```

```
ls -l $1 | wc -l; echo $2 | wc -w
```

```
chmod a+x script_1
```

```
sh script_1 D1 a1
```

Скрипт подсчитывает число файлов в каталоге, указанном в параметре **\$1** и число слов в файле, указанном в параметре **\$2**. Значение параметров **\$1** и **\$2** задаются при запуске скрипта через пробел.

Запрос информации у пользователя

Командный интерпретатор позволяет, при необходимости, запрашивать у пользователя информацию, которая помещается в указанную переменную.

Для этого используется команда **read**.

Пример:

echo "задай значение" #запрос на ввод

read x #ввести с клавиатуры значение
переменной x

Hello, world! #вводим значение переменной x

echo \$x # вывести значение переменной на экран

Hello, world!

- ✓ На практике имеет смысл перед запросом выдать приглашение с помощью команды **echo**.