

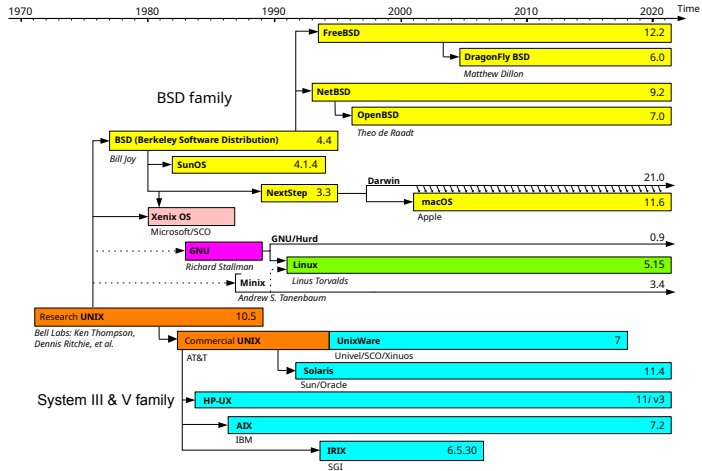
Основы работы с Unix

Инструментарий Современного Программиста

Иван Трепаков

NSU Sys.Pro

История Unix



Следует писать программы, которые

Следует писать программы, которые

- делают что-то одно и делают это хорошо

Следует писать программы, которые

- делают что-то одно и делают это хорошо
- взаимодействуют с другими программами

Следует писать программы, которые

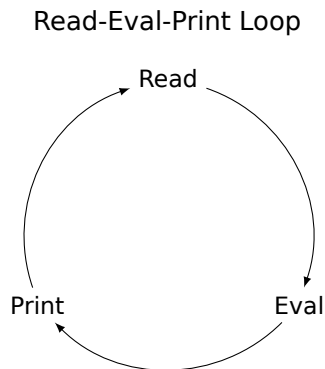
- делают что-то одно и делают это хорошо
- взаимодействуют с другими программами
- поддерживают стандартный ввод/вывод

- Терминал (*terminal emulator*)
 - GNOME Terminal
 - xterm
 - alacritty
 - kitty
 - PuTTY

- Терминал (*terminal emulator*)
 - GNOME Terminal
 - xterm
 - alacritty
 - kitty
 - PuTTY
- Командная оболочка (*command-line interface, cli* или *shell*)
 - bash (Bourne-Again SHell)
 - sh (Bourne shell)
 - zsh (Z shell)
 - fish (friendly interactive shell)

- Терминал (*terminal emulator*)
 - GNOME Terminal
 - xterm
 - alacritty
 - kitty
 - PuTTY
- Командная оболочка (*command-line interface, cli* или *shell*)
 - bash (Bourne-Again SHell)
 - sh (Bourne shell)
 - zsh (Z shell)
 - fish (friendly interactive shell)
- Интерпретатор команд

- Терминал (*terminal emulator*)
 - GNOME Terminal
 - xterm
 - alacritty
 - kitty
 - PuTTY
- Командная оболочка (*command-line interface, cli* или *shell*)
 - bash (Bourne-Again SHell)
 - sh (Bourne shell)
 - zsh (Z shell)
 - fish (friendly interactive shell)
- Интерпретатор команд
- Read-Eval-Print Loop (REPL)



Аргументы

```
$ echo Hello World!
```

```
Hello World!
```

```
$ echo Hello      World!
```

```
Hello World!
```

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

```
Hello      World!
```

```
$ echo 'Good      day!'
```

```
Good      day!
```

```
$ echo See\ \ \ you\ \ later
```

```
See      you      later
```

Аргументы

```
$ echo Hello World!  
Hello World!  
$ echo Hello      World!  
Hello World!  
$ echo "Hello      World!"  
Hello      World!  
$ echo 'Good      day!'  
Good      day!  
$ echo See\ \ \ you\ \ later  
See      you      later
```

Подстановка

```
$ echo "$PATH"  
/usr/local/bin:/usr/bin:/bin  
$ echo '$PATH'  
$PATH  
$ echo $PATH  
/usr/local/bin:/usr/bin:/bin  
$ export TEST="a  b  c"  
$ echo $TEST  
a b c  
$ echo "$TEST"  
a  b  c
```

Вывод команд

```
$ echo "Today: date"
Today: date
$ echo "Today: $date"
Today:
$ echo "Today: $(date)"
Today: Wed Aug 14 05:42:35 UTC 2024
```

Подстановка

```
$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin
$ echo '$PATH'
$PATH
$ echo $PATH
/usr/local/bin:/usr/bin:/bin
$ export TEST="a b c"
$ echo $TEST
a b c
$ echo "$TEST"
a b c
```


Redirect stdout

```
$ echo "Foo bar" > foo.txt  
$ cat foo.txt  
Foo bar
```

Redirect stdout with append

```
$ echo "Baz" >> foo.txt  
$ cat foo.txt  
Foo bar  
Baz
```

Redirect stdin

```
$ cat < foo.txt  
Foo bar  
Baz
```

Redirect both stdin and stdout

```
$ cat < foo.txt > bar.txt  
$ cat bar.txt  
Foo bar  
Baz
```

Redirect stderr

```
$ date foo
date: invalid date 'foo'
$ date foo >out.txt
date: invalid date 'foo'
$ cat out.txt
$ date foo 2>err.txt
$ cat err.txt
date: invalid date 'foo'
```

Redirect stderr to stdout

```
$ date foo >out.txt 2>&1
$ cat out.txt
date: invalid date 'foo'
```

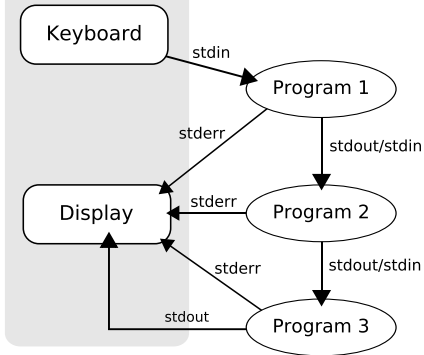
Redirect both stdout and stderr

```
$ date foo >out.txt 2>err.txt
$ cat out.txt
$ cat err.txt
date: invalid date 'foo'
```

Конвейер (Pipeline)

```
$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin
$ echo "$PATH" | sed 's:/:/\n/g'
/usr/local/bin
/usr/bin
/bin
$ echo "$PATH" | sed 's:/:/\n/g' | sort
/bin
/usr/bin
/usr/local/bin
```

Text terminal



Композиция программ

Код возврата (Exit code)

```
$ date
Sat Sep 7 02:37:24 UTC 2024
$ echo $?
0
$ date foo
date: invalid date 'foo'
$ echo $?
1
```

Безусловное исполнение

```
$ date foo; echo $?
date: invalid date 'foo'
1
```

Условное исполнение

```
$ echo "Hello" && date foo
Hello
date: invalid date 'foo'
$ echo $?
1
$ date foo && echo "Hello"
date: invalid date 'foo'
$ echo $?
1
$ date foo || echo "Hello"
date: invalid date 'foo'
$ echo $?
1
```

Зарезервированные директории

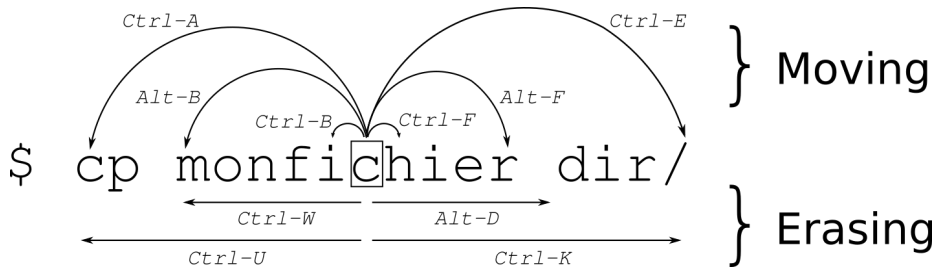
- / — корневая директория
- ~ — домашняя директория пользователя
- . — текущая директория
- .. — родительская директория

Права доступа

- d/./l — вид файла (директория, обычный файл, символическая ссылка)
- r — (*Read*) доступ на чтение
- w — (*Write*) доступ на запись
- x — (*eXecute*) доступ на исполнение

Команды

- pwd
- ls
- cd
- mkdir
- cp
- mv
- rm
- find



Q&A