Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 02

Course code: ICT-3110

Course title: Operating Systems Lab

Date of Performance :

Date of Submission : 30/09/2020

Submitted by

Name: Humayun Kabir

ID:IT-18008

3rd year 1stsemester

Session: 2017-2018

E Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No : 02

Experiment Name : Basic Command of Linux operating System.

Theory :

Linux is a Unix-Like operating system. All the Linux/Unix commands are run in the terminal provided by the Linux system. This terminal is just like the command prompt of Windows OS. Linux/Unix commands are *case-sensitive.* The terminal can be used to accomplish all Administrative tasks. This includes package installation, file manipulation, and user management. Linux terminal is user-interactive. The terminal outputs the results of commands which are specified by the user itself. Execution of typed command is done only after you press the Enter key.

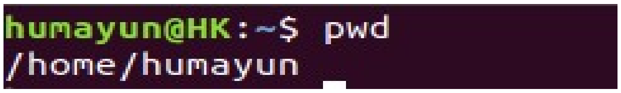
15 commands of Linux Operating System -

1. pwd
2. ls
3. cd
4. mkdir & rmdir
5. touch
6. man & --help
7. df
8. echo
9. cat
10. nano, vi, jed
11. uname
12. hostname

Working Process :

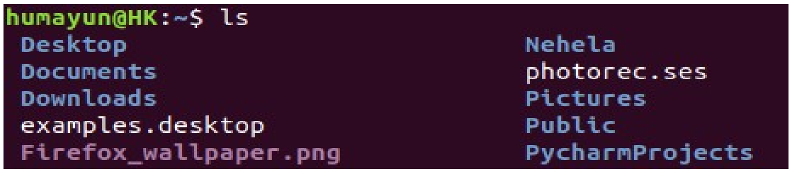
The operation of Linux Basic Command -

1. **pwd** — When we first open the terminal, we are in the home directory of our user. To know which directory we are in, we can use the **"pwd"** command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/). The user directory is usually something like "/home/username".



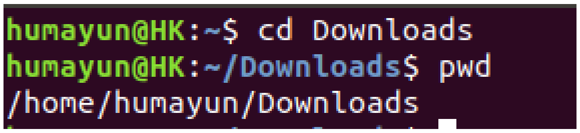
```
humayun@HK:~$ pwd
/home/humayun
```

2. ls — we use the **"ls"** command to know what files are in the directory we are in. We can see all the hidden files by using the command **"ls -a"**.



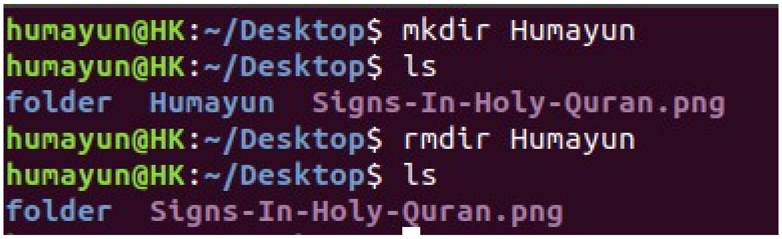
```
humayun@HK:~$ ls
 Desktop                              Nehela
 Documents                            photorec.ses
 Downloads                            Pictures
 examples.desktop                     Public
 Firefox_wallpaper.png                PycharmProjects
```

3. cd — We use the **"cd"** command to go to a directory. For example, if we are in the home folder, and we want to go to the downloads folder, then we can type in **"cd Downloads"**. Remember, this command is case sensitive, and we have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine we have a folder named "Raspberry Pi". In this case, when we type in **"cd Raspberry Pi"**, the shell will take the second argument of the command as a different one, so we will get an error saying that the directory does not exist. Here, we can use a backward slash. That is, we can use **"cd Raspberry\ Pi"** in this case. Spaces are denoted like this: If we just type **"cd"** and press enter, it takes you to the home directory. To go back from a folder to the folder before that, you can type "**cd** .." . The two dots represent back.



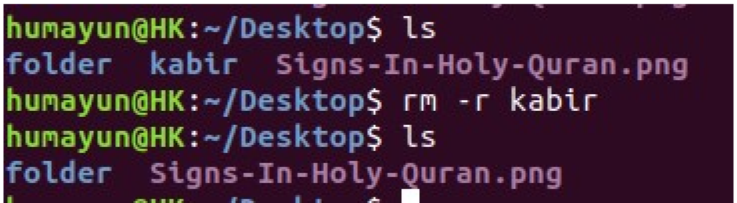
```
humayun@HK:~$ cd Downloads
humayun@HK:~/Downloads$ pwd
/home/humayun/Downloads
```

4. mkdir & rmdir — We use the **mkdir** command when we need to create a folder or a directory. For example, if we want to make a directory called "DIY", then we can type **"mkdir DIY**". Remember, as told before, if we want to create a directory named "DIY Hacking", then we can type "mkdir **DIY\ Hacking**". Use **rmdir** to delete a directory. But **rmdir** can only be used to delete an empty directory. To delete a directory containing files, use **rm**.



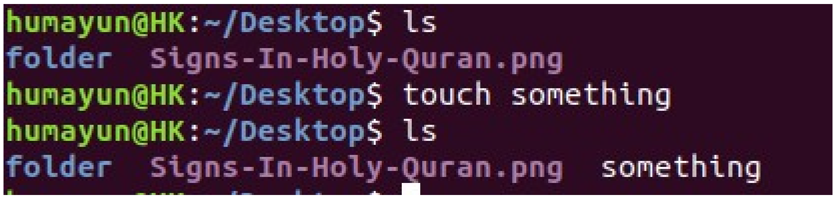
```
humayun@HK:~/Desktop$ mkdir Humayun
humayun@HK:~/Desktop$ ls
folder  Humayun  Signs-In-Holy-Quran.png
humayun@HK:~/Desktop$ rmdir Humayun
humayun@HK:~/Desktop$ ls
folder  Signs-In-Holy-Quran.png
```

5. rm - We use the **rm** command to delete files and directories. Use "**rm -r**" to delete just the directory. It deletes both the folder and the files it contains when using only the **rm** command.



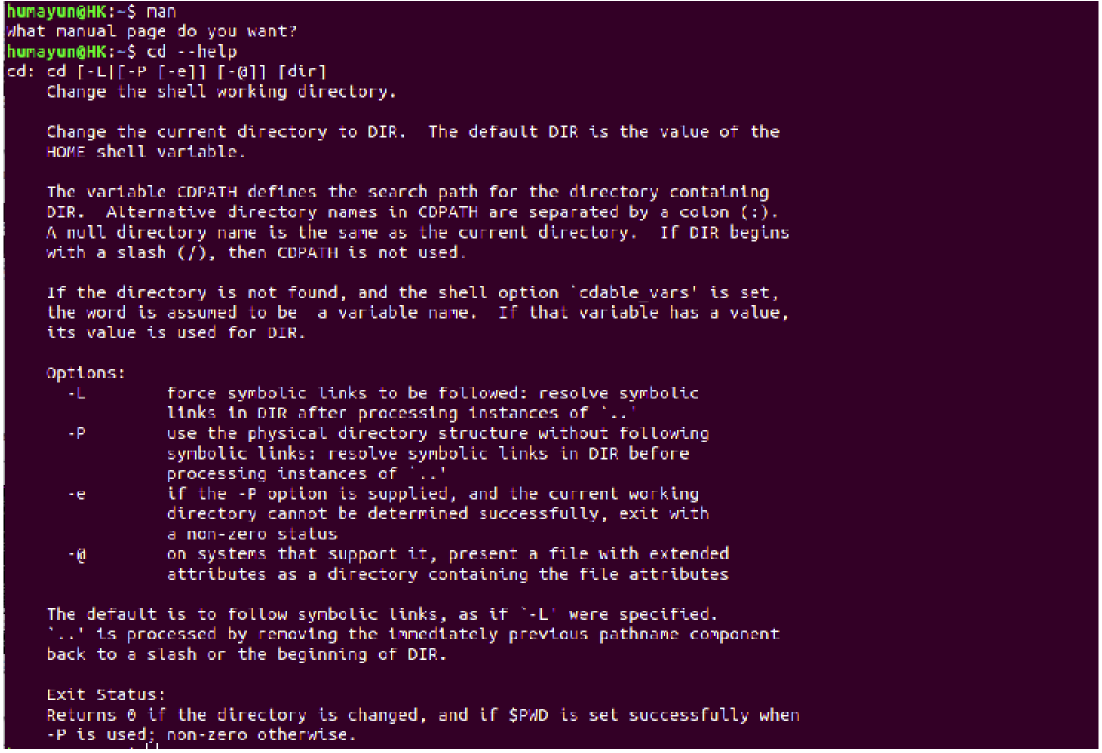
```
humayun@HK:~/Desktop$ ls
folder  kabir  Signs-In-Holy-Quran.png
humayun@HK:~/Desktop$ rm -r kabir
humayun@HK:~/Desktop$ ls
folder  Signs-In-Holy-Quran.png
```

6. touch — The **touch** command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, "**touch new.txt**".



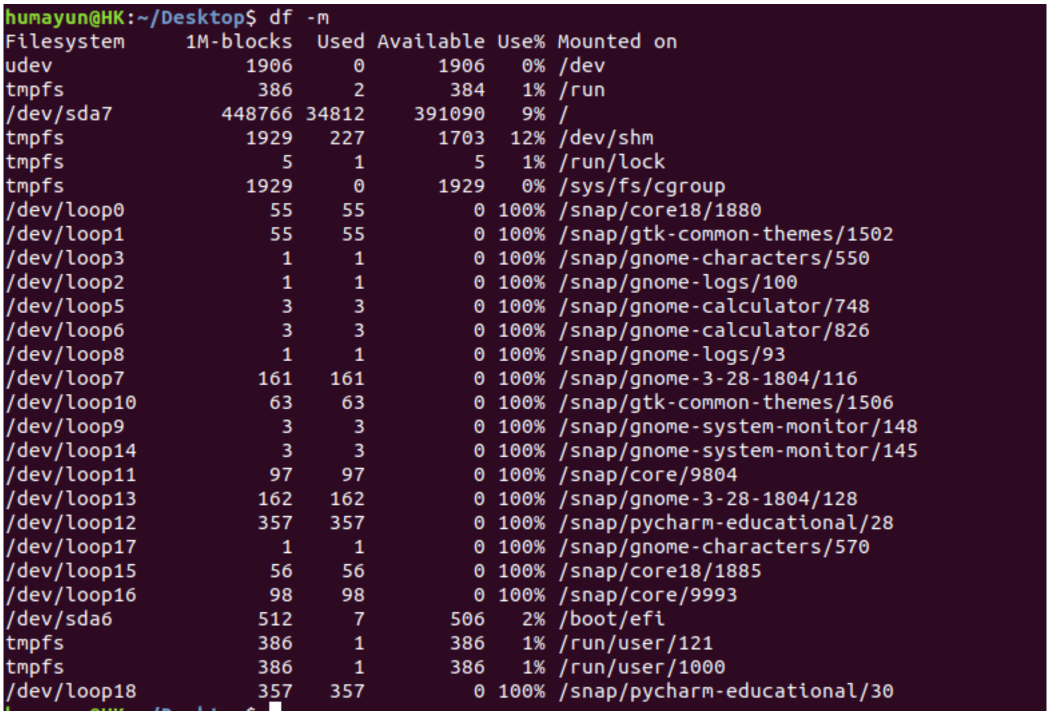
```
humayun@HK:~/Desktop$ ls
folder  Signs-In-Holy-Quran.png
humayun@HK:~/Desktop$ touch something
humayun@HK:~/Desktop$ ls
folder  Signs-In-Holy-Quran.png  something
```

7. man & –help — To know more about a command and how to use it, we use the **man** command. It shows the manual pages of the command. For example, "**man cd**" shows the manual pages of the **cd** command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., **cd -help**).



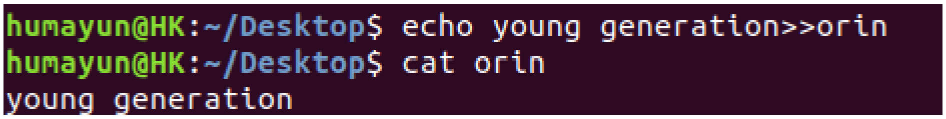
```
humayun@HK:~$ man
What manual page do you want?
humayun@HK:~$ cd --help
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option `cdable_vars' is set,
    the word is assumed to be  a variable name.  If that variable has a value,
    its value is used for DIR.

    Options:
      -L        force symbolic links to be followed: resolve symbolic
                links in DIR after processing instances of `..'
      -P        use the physical directory structure without following
                symbolic links: resolve symbolic links in DIR before
                processing instances of `..'
      -e        if the -P option is supplied, and the current working
                directory cannot be determined successfully, exit with
                a non-zero status
      -@        on systems that support it, present a file with extended
                attributes as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
    back to a slash or the beginning of DIR.

    Exit Status:
    Returns 0 if the directory is changed, and if $PWD is set successfully when
    -P is used; non-zero otherwise.
```

8. df — We use the **df** command to see the available disk space in each of the partitions in our system. We can just type in **df** in the command line and you can see each mounted partition and their used/available space in % and in KBs. If you want it shown in megabytes, you can use the command "**df -m**".



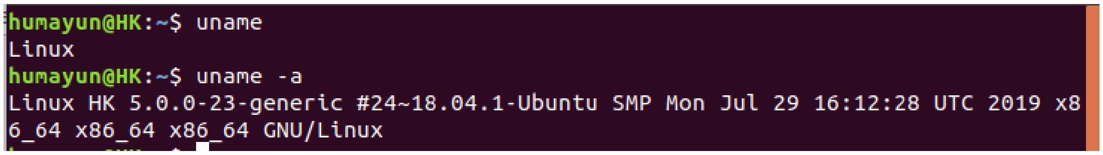
```
humayun@HK:~/Desktop$ df -m
Filesystem     1M-blocks  Used Available Use% Mounted on
udev                1906     0      1906   0% /dev
tmpfs                386     2       384   1% /run
/dev/sda7         448766 34812    391090   9% /
tmpfs               1929   227      1703  12% /dev/shm
tmpfs                  5     1         5   1% /run/lock
tmpfs               1929     0      1929   0% /sys/fs/cgroup
/dev/loop0            55    55         0 100% /snap/core18/1880
/dev/loop1            55    55         0 100% /snap/gtk-common-themes/1502
/dev/loop3             1     1         0 100% /snap/gnome-characters/550
/dev/loop2             1     1         0 100% /snap/gnome-logs/100
/dev/loop5             3     3         0 100% /snap/gnome-calculator/748
/dev/loop6             3     3         0 100% /snap/gnome-calculator/826
/dev/loop8             1     1         0 100% /snap/gnome-logs/93
/dev/loop7           161   161         0 100% /snap/gnome-3-28-1804/116
/dev/loop10           63    63         0 100% /snap/gtk-common-themes/1506
/dev/loop9             3     3         0 100% /snap/gnome-system-monitor/148
/dev/loop14            3     3         0 100% /snap/gnome-system-monitor/145
/dev/loop11           97    97         0 100% /snap/core/9804
/dev/loop13          162   162         0 100% /snap/gnome-3-28-1804/128
/dev/loop12          357   357         0 100% /snap/pycharm-educational/28
/dev/loop17            1     1         0 100% /snap/gnome-characters/570
/dev/loop15           56    56         0 100% /snap/core18/1885
/dev/loop16           98    98         0 100% /snap/core/9993
/dev/sda6            512     7       506   2% /boot/efi
tmpfs                386     1       386   1% /run/user/121
tmpfs                386     1       386   1% /run/user/1000
/dev/loop18          357   357         0 100% /snap/pycharm-educational/30
```

9. echo & **cat** — The "**echo**" command helps us move some data, usually text into a file. For example, if we want to create a new text file or add to an already made text file, we just need to type in, "**echo hello, my name is AfrinZaman >> orin**". We do not need to separate the spaces by using the backward slash here, because we put in two triangular brackets when we finish what we need to write. We use the **cat** command to display the contents of a file. It is usually used to easily view programs.



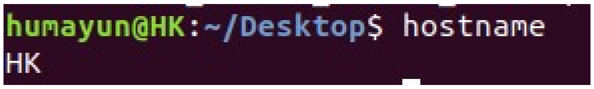
```
humayun@HK:~/Desktop$ echo young generation>>orin
humayun@HK:~/Desktop$ cat orin
young generation
```

10. uname — We use **uname** to show the information about the system our Linux distro is running. Using the command "**uname -a**" prints most of the information about the system. This prints the kernel release date, version, processor type, etc.



```
humayun@HK:~$ uname
Linux
humayun@HK:~$ uname -a
Linux HK 5.0.0-23-generic #24~18.04.1-Ubuntu SMP Mon Jul 29 16:12:28 UTC 2019 x8
6_64 x86_64 x86_64 GNU/Linux
```

11. **hostname** — We use **hostname** to know our name in our host or network. Basically, it displays our hostname and IP address. Just typing "**hostname**" gives the output. Typing in "**hostname -I**" gives us our IP address in our network.



```
humayun@HK:~/Desktop$ hostname
HK
```

Discussion :

This lab helps us to learn about linux command and to know more about Linux Operating System. We have used 15 linux commands in this lab and this lab have helped us a lot. In future we can use Linux Operating System.