BASIC LINUX COMMANDS 2

1.echo: echo command is used to move some data into a file. If you want to add the text, "Hello, my name is John" into a file called name.txt, you would type echo Hello, my name is John >> name.txt 2. head .

Eg:

```
user@user-HP-Laptop-15-da0xxx:~$ pwd
/home/user
user@user-HP-Laptop-15-da0xxx:~$ cd silja
user@user-HP-Laptop-15-da0xxx:~/silja$ echo hello,my name is silja>>tst
user@user-HP-Laptop-15-da0xxx:~/silja$ cat tst
hello,my name is silja
```

2.head: The head command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. If you only want to show the first five lines, type head -n 5 filename.txt.

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ head -n 2 lin
helooo
hiiii
user@user-HP-Laptop-15-da0xxx:~/silja$
```

3. tail : This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. tail -n filename.txt.

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ tail -n 2 lin
yeloo
heyyyy
user@user-HP-Laptop-15-da0xxx:~/silja$
```

4. read : read the contents of a line into a variable. The read command can be used with and without arguments .read command is used to read [options] [name...] . $read • $read var1 var2 var3 .$echo "[$var1] [$var2] [$var3].

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ $read
user@user-HP-Laptop-15-da0xxx:~/silja$ $read a
a: command not found
user@user-HP-Laptop-15-da0xxx:~/silja$ read a

user@user-HP-Laptop-15-da0xxx:~/silja$
user@user-HP-Laptop-15-da0xxx:~/silja$ echo "a"
a
user@user-HP-Laptop-15-da0xxx:~/silja$
```

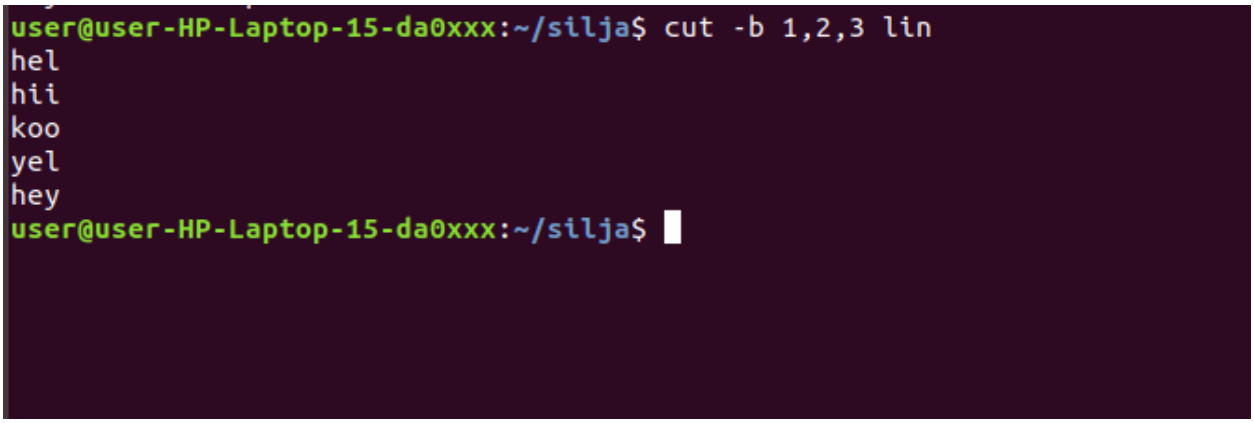
5. more :Like cat command, more command displays the content of a file. Only difference is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time. Enter key:

Eg:

6. less : The 'less' command is same as 'more' command but include some more features. It automatically adjust with the width and height of the teminal window, while 'more' command cuts the content as the width of the terminal window get shorter.

Eg:

7. cut : The cut command is used for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and fiel. Eg:



8. paste : It is used to join files horizontally (parallel merging) by

outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. paste [OPTION]... [FILES]... $ paste state.txt capital.txt .

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ paste tst lin
hello,my name is silja  helooo
        hiiii
        koooooii
        yeloo
        heyyyy
user@user-HP-Laptop-15-da0xxx:~/silja$
```

9. uname :The uname command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on. $uname ,$uname –r.

Eg:

```
user@user-HP-Laptop-15-da0xxx:~$ pwd
/home/user
user@user-HP-Laptop-15-da0xxx:~$ cd silja
user@user-HP-Laptop-15-da0xxx:~/silja$ uname -r
4.18.0-15-generic
user@user-HP-Laptop-15-da0xxx:~/silja$
```

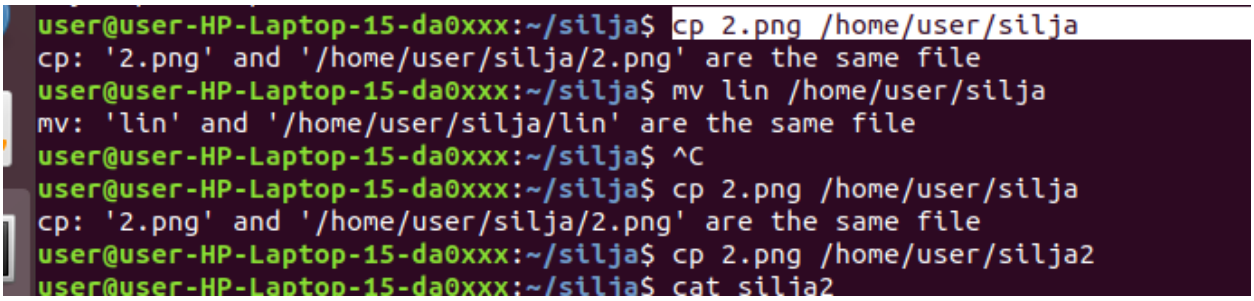
10. cp : cp command is used to copy files from the current directory to a different directory. For instance, the command cp scenery.jpg /home/username/Pictures would create a copy of scenery.jpg (from your current directory) into the Pictures directory. cp -i will ask for user's consent in case of a potential file overwrite. cp -p will preserve source files'mode, ownership and timestamp. cp -r will copy directories recursively. cp -u copies files only if the destination file is not existing or the source file is newer than the destination file.

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ cp 2.png /home/user/silja
cp: '2.png' and '/home/user/silja/2.png' are the same file
user@user-HP-Laptop-15-da0xxx:~/silja$ mv lin /home/user/silja
mv: 'lin' and '/home/user/silja/lin' are the same file
user@user-HP-Laptop-15-da0xxx:~/silja$ ^C
user@user-HP-Laptop-15-da0xxx:~/silja$ cp 2.png /home/user/silja
cp: '2.png' and '/home/user/silja/2.png' are the same file
user@user-HP-Laptop-15-da0xxx:~/silja$ cp 2.png /home/user/silja2
user@user-HP-Laptop-15-da0xxx:~/silja$ cat silja2
```

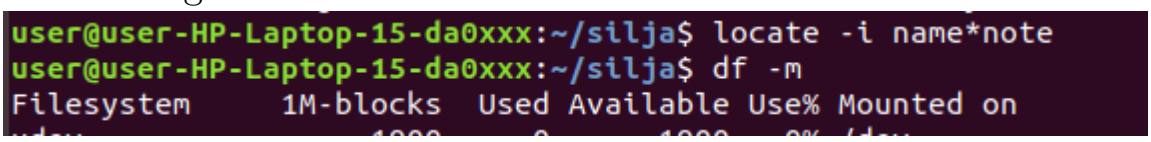
11. **mv :** The primary use of the mv command is to move files, it can also be used to rename files. The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory. mv file.txt /home/username/Documents .To rename files, the Linux is mv oldname.ext newname.ext.

Eg:



12. **locate :**To locate a file, just like the search command in Windows. What's more, using the -i argument along with this command will make it caseinsensitive, so you can search for a file even if you don't remember its exact name. To search for a file that contains two or more words, use an asterisk (*). For example, locate -i school*note command will search for any file that contains the word "school" and "note", whether it is uppercase or lowercase.

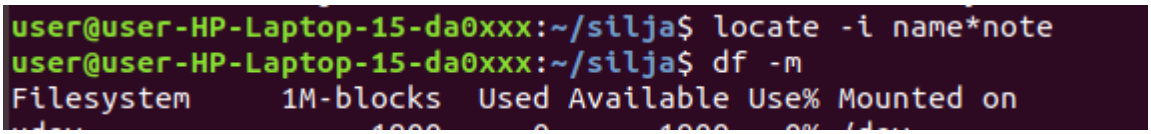
Eg:



13. **find** :Similar to the locate command, using find also searches for files

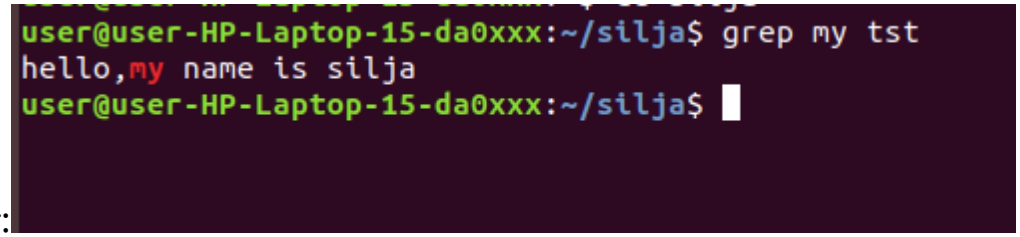
and directories. The difference is, you use the find command to locate files within a given directory. As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its subdirectories. Other variations when using the find are: To find files in the current directory use, find . -name notes.txt .To look for directories use, / -type d -name notes. Txt.

Eg:



14. grep : Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file. To illustrate, grep blue notepad.txt will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. Usually output of a previous command is piped into the grep command. For example ls -l | grep "kernel" .

Eg:



15. df :Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.

Eg:

```
user@user-HP-Laptop-15-da0xxx:~/silja$ df -m
Filesystem     1M-blocks  Used Available Use% Mounted on
udev                1900     0      1900   0% /dev
tmpfs                387     2       385   1% /run
/dev/sda8          48961  6548     39897  15% /
tmpfs                1931     0      1931   0% /dev/shm
tmpfs                   5     1         5   1% /run/lock
tmpfs                1931     0      1931   0% /sys/fs/cgroup
/dev/loop3             15    15         0 100% /snap/gnome-logs/45
/dev/loop2            141   141         0 100% /snap/gnome-3-26-1604/74
/dev/loop4              3     3         0 100% /snap/gnome-calculator/260
/dev/loop0             91    91         0 100% /snap/core/6350
/dev/loop5             13    13         0 100% /snap/gnome-characters/139
/dev/loop6              4     4         0 100% /snap/gnome-system-monitor/57
/dev/loop1             35    35         0 100% /snap/gtk-common-themes/818
/dev/sda2              96    31        66  32% /boot/efi
tmpfs                387     1       386   1% /run/user/1000
user@user-HP-Laptop-15-da0xxx:~/silja$
```

16. **du** :If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.

Eg:

17. **useradd :** This is available only to system admins .Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. useradd is used to create a new user, while passwd is adding a password to that user's account. To add a new person named John type, useradd John and then to add his password type, passwd 123456789 .

18. **userdel** :Remove a user is very similar to adding a new user. To delete the users account type, userdel UserName .

Eg:

19. **sudo** :Short for "SuperUser Do", this command enables you to perform tasks that require administrative or root permissions. You must

have sufficient permissions to use this command. sudo useradd maria.

Eg:

20. passwd : Changes passwords for user accounts. A normal user may only change the password for their own account, while the superuser may change the password for any account.

Eg: