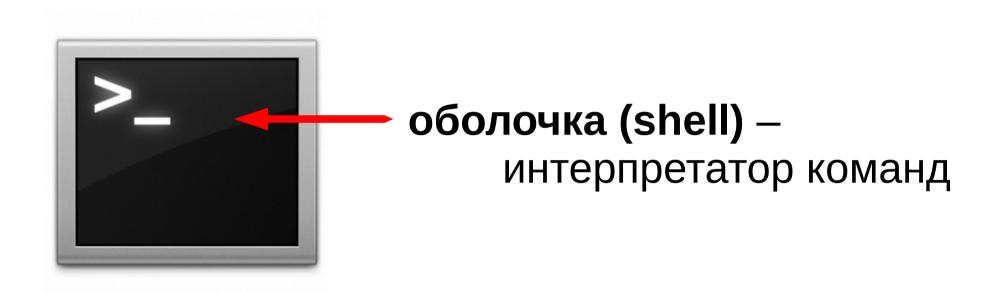
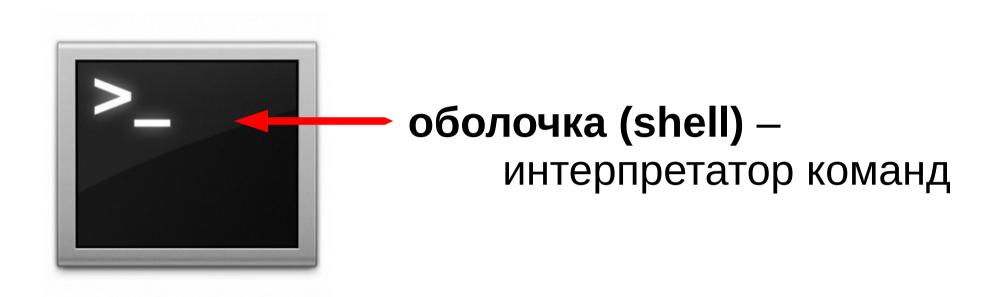


sh (*Bourne shell*) – стандарт и доступна почти в любом дистрибутиве (1977)



sh (*Bourne shell*) – стандарт и доступна почти в любом дистрибутиве (1977)

bash (*Bourne again shell*) – усовершенствованная и модернизированная вариация sh (1989)



sh (*Bourne shell*) – стандарт и доступна почти в любом дистрибутиве (1977)

bash (*Bourne again shell*) – усовершенствованная и модернизированная вариация sh (1989)

csh, tcsh, zsh, ...

Создаем программу:

vim my_first_script.sh

Создаем программу:

vim my_first_script.sh

```
#!/bin/bash
echo "Hello, world!"
```

Запускаем программу:

chmod +x my_first_script.sh
./my_first_script.sh

Переменная (variable)

именованная область памяти

Переменная (variable)

именованная область памяти

имя (пате):

буквы, цифры, _;

не может начинаться с цифры.

значение (value):

числа;

строки (если есть пробелы, то в кавычках); отдельные символы.

Запись (перезапись) значения:

<umя>=<3начение>

path=~/Docs

Запись (перезапись) значения:

```
<uмя>=<3начение>
```

path=~/Docs

Чтение:

\$<uмя> или **\$**{<uмя>}

```
path2=$path/file1.txt
echo "Path is $path2"
echo "Path is ${path}2"
```

```
# path2=~/Docs/file1.txt
# Path is ~/Docs/file1.txt
# Path is ~/Docs2
```

Передача аргументов скрипту:

./script.sh arg1 arg2 arg3 ...

Передача аргументов скрипту:

./script.sh arg1 arg2 arg3 ...

Обработка внутри скрипта:

\$1 первый аргумент

\$2 второй аргумент

• • • •

\$0 имя скрипта

\$# количество аргументов

Передача аргументов скрипту:

./script.sh arg1 arg2 arg3 ...

Обработка внутри скрипта:

```
$1 первый аргумент
```

\$2 второй аргумент

• • • •

\$0 имя скрипта

\$# количество аргументов

var="Первый аргумент равен \$1" echo "Всего было передано \$# аргументов"