



# Operating System

## Лабораторная работа 1

**SCE**

המכללה האקדמית להנדסה סמי שמעון

תתחבר לרעיונות גדולים

באר שבע | אשדוד ■ [www.sce.ac.il](http://www.sce.ac.il) \*מהנדס

# ?Что такое операционная система



- Промежуточный слой между аппаратным обеспечением и программой.
- Обеспечивает легкий доступ для пользователей или программистов.
- Примеры операционных систем?  
Windows, Linux, Android, DOS, IOS .....

# Типы файлов в Linux

Поскольку понятие файла применяется к достаточно разнородным вещам (файл как таковой, физические устройства, каталоги и т. п.), поневоле возникает разделение файлов на типы. В Linux существует шесть типов файлов:

- файл;
- каталог;
- файл устройства;
- канал (FIFO, PIPE);
- символическая или мягкая ссылка (soft link);
- сокет (Socket).

# Основные команды

**pwd**- Команда выводит на экран пользователю полное имя текущей папки.

**ls** – Просмотр всех существующих файлов и папок в текущем каталоге.

**ls -l** – Просмотр разрешений, размера, даты создания и т. д.

**ls -lr** – Просмотр папок и подпапок.

## Разрешения:

- Каждый файл имеет права доступа для трех типов пользователей.

u – user

g – group

o – others

Когда a – all относится ко всем трем.

Каждая группа имеет три разрешения:

r - чтение

w - запись

x - запуск

Они будут отображаться перед файлом в последовательности из 10 символов, при этом первый символ будет символизировать тип файла:

1. l - Ссылка.
2. d - папка.
3. '-' - любой другой файл.

Первые три, идущие после него, принадлежат владельцу, а три после них - группе, а три последних для всех.

| u | g | o |

- rwxrwxrwx file

# ИЗМЕНЕНИЯ ПРАВ ДОСТУПА – КОМАНДА `chmod`

**`chmod <symbolic_mod_list> filename`**

`chmod [ugo] [+ -=] [rwx] filename`

Команда получит 2 параметра:

- 1) Тип авторизации (в виде символов / цифр)
- 2) Имя файла

*\* Мы поговорим о двух методах предоставления разрешений - символьном и числовом.*

# разрешения в команде chmod:

Permissions	Decimal	Binary
none	0	000
x--	1	001
-w-	2	010
wx-	3	011
--r	4	100
r-x	5	101
-rw	6	110
rwX	7	111

```
lrwxrwxrwx 1 root root 4 Авг 31 10:15 [ -> test
-rwxr-xr-x 1 root root 93 Янв 22 2010 4odb_clean
-rwxr-xr-x 1 root root 93 Янв 22 2010 4odb_clear
-rwxr-xr-x 1 root root 95 Янв 22 2010 4odb_create
-rwxr-xr-x 1 root root 97 Янв 22 2010 4odb_destroy
-rwxr-xr-x 1 root root 89 Янв 22 2010 4odb_dig
-rwxr-xr-x 1 root root 93 Янв 22 2010 4odb_grant
-rwxr-xr-x 1 root root 97 Янв 22 2010 4odb_metadig
-rwxr-xr-x 1 root root 99 Янв 22 2010 4odb_odmsdump
```

## разрешения в команде chmod:

`chmod [ugo] [+ -=] [rwx] file`

+ добавляет разрешение

- удаляет разрешение

= удаляет все предыдущие разрешения и устанавливает новые

### ИЗМЕНЕНИЯ ПРАВ ДОСТУПА - УПРАЖНЕНИЕ

Дается следующее разрешение: r-x r-x -w- file

Его следует изменить с помощью следующих команд:

1. `Chmod ug-r file`
2. `Chmod 651 file`
3. `Chmod u=r file`
4. `Chmod u+x,g-r file`

## ИЗМЕНЕНИЯ ПРАВ ДОСТУПА - УПРАЖНЕНИЕ

Дается следующее разрешение: r-x r-x -w- file

Решение:

```
# chmod ug-r file  
- --x --x -w-
```

```
# chmod 651 file  
- rw- r-x --x
```

```
# chmod U=r file  
- r-- r-x --x
```

```
# chmod U+x,g-r file  
- r-x --x --x
```

# Основные команды

- **man** - Отображает информацию о каждой команде или функции папки, которую вы хотите использовать  
**например:** для получения информации о команде `ls` наберите `man ls`
- **clear** - Очистка экрана
- **date** - Печатает системную дату на экране.
- **cal <month> <year>** - Календарь на определенный месяц.
- **mkdir** - создание новой библиотеки.  
`mkdir folder` - создаёт папку в текущей папке.  
`mkdir -p folder1/folder2` создание полного пути к файлу.

# Основные команды

## cp <source> <destination>

- Для копирования файлов может быть использована команда **cp** (сокращение от **copy**).
- Команда **cp** умеет копировать не только отдельный файл, но и набор файлов, и даже целиком директорию вместе со всеми входящими в нее поддиректориями (рекурсивное копирование).
  - Если < **source** > и < **destination** > являются файлами: команда копирует файл < **source** > или список файлов в файл < **destination** >. Старый файл или список файлов (< **source** >) не был удален. После выполнения команды в текущем каталоге остаются два файла (< **source** >, < **destination** >) с одинаковым содержимым.
  - Если каталог < **destination** > и файл < **source** > или список файлов: команда копирует <источник> в существующий каталог < **destination** >. После выполнения команды в текущем каталоге и в каталоге < **destination** > находятся файлы < **source** >.
  - Если < **destination** > и < **source** > - каталоги: команда копирует (рекурсивное копирование) каталог или список каталогов (включая подкаталоги) (< **source** >) в каталог < **destination** >.

# Основные команды

## **mv <source> <destination>**

Если каталог **< destination >** и **< source >** - файлы: команда перемещает файл **< source >** в место **<destination >** (старый файл удаляется).

## **rm <file\_name>**

Удаляет файл с именем **< file\_name >** из текущего каталога (вы также можете удалить список файлов).

**rm -r < dir\_name >** Удалить папку и все ее содержимое.

## **rmdir <dir\_name>**

удалить пустую папку.

# Основные команды

- **cd** - Замена текущей библиотеки. Важно помнить, что после имени команды `cd` Вы должны поставить пробел перед именем каталога, который требуется переместить.

Использование команды без параметров приведет к переходу в главный каталог пользователя.

`cd /folder1/folder2` - перейти в определенную папку

`cd ..` – вернуться на один уровень назад.

# Основные команды

- **cat file\_name –**

- Команда cat используется для просмотра файлов.

cat file1 file2 - Вывести цепочку файлов на экране: file1 а затем file2.

## **cat > new\_file**

Создаётся новый текстовый файл.

Знак <- ввод в файл

например: ls -l > file.txt

Вывод команды вставляется как ввод в текстовый файл.

# Основные команды

- **wc file\_name**

Количество строк, слов и букв в содержимом файла.

## Options:

wc -c file // count letters

wc -l file // count lines

wc -w file // count words

# Основные команды

## FIND

**find <local or sub directory> <criteria> <action to be taken>** – Команда для поиска файла / папки в системе.

**<criteria>**

**: name-**

Поиск по имени файла

”example : find / -name “targil.c

”find / -name “\*.txt

\* Есть много вариантов поиска также по типу / размеру и т. д.

# Стандартный ввод / вывод и маршрутизация (redirection)

<b>redirections</b>	
<	Перенаправление вывода в файл
<<	Добавить в конец файла
>	Получение ввода из файла

## Standard error

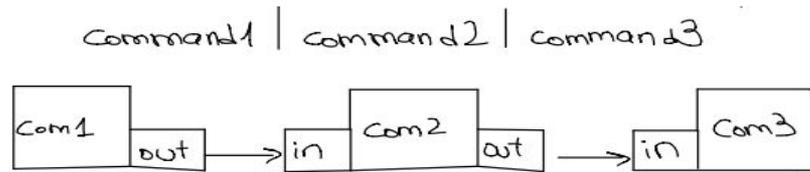
Сообщения об ошибках отображаются на экране (который используется и для стандартного вывода). Можно вывести информацию в файл с помощью символов “>&”

```
gcc f1 >& f2
```

В примере после компиляции файла f1 сообщения об ошибках не выводятся на экран, а перенаправляются в файл f2.

# Pipes

Используем символ | чтобы запустить цепочку команд. Pipe это цепочка команд, когда standard output одной команды будет standard input для следующей.



```
cat f1 | wc
```

В примере запускается команда `cat` на файл `f1`, которая отображает текстовый файл на экран. Pipe захватывает вывод и запускает команду `wc`, которая подсчитывает количество строк, слов и букв в файле. Результат команды `wc` мы видим на экране.

# Упражнения

1. Написать команду, которая записывает в файл FF.txt список всех текстовых файлов.
2. Написать команду, которая записывает в файл codes.txt список всех файлов с расширением “.c”

# nano

Этот редактор многие программы LINUX используют для написания различных текстов. Во время редактирования мы можем сохранить текущий файл, нажав Ctrl + X и выбрав имя файла, в котором мы хотим сохранить документ (или Enter, чтобы перезаписать существующий документ). Комбинация клавиш Ctrl + X будет использоваться нами для выхода из nano. Если мы не сохранили активный документ, нам будет предоставлена возможность сделать это. Для редактирования файлов в командной строке пишем:

**nano < File\_name >**

Где File\_name - это имя файла, который вы хотите отредактировать.

Например:

**нано myfile.txt**

Для редактирования откроется файл с именем myfile.txt. Если файл не существует в системе, nano откроет новый пустой файл с тем же именем.