



# Linux

## Basic command line skills

## **man (Your best friend!)**

---

man means manual pages, its like reading a manual of any home appliance. In software it means reading documentation. Good softwares have always a good documentation part.

\$ man <command>

\$ man clear

---

# command1 > file

---

> – redirects command output to file (instead of terminal). File will be rewritten if exists.

## Example:

```
$ ls -la > file
```

```
$ cat file
```

```
...
```

# command1 >> file

---

>> – redirects command output to file (instead of terminal). File will be upended if exists.

## Example:

```
$ ls -la /home/Linus >> file
```

```
$ ls -la /home/Mary >> file
```

```
$ cat file
```

```
...
```

# Commands

---

- The *apropos* command displays a list of all topics in the man pages (i.e., the standard manual that is built into Unix-like operating systems) that are related to the subject of a query.
- It is particularly useful when searching for commands without knowing their exact names.

\$ *apropos* rename

## ls (list items)

---

- The **ls** command ( lowercase L and lowercase S ) lists the contents of your current working directory.
  - **ls** by default does not show hidden files
  - to see all files use **ls -al**
  - **-al** is flag in this case
-

# ls -al output

**UserName** **HostName** **Current directory**

```
miroslav_andrushchenko@epam.com@ECSE001009FA:~$ ls -al
```

total 32

drwxr-xr-x	4	miroslav_andrushchenko@epam.com	users	4096	Jul	6	19:52	.
drwxr-xr-x	10	root	root	4096	Jul	6	19:50	..
-rw-----	1	miroslav_andrushchenko@epam.com	users	70	Jul	6	20:03	.bash_history
-r--r--r--	1	miroslav_andrushchenko@epam.com	users	220	Jul	6	19:50	.bash_logout
-rw-r--r--	1	miroslav_andrushchenko@epam.com	users	3771	Jul	6	19:50	.bashrc
drwx----	2	miroslav_andrushchenko@epam.com	users	4096	Jul	6	19:50	.cache
-r--r--r--	1	miroslav_andrushchenko@epam.com	users	655	Jul	6	19:50	.profile
drwxr-xr-x	2	miroslav_andrushchenko@epam.com	users	4096	Jul	6	19:50	.ssh

**FileType** **Directory** **Regular File** **FileName**

## **mkdir (make directory)**

---

- creates a directory
  - use -p to create as many directories.
  - `$ mkdir first/second/third -p`
  - `$ cd first/second/third`
-



## **cd (change directory)**

---

- changes the current directory
  - `cd .` to remain inside same directory
  - `cd ..` to go back one directory
-

## **~ Your Home Directory**

---

By typing `cd ~` in terminal it will take you to the home directory of current user

`$ cd ~`

`~student` means home directory of user with name 'student'

`$ cd ~root` – will take you to the home directory of the user 'root'

## **pwd (print working directory)**

For example, to find out the absolute pathname of your home-directory, type `cd` to get back to your home-directory and then type

```
$ pwd
```

it will print current directory

## **cp (copy)**

---

copies a file or entire directory to another path

If you want to copy files from another directory to current directory use a . for destination

```
$ cp <source_file> <destination_file>
```

Tip: Always read the error carefully and please don't freak out :)

## **mv (move file)**

---

move command renames or move a file

use -r flag to recursively move all files

```
$ mv file1.txt file2.txt
```

# Commands

---

## **clear**

---

clears your screen

\$ clear

# Commands

---

## **cat**

---

displays the content of files on screen.

```
$ cat file.txt
```

## **less**

---

less is similar to cat but it displays content one page at a time

```
$ less file.txt
```



# More and navigation

## SUMMARY OF LESS COMMANDS

Commands marked with \* may be preceded by a number, N.

Notes in parentheses indicate the behavior if N is given.

A key preceded by a caret indicates the Ctrl key; thus ^K is ctrl-K.

```
h H      Display this help.
q :q Q :Q ZZ  Exit.
```

---

## MOVING

```
e ^E j ^N CR * Forward one line (or N lines).
y ^Y k ^K ^P * Backward one line (or N lines).
f ^F ^V SPACE * Forward one window (or N lines).
b ^B ESC-v    * Backward one window (or N lines).
z             * Forward one window (and set window to N).
w             * Backward one window (and set window to N).
ESC-SPACE    * Forward one window, but don't stop at end-of-file.
d ^D         * Forward one half-window (and set half-window to N).
u ^U         * Backward one half-window (and set half-window to N).
ESC-) RightArrow * Left one half screen width (or N positions).
ESC-( LeftArrow  * Right one half screen width (or N positions).
F             Forward forever; like "tail -f".
r ^R ^L       Repaint screen.
R             Repaint screen, discarding buffered input.
```

---

Default "window" is the screen height.

Default "half-window" is half of the screen height.

---

## head

---

head is a short version of less command. It prints only first 10 lines.

You can set the number of lines by giving a -<number> flag e.x

```
$ head -5 file.txt
```

## **tail**

---

tail is opposite to head, it shows last 10 lines of the file.

## searching within text

---

You can use less command to search within text. Do you remember what less command do?

After executing the less command use / and type your required text to search

```
$ less file.txt
```

```
/science
```

## **grep (don't ask what that's called :) )**

This command search a file for specific word or a pattern. for example

grep Science file.txt will search string Science in file.txt

grep command by default is case sensitive

## **wc (word count)**

---

counts number of words in a given file.

to find number of lines set the -l flag

```
$ wc -l file.txt
```

# command1 | command2

---

| – transfer command1 output to command2 input

## Example:

```
$ ls -la | grep mp3
```

```
$ ps -ef | less
```

## **rm (remove)**

---

rm removes a file or entire directory with -r flag

```
$ rm <file>
```

```
$ rm <directory> -r
```



1. Войти под своим логином/паролем
2. Создать каталог для упражнений и перейти в него
3. Создать файлы f1 и f2, а также директорию dir1 и в ней файл dir2
4. Скопировать dir1 в dir2 со всеми файлами
5. Удалить dir1
6. Скопировать файл /etc/services в свой каталог
7. Вывести все строки, содержащие HTTP
8. Посчитать количество строчек в нем
9. Исследовать команды more, less, tail, head на файле services
10. Исследовать команды cal, bc
11. Выяснить дату православной пасхи на 2051 год.

**THANK YOU**