



Введение в команды UNIX



UNIX Shell

UNIX Shell - командный интерпретатор, используемый в операционных системах семейства UNIX, посредник между пользователем и операционной системой.

Пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд.



POSIX

POSIX (ISO/IEC 9945) – (Portable Operating System Interface for Unix - Переносимый интерфейс операционных систем Unix) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой.

Название «POSIX» было предложено Ричардом Столлманом.



POSIX

Том 3. Оболочка и утилиты

Определён язык оболочки, включающий конструкции последовательного (перевод строки, точка с запятой), условного (if, case, ||, &&) и циклического (for, for in, while, until) исполнения команд, а также оператор присваивания.

Стандартом также определён режим редактирования вводимых команд, являющийся подмножеством команд стандартного текстового редактора (vi).



UNIX Shells

Bourne shell compatible

- sh, ash, bash, dash, ksh, zsh...

C shell compatible

- csh, tcsh

Exotic

- es (functional), psh (Perl), pysh (Python)



Bourne Shell

Разработана Стивеном Борном из Bell Labs и выпущенная в составе 7-го издания операционной системы UNIX (UNIX Version 7, 1977).

Основана на языке ALGOL 68

"Nobody really knows what the Bourne shell's grammar is. Even examination of the source code is little help."

Tom Duff



C Shell

Разработан Биллом Джоем (университет Беркли) в конце 1970х

Основан на языке C.

Имеет множество улучшений по сравнению с sh



Korn Shell

Разработан Дэвидом Корном в начале 1980х. Имеет полную обратную совместимость с Bourne shell и включает в себя возможности C shell.

«The POSIX standard specifies its standard shell as a strict subset of the Korn shell»



Bourne Again Shell (bash)

Разработан для проекта GNU в замен sh
Брайаном Фоксом и представленный в
1989 году. Собрал в себе множество
идей из csh и ksh.

Основной shell для Linux и MacOS



Bash features

- Редактирование командной строки
- История команд
- Стек каталогов
- Переменные \$RANDOM и \$PPID
- Автозавершение
- ...



Внутренние команды

- `echo` - выводит выражение или содержимое переменной (на `stdout`)
- `printf` - команда форматированного вывода, расширенный вариант команды `echo`
- `read` - «читает» значение переменной со стандартного ввода (`stdin`), в интерактивном режиме это клавиатура



Hello World!

```
echo "Hello World!"
```

```
printf "%s\n" "Hello World"
```



Ключи

Любая команда может быть запущена с ключами:

```
$ echo -e "a\n"
```

a

-h, --help – помощь

-? – помощь (работает не у всех команд)

Группировка ключей:

```
ls -ltra
```



Помощь

- `help COMMAND` – выводит краткую справку по использованию внутренней команды *COMMAND*
- `man COMMAND` – выводит подробную справку по использованию *COMMAND*
- `info COMMAND` – дополнительная информация о команде



Работа с файлами

- `cd` – перемещение по директориям
- `pwd` – текущий каталог
- `mkdir` – создание каталогов
- `rm` – удаление файлов и каталогов
- `cp` – копирование файлов
- `touch` – создать файл
- `ls` – вывод содержимого каталога



Полезные команды

- more, less
- head, tail
- wc
- sort
- tee
- find
- locate
- grep



Автозавершение

Напечатать первые буква команды

Нажать Tab

Если один вариант – автоматическая печать команды

Если нет – нажав еще раз Tab увидим полный список вариантов



ПОТОКИ

Поток номер 0 (stdin) – входные данные.

Можно изменять цель с помощью символа >

```
ls > a.txt
```

Поток номер 1 (stdout) - выходные данные

Можно перенаправить символом >

```
grep 'apt' < a
```

```
grep 'apt' < a > b
```

Поток номер 2 (stderr) - ошибки

Перенаправлять символом 2>

```
ping -c 5 google.com > result.txt 2> error.txt
```



Потоки

- > - перезапись
- >> - добавление в конец



Конвейер (pipe)

Конвейер передает вывод предыдущей команды на ввод следующей или на вход командного интерпретатора. Метод часто используется для связывания последовательности команд в единую цепочку. Конвейер обозначается следующим символом: |.

```
cat filename | grep something
```



Hotkeys

- Ctrl-C - посылает сигнал SIGINT, заканчивает процесс.
- Ctrl-D - завершение ввода с клавиатуры (EOF).
- Ctrl-Z - посылает сигнал SIGTSTP, приостанавливает процесс.



Задание

- Написать команду, которая в текущей директории считает кол-во файлов с расширением `.txt`.
- Команда `cal`. Используя её, вывести текущее число.
- В файле `ip.txt` записан IP адрес. Пропинговать компьютер и записать результат в `res.txt`, а ошибки — в `err.txt`.
- Создать файл, хранящий в себе список процессов, которые начинаются на `g`.
- Напишите команду, которая копирует содержимое файла в тот же файл.