

# BITS Pilani - Hyderabad Campus

## Operating Systems (CS F372)

### Tutorial 1

**The objective of this tutorial is to learn some Linux OS concepts and commands.**

1. Run the command `ps` with various options and explain the output (important fields)

`$ ps` ( to read manual page for any command, run *man <command>*)

`$ ps -e`

You will see that each process has a unique id (called *pid*). Use pipe ( `|` ) to send output to 'more' to view output one screen at a time. What are Unix pipes?

`$ ps -ef`

`$ pstree` (to see process tree. Look at the tree and find the bash process. Go up the tree by using `grep` to look at parents recursively. You can see that init process is at the root of this tree.)

`$ ps -ef | grep [parent pid]..`

2. Run a program and use `kill pid` to terminate it. E.g. open `vi` in the background ( tell them about background processes), and terminate it by its pid.

`$ sleep 100&` (command to sleep for 100 seconds)

`[1] 12175` ← shows pid

`$ kill 12175` (use *kill -9 <pid>* if this doesn't work)

3. Create hard link to a file. What is a hard link? Need for hard links?

Assume a file `1.c` is present with some text content; run the following commands

`$ ln 1.c 2.c`

`$ ls -li`

After running the above command, you will see two files with same size and **inode** number (what is inode?)

Now, delete `2.c`, and again run "`ls -li`"

Repeat the above process but now delete the original file instead of the new one.

Nothing changes except the file name. Directories have at least 2 links. Why?

4. Create symbolic link to a file. What is a symbolic link *aka* soft link *aka* sym link?

`$ ln -s 1.c 2.c`

`$ ls -li`

The output is different this time, `2.c` is shown as a symbolic link to `1.c` (look at the first letter in the permission field, and also look at the end of the output line corresponding to `2.c`). The inode is also different. Attempting to read content of `2.c`

will result in content of 1.c being read instead. The symbolic link file just contains the name of the file it is pointing to; you can see this by running:

```
$ readlink 2.c
```

it will show 1.c as its content

Copying the symbolic link to another file copies the content of the original file the symbolic link is pointing to.

```
$ cp 2.c 3.c
```

```
$ ls -li to see the inodes
```

Now delete the symbolic link.

```
$ rm 2.c
```

```
$ ls -li
```

Now delete all .c files (use *rm <filename>*)

Repeat the above process of creating symbolic link, but now delete the original file instead of the symbolic link.

```
$ ln -s 1.c 2.c
```

```
$ ls -li
```

```
$ rm 1.c
```

The symbolic link remains but the file it points to has become invalid. Try copying 2.c to another file....will result in an error.

```
$ cp 2.c 3.c
```

```
cp: cannot stat '2.c': No such file or directory
```

Why are symbolic links needed?

5. What is sudo? Use sudo for running privileged commands e.g .

```
$ sudo apt-get install vim
```

[You may need to enter password]

It will download and install vim from the repository on the Internet.

To uninstall, run

```
$ sudo apt-get remove vim
```

6. Boot process: Linux booting comprises the following steps:

- BIOS (from ROM)
- Boot Loader
  - MBR (master boot record – in the first sector, used to load GRUB)
  - GRUB (Grand Unified Boot Loader)
- Kernel
- Init
- Runlevel scripts

Read more at : [https://en.wikipedia.org/wiki/Linux\\_startup\\_process](https://en.wikipedia.org/wiki/Linux_startup_process)