

## **Две разновидности shell:**

- Bourne shell
- C shell

## **Почему администратор должен знать shell script?**

## **Дополнительные возможности, встроенные в shell script:**

- переменные
- массивы
- условные операторы
- циклы

**При запуске программы проверяются условия:**

- право на исполнение
- является ли файл бинарным исполняемым файлом
- если это текстовый файл, есть ли в первой строке путь к интерпретатору

**При использовании `bash` первая строка `shell script` должна быть такой:**

```
#!/bin/bash
```

## **Переменные в shell script:**

- не типизированы
- область видимости переменных - весь код программы
- при обращении к неопределенной переменной не выдаются ошибки

## **Пример использования переменной:**

```
PERM=value
```

```
echo $PERM
```

## **Три способа определения массивов:**

MASS[o]=value

MASS=([o]=value1 [5]=value2)

MASS=(value1 value2)

## **Получение значения элемента массива:**

$\${\text{MASS[o]}}$

## **Получение значений всех элементов массива:**

$\${\text{MASS[*]}}$  или  $\${\text{MASS[@]}}$

## **Экспортирование переменных.**

PERM=value

export PERM

## **Область видимости переменных окружения.**

## **Просмотр переменных окружения.**

set, env и export

## **Удаление переменных.**

## **Встроенные переменные shell.**

## **Оператор read.**

read [переменная ...]

**В подстановке используется то, что программа будет выводить на стандартный вывод.**

**Два варианта записи подстановки:**

``program``

`$(program)`

**В shell script используется ограниченный набор арифметических операций:**

**+ - \* / и круглые скобки.**

**Для подстановки значения арифметических выражений используется:**

**`$(( выражение ))`**

**Если значение переменной, используемой в арифметическом выражении, не является целым числом, то её значение считается равным 0.**



**if** условие

**then**

    список операторов

**[else**

    список операторов ]

**[elif** условие

    список операторов ]

**fi**

**test** [опции] условия ...

**Программа test предназначена для проверки следующих типов условий:**

- сравнение различных значений,
- проверка типов и наличия файлов,
- проверка логических условий.

**&&** – логическое И.

**||** – логическое ИЛИ.

**case** строка **in**

шаблон)

список операторов

;;

[ шаблон)

список операторов

[;;] ]

**esac**

**for** переменная [ **in** список ]

**do**

список операторов

**done**

```
for I in `cat ./sample13-data`
```

**while** условие

**do**

набор операторов

**done**

**select** переменная **in** список  
**do**  
набор операторов  
**done**



**Оператор точка позволяет включать код, находящийся в другом файле.**

```
function sample
```

```
{
```

```
    Тело функции.
```

```
}
```

```
sample()
```

```
{
```

```
    Тело функции.
```

```
}
```

**\$0 \$1 ... \$9** — позиционные переменные.

**\$\*** и **\$@** — все параметры командной строки.

**\$#** — количество параметров командной строки.

**\$?** — код возврата программы.

**\$\_** — PID программы, запущенной в background режиме.

**\$\$** — PID процесса shell.

**getopts** параметры переменная

**trap** команда список сигналов