Linux Important Commands

```
pwd
mkdir
clear
```

Open a Terminal

```
xterm
uxterm
lxterminal
terminator
gnome-terminal
```

Listing

```
ls
```

to list with permission details

```
ls -ll
```

to list with with hidden files

```
ls -la
```

to list sub-content in tree view

```
tree -L <depth>
```

where <depth> is an integer defines depth of listing in the tree

---

extract files: `tar  -x… filename`
compress files: `tar -c... <foldername> <targetname>`

Directory Size

To know size of directory

```
du -s -h directory_name
```

-s: count silent,   -h: to readable for human (in M or G)

If you want to check sizes of some subdirectories in a directory:

`du -s -h *` (or `du -sh` or `du -hs`)

check sizes of some subdirectories in a directory in descending order:

```
du -ks * | sort -nr | cut -f2 | xargs -d '\n' du -sh
```

Disk Space

```
df -h
```

To make repartition for the disk spaces use gparted tool.

Delete Directory

```
rm -r directory_name
```

Search

To search for a word: grep

```
grep 'word' filename
grep 'word' file1 file2 file3
grep 'string1 string2'  filename
grep --color 'data' fileName
```

To search for a file

```
find / -type f -name "file_name"
```

where / searches for the whole disk, “” write file name inside.

To know if a package is installed or not

```
ldconfig -p | grep libraryname
```

Copy

** To copy a file

```
cp /path/to/original/file /path/to/copied/file
```

** To copy a directory

```
cp -r /path/to/original/directory /path/to/copied/directory
```

Copy the file "foobar.txt" from a remote host to the local host

```
$ scp your_username@remotehost.edu:foobar.txt /some/local/directory
```

Copy the file "foobar.txt" from the local host to a remote host

```
$ scp foobar.txt your_username@remotehost.edu:/some/remote/directory
```

Copy the directory "foo" from the local host to a remote host's directory "bar"

```
$ scp -r foo your_username@remotehost.edu:/some/remote/directory/bar
```

Copying the files "foo.txt" and "bar.txt" from the local host to remote host

```
$ scp foo.txt bar.txt your_username@remotehost.edu:~
```

Copy the file "foobar.txt" from remote host "rh1.edu" to remote host "rh2.edu"

```
$ scp your_username@rh1.edu:/some/remote/directory/foobar.txt \
your_username@rh2.edu:/some/remote/directory/
```

Add a path permanently to **PATH environment variables**

```
sudo gedit /etc/environment
```

add your path in that file:
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/path/to/your/new/bin"
then restart your machine.

This can be defined also in .bashrc file or .profile or in etc/profile

---

Multiple Execution

```
A; B     Run A and then B, regardless of success of A
A && B   Run B if A succeeded
A || B   Run B if A failed
A &      Run A in background
```

Permissions
Type ls –l to list the content of current directory, the output should be listed like that

```
total 24
-rwxrwxrwx 1 ali  Projectmem  293    Mar  6 14:53  compileTool.sh
drwxr-xr-x 2 ali  Projectmem  4096   Oct 17  2016  Desktop
drwxr-xr-x 3 ali  Projectmem  4096   Mar 17 16:22  gitprj
-rwxrwxrwx 1 ali  Projectmem  82     Feb 22 11:06  OpenWorkSpace.sh
-rwxrwxrwx 1 ali  Projectmem  240    Mar  7 08:09  RunLicenses.csh
drwxr-xr-x 2 ali  Projectmem  4096   Oct 19  2016  Test
```

r refers to read - w refers to write - x refers to execute - d refers to directory.

In this example “compileTool.sh” has rwx permission for the user “ali” and rwx permission for anyone in the group “Projectmem” and rwx permission for anyone using the machine.
This is denoted in binary as 111 for “ali”, 111 for “Projectmem”, 111 for anyone.
In decimal we can say it has 777.
Gitprj directory in this example has permission 775.

User & Group Accounts

To know current user:

```
whoami
```

To create user:

```
sudo adduser NewUser
sudo passwd NewUser
```

To give the user "foo" unlimited access to root privileges, edit /etc/sudoers and add the line

```
foo ALL = (ALL:ALL) ALL
```

To give access without passwords

```
foo ALL = NOPASSWD: ALL
```

To switch between users accounts:

```
su - NewUser
```

To return to the first user:

```
exit
```

To switch to root user

```
sudo su -
```

To create new group

```
sudo groupadd NewGroup
```

To include user in a group as his primary group

```
sudo usermod -g GroupName UserName
```

To include user in a group as his secondary group

```
sudo usermod -a -G GroupName UserName
```

Note that you have to log off and login again to see effect

SSH

To connect to a remote host through SSH port (22):

```
ssh <user_name>@<remote_host>
```

To connect to a remote host through SSH port without need to re-type password everytime:

* Generate a passphrase in the local host:

```
ssh-keygen -t rsa
```

* Generated passphrase can be found at:

```
~/.ssh/id_rsa.pub
```

* Send the generated passphrase to the remote host:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub <user_name>@<remote_host>
```

alternatively (instead of last command), you can copy content of id_rsa.pub from local host and insert it in `~/.ssh/authorized_keys` in remote host

---

Networking

IP Address can be assigned dynamically or statically.

* For dynamic IP

```
vi /etc/network/interfaces
```

add the following line for the target interface

```
auto eth0
iface eth0 inet dhcp
```

* For static IP

```
vi /etc/network/interfaces
```

add the following line for the target interface

```
auto eth0
iface eth0 inet static
address 192.168.1.102
netmask 255.255.255.0
gateway 192.168.1.10
```

These settings will become active after rebooting or restarting the service

```
sudo /etc/init.d/networking restart
```

Building Cronjob

to build a cronjob in linux, open terminal window, type

```
crontab -e
```

this shall open editor to add your task that you want to run. Adding a job is a simple line of code. First you have to decide when do yo need your job to run (hourly, daily, monthly, ...)

```
15 *   *   *   * /path/to/file/file.sh >> /path/to/output/outputfile.txt
|  |   |   |   |-----> Day of Week (0~7, 0 is sunday, 7 is also sunday)
|  |   |   |--------> Month (1~12)
|  |   |-----------> Day of Month (1~31)
|  |--------------> Hour (0~23)
|-----------------> Minute (0~59)
# note that 15 here means 15 minutes.
```

if you want your cron job to run a file using an application, type command of your application then path to executable file. For instance if you need to run a python file you can run it as follows:

```
* * * * * python /absolute/path/to/script.py >>/tmp/out.txt 2>&1
```

EOF