

Bash Script (Linux commands)

Here you will see a summary of the common commands use in Linux (Unix) environment. We will just give examples of the ones we will use more frequently during the workshop. This commands can't be used in Windows environment unless you use a [Linux emulator](#). If you need information in the terminal about an specific Linux/Unix command you can type a command line on the terminal:

```
$ man ls
```

To escape from the manual, just type: *q*

File Commands

FILE COMMANDS

```
ls -al          =>Display all information about files/ directories
pwd             =>Show the path of current directory
mkdir directory-name  =>Create a directory
rm file-name     =>Delete file
rm -r directory-nam  =>Delete directory recursively
rm -f file-name    =>Forcefully remove file
rm -rf directory-name  =>Forcefully remove directory recursively
cp file1 file2    =>Copy file1 to file2
cp -r dir1 dir2    =>Copy dir1 to dir2, create dir2 if it doesn't exist
mv file1 file2     =>Rename source to dest / move source to directory
ln -s /path/to/file-name link-name  #Create symbolic link to file-name
touch file        =>Create or update file
cat > file        =>Place standard input into file
more file         =>Output contents of file
head file         =>Output first 10 lines of file
tail file         =>Output last 10 lines of file
tail -f file      =>Output contents of file as it grows starting with the
                  last 10 lines
gpg -c file       =>Encrypt file
gpg file.gpg      =>Decrypt file
wc               =>print the number of bytes, words, and lines in files
xargs            =>Execute command lines from standard input
```

Examples - working with directories:

```
$ ls
$ ls -al
$ ls -alrt
$ pwd
$ mkdir codes
$ mkdir temp temp1
$ mkdir temp2/ok
$ cat file.txt
$ cat file.txt file2.txt > fileOut.txt
$ rm file.txt
$ rm -r temp2/ok
$ rm -r temp2
$ df -h
```

Note: `rm *` it is a dangerous command that should be use with care, you might lose all your work!!

Examples - Create a File:**

Method 1: Using nano editor:

```
$ nano file.txt
```

Then you will get the screen editor where you can type in e.g. type `echo "Hello World!"` .

To save your changes and exit the editor type: **ctrl + x** and choose **Yes** to save your changes. **Method 2:** Using **cat** command.

```
$ cat >file2.txt  
"Hello Universe!"
```

To exit press: **ctrl + d**".

Directory

DIRECTORY TRAVERSE

cd ..	=>To go up one level of the directory tree
cd	=>Go to \$HOME directory
cd /test	=>Change to /test directory

Disk usage

DISK USAGE

df -h	=>Show free space on mounted filesystems
df -i	=>Show free inodes on mounted filesystems
fdisk -l	=>Show disks partitions sizes and types
du -ah	=>Display disk usage in human readable form
du -sh	=>Display total disk usage on the current directory
findmnt	=>Displays target mount point for all filesystem
mount device-path mount-point	=>Mount a device

File transfer and Remote Access

Remote_access

Search

SEARCH

grep pattern files	=>Search for pattern in files
grep -r pattern dir	=>Search recursively for pattern in dir
locate file	=>Find all instances of file
find /home/tom -name "index"	=>Find files names that start with "index"
find /home -size +10000k	=>Find files larger than 10000k in /home

System

SYSTEM	
uname -a	=>Displaylinux system information
uname -r	=>Display kernel release information
uptime	=>Show how long the system has been running + load
hostname	=>Show system host name
hostname -i	=>Display the IP address of the host
last reboot	=>Show system reboot history
date	=>Show the current date and time
cal	=>Show this month calendar
w	=>Display who is online
whoami	=>Who you are logged in as
finger user	=>Display information about user

Process Related

PROCESS RELATED	
ps	=>Display your currently active processes
ps aux grep 'telnet'	=>Find all process id related to telnet process
pmap	=>Memory map of process
top	=>Display all running processes
kill pid	=>Kill process with mentioned pid id
killall proc	=>Kill all processes named proc
pkill process-name	=>Send signal to a process with its name
bg	=>Resumes suspended jobs without bringing them to foreground
fg	=>Brings the most recent job to foreground
fg n	=>Brings job n to the foreground

Virtual terminal (Screen)

Screen command	Description
<code>screen -S name_of_terminal</code>	Assigning name to the virtual terminal or screen session.
<code>screen -ls</code>	List all the virtual sessions or screens opened.
<code>screen -X -S name_of_terminal quit</code>	Kill an specific virtual terminal.
<code>screen -r name_of_terminal</code>	Attach to the virtual terminal or screen.
Press "Ctrl-A" and "d"	Detach from virtual terminal or screen.
Press "Ctrl-A" and "K"	This command will leave and kill the virtual terminal or screen
Press "Ctrl-A" and "n"	Switching to the next virtual terminal or screen.
Press "Ctrl-A" and "p"	Switching to the previous virtual terminal or screen

Reference