

02. Основы работы с командной строкой

- 1 Интерфейс командной строки
- 2 Типы команд
- 3 Навигация по дереву каталогов
- 4 История команд и автодополнение

Раздел 1

Интерфейс командной строки

- **ТПИ - текстовой пользовательский интерфейс** (Text user interface, TUI; также Character User Interface, CUI) - Разновидность интерфейса пользователя, использующая при вводе-выводе и представлении информации исключительно набор буквенно-цифровых символов и символов псевдографики.
- **Интерфейс командной строки** (Command line interface, CLI) - Частный случай ТПИ. Способ взаимодействия между человеком и компьютером путём отправки компьютеру команд, представляющих собой последовательность символов. Команды интерпретируются с помощью специального интерпретатора - **командного интерпретатора**

- Графический интерфейс пользователя (ГИП), графический пользовательский интерфейс (ГПИ) (Graphical User Interface, GUI)
Система взаимодействия с пользователем, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т. п.)

Командный интерпретатор

- **Командный интерпретатор, интерпретатор командной строки -**
Компьютерная программа, часть операционной системы, обеспечивающая базовые возможности управления компьютером посредством интерактивного ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов.
- **DOS: COMMAND.COM**
- **Windows: cmd.exe, PowerShell**
- **Unix: sh, bash, csh (C shell), ksh (Korn shell), zsh (Z shell)**

- Интерактивный режим работы - возможность ввода команд интерактивно, взаимодействуя с интерфейсом командной строки
- Приглашение ввода команд - заканчивается обычно символами \$ или #. Говорит о том, что командный интерпретатор готов к обработке пользовательских команд (работает в интерактивном режиме).

[sysadmin@altwks:/etc]\$

- Пакетный режим работы - поддерживает написание и выполнение скриптов - для автоматизации однотипных/периодических задач.

Роль командного интерпретатора в системе

- Формирует интерфейс командной строки - позволяет пользователю запускать другие программы и, тем самым решать свои задачи в рамках ОС
- Запускается при входе пользователя в систему - является первым процессом, запускаемым при входе пользователя в систему (при входе посредством алфавитно-цифрового терминала)
- Формирует пользовательское окружение - командный интерпретатор формирует пользовательское окружение - среду для выполнения команд пользователя (переменные, алиасы)

Разновидности командных интерпретаторов

- **sh** - Bourne Shell - классический командный интерпретатор UNIX-систем, предоставляет основные возможности интерактивной работы, как правило используется в пакетном режиме, т.к. поддерживается практически повсеместно, скрипты, написанные для обработки средствами **sh** являются максимально переносимыми между системами
- **bash** - Bourne again shell - усовершенствованная и модернизированная вариация командной оболочки Bourne shell. Одна из наиболее популярных современных разновидностей командной оболочки. Как правило, является командным интерпретатором по-умолчанию для пользователя в ОС Linux.

- **csh** - C shell - командная оболочка UNIX со встроенным скриптовым языком, разработанная Биллом Джоем, активным разработчиком BSD UNIX и создателем редактора vi, в 1979 году. Скриптовый язык не уступает Bourne shell по мощности, но отличается синтаксисом.
- **ksh** - Korn shell - командная оболочка UNIX, разработана Дэвидом Корном (AT&T) в 1980-х годах. Имеет полную обратную совместимость с Bourne shell и включает в себя возможности C shell.
- **zsh** - Z shell - одна из современных командных оболочек UNIX, использующаяся непосредственно как интерактивная оболочка, либо как скриптовый интерпретатор. Zsh является расширенным bourne shell с большим количеством улучшений

Стандартный синтаксис POSIX.2

- **Стандартный синтаксис командной строки** - определен во 2ой части стандарта POSIX, поддерживается всеми командными интерпретаторами

```
$ ls -l -rt --color=always -I "*.mp3" /tmp /srv
```

- Пример командной конструкции соответствующей **стандарту POSIX.2**

\$

Приглашение к вводу команд

`ls`

Команда

`-l`

Одиночная односимвольная опция. Принято начинать с одного дифиса. Длинный вывод (атрибуты файлов).

`-rt`

Две односимвольные опции, состыкованы. Сортировка по времени изменения и обратная (reverse) сортировка.

--color=always

Многосимвольная опция. Принято начинать с двух дефисов.

-I "* .mp3"

Игнорировать файлы в соответствии с заданным шаблоном

/tmp /srv

Параметры - объекты, в отношении которых будет выполняться команда.

- Разделители команд, опций, параметров - пробельные символы. Один либо несколько пробелов.
- **LF** - символ завершения командной конструкции - перенос строки. Формируется драйвером терминала при нажатии клавиши **Enter**. Получение символа переноса строки говорит командному интерпретатору о необходимости начать выполнять введенную командную конструкцию.

Раздел 2

Типы команд

Типы команд

- Типы команд
 - внутренние команды командного интерпретатора
 - внешние команды
 - синонимы команд
- Команда **type** - позволяет определить типы команд

```
$ type pwd
pwd is a shell builtin
$ type ls
ls is aliased to `ls --color=auto'
$ type find
find is /usr/bin/find
```

Внутренние команды командного интерпретатора

Внутренние команды реализованы в самом командном интерпретаторе

```
$ type pwd  
pwd is a shell builtin  
$ type cd  
cd is a shell builtin  
$ type echo  
echo is a shell builtin  
$ type -a echo  
echo is a shell builtin  
echo is /bin/echo
```

Внешние команды

Реализованы в виде отдельных исполняемых файлов

- команда **which** - расположение исполняемого файла команды в системе

```
$ type cal  
cal is /usr/bin/cal  
$ type find  
find is /usr/bin/find  
$ which bash  
/bin/bash
```

Алиасы (псевдонимы) команд

Обычно короткие, более удобные имена длинным командам

```
$ alias  
alias grep='grep --color=auto'  
alias ls='ls --color=auto'  
$ type -a ls  
ls is aliased to `ls --color=auto'  
ls is /bin/ls
```

Задание алиасов

```
$ alias mycal="cal 2020"  
$ mycal
```

- Обычно производится в файлах инициализации командного интерпретатора

.bashrc
.bash_aliases

Раздел 3

Навигация по дереву каталогов

ls

- **-lh**

Длинный вывод с человеческими единицами

- **-a**

Включая скрытые файлы

- **-d**
Информация по каталогу, а не по содержимому
- **-tr**
Сортировка по времени в обратной последовательности
- **-S**
Сортировка по размеру

cd

- -

Ранее используемый каталог

- ..

На уровень вверх

- ~

Домашний каталог текущего пользователя

\$ pwd

Отображение текущего каталога

mc

\$ mc

Двухпанельный менеджер файлов

\$ mcedit

Встроенный в MC редактор файлов

Раздел 4

История команд и автодополнение

История команд

- **Ранее введенные** пользователем команды - хранятся в памяти командного интерпретатора

Перемещение по истории команд

*В приглашении командного интерпретатора -
стрелка вверх / стрелка вниз*

\$ history

*Отображение хранимой истории команд с
порядковыми номерами*

\$ history 3

Указание количества отображаемых команд

```
$ !-3
```

Выполнение команды из истории запуск Зьей команды “снизу” истории

```
$ !ls
```

Запуск команды по имени - запускается последняя команда с данным именем и с теми параметрами, с которыми она вызывалась ранее

Поиск по истории команд

- **Ctrl+R**
Поиск по истории команд

Персистентное хранение истории команд

- Файл **.bash_history** - хранение истории команд в домашнем каталоге пользователя. Запись производится при завершении сеанса командного интерпретатора. Чтение из файла в память - при запуске.

- **HISTSIZE**
Переменная окружения, определяет объем хранимой истории (кол-во команд)
- **HISTFILESIZE**
Переменная окружения, определяет объем хранимой истории (размер файла истории)

Дополнение имён в командном интерпретаторе

- Клавиша **Tab** позволяет упростить ввод команд за счет дополнения:
 - имена команд/функций/псевдонимов
 - имена файлов
 - имена переменных
 - опции/аргументы
- Если однозначно предполагаемое для дополнения имя не определить, **Tab** не выводит ничего.
- Повторное нажатие **Tab** - выводит возможные варианты (если есть)