NETWORKING & SYSTEM ADMINISTRATION LAB

Name: Vismaya Mohan

Roll No: 54

Batch: B

Date: 28/3/2022

Experiment No.: 3

Aim

Familiarization of the linux commands.

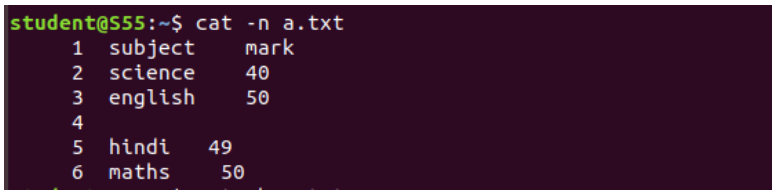
Procedure

17. cat -n

This command is used to specify the contents with line number.

Syntax :- $ cat -n filename

Output :



18. cat -b

This command is used to remove the empty lines from the file.

Syntax :- $ cat -b filename

Output :

19. touch

This command is used to create an empty file.

Syntax :- $ touch filename

Output :

```
student@S55:~$ touch b.txt
```

20. echo

This command is used to add contents to the file.

Syntax :- $ echo content >> filename

Output :- Contents will be added to the file.

```
student@S55:~$ touch b.txt
student@S55:~$ echo subject mark with different student >> b.txt
```

21. head

This command is used to display the first 10 lines of the file by default.

Syntax :- $ head filename

Output:

```
student@S55:~$ head a.txt
subject    mark
science    40
english    50

hindi   49
maths    50
```

```
vismi
vysh
devika
anu
```

22. head -4 filename

This command is used to display the lines of the file to the specified number from head.

Syntax :- $ head -n filename

Output :

```
student@S55:~$ head -4 a.txt
subject    mark
science    40
english    50
```

23. tail

This command is used to display the last 10 lines of the file by default.

Syntax :- $ tail filename

Output :

```
student@S55:~$ tail a.txt
maths    50
vismi
vysh
devika
anu
sree
lachu
appu
achu
ammu
```

24. tail -3 a.txt

This command is used to display the lines of the file to the specified number from tail.

Syntax :- $ tail -n filename

Output :-

```
student@S55:~$ tail -3 a.txt
appu
achu
ammu
```

25. cut -d- -f2

This command is used to cut and the display the contents based on the delimiter given.

Syntax :- $ cut -d- -f2 filename

Output :

```
student@S55:~$ cut -d- -f2 a.txt
subject    mark
science    40
english    50

hindi   49
maths    50
vismi
vysh
devika
anu
sree
lachu
appu
achu
ammu
```

26. rmdir :

Used to delete a directory if it is empty.

Syntax : $ rmdir [directory name]

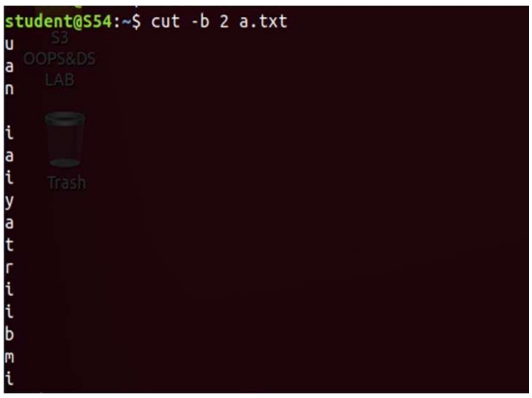
Output :

```
student@S54:~$ rmdir network
```

27.cut -b 2 [filename] : It can be used to cut parts of a line by byte position(n).

syntax : cut -b number filename

output:

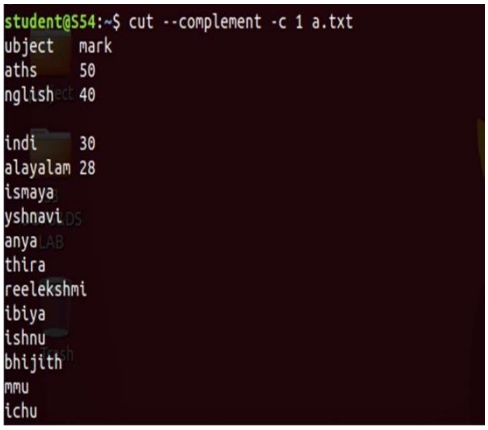


28.cut --complement -c 1 [filename] :

This option instructs cut to display all thefields, bytes or characters except the selected.

Syntax : $ cut –complement -c num [filename]

Output :



29.paste marvel1 marvel2 :

To paste the content in one file to another file.

Syntax : $ paste file1 file2

Output:

```
student@S54:~$ cat > e.txt
students name
abcd
akhil
aravind
abhirami
arun
^Z
[3]+  Stopped                 cat > e.txt
student@S54:~$ cat e.txt
students name
abcd
akhil
aravind
abhirami
arun
```

```
student@S54:~$ paste a.txt e.txt
subject   mark  students name
maths     50    abcd
english   40    akhil
        aravind
hindi     30    abhirami
malayalam 28    arun
vismaya
vyshnavi
manya
athira
sreelekshmi
libiya
vishnu
abhijith
ammu
kichu
```

30.paste marvel1 marvel2 > marvel3 :

To copy the contents of two given files to athird file.

Syntax : $ paste file1 file2 > file

Output:

```
student@S54:~$ paste a.txt e.txt > f.txt
student@S54:~$ cat f.txt
subject   mark  students name
maths     50    abcd
english   40    akhil
        aravind
hindi     30    abhirami
malayalam 28    arun
vismaya
vyshnavi
manya
athira
sreelekshmi
libiya
vishnu
abhijith
ammu
kichu
```

31. **paste -d ‘ - ’ marvel1 marvel :**

The -d, -delimiters option allows you to specify a

list of characters to be used as delimiters instead of the default TAB saparator

Syntax : $ paste -d ‘ - ’ file1 file2

Output:

32. paste -d ‘%|’ marvel1 marvel2 marvel1 :

Two delimiters are used. The lines from the first and the second file are separated with the first character from the delimiters list. The second and the third file lines are separated with the second delimiter.

Syntsx : $ paste -d ‘%|’ file2 file 1

Output:

33. **paste -s marvel1 marvel2:**

This command will merge all lines from the given filein separated lines.

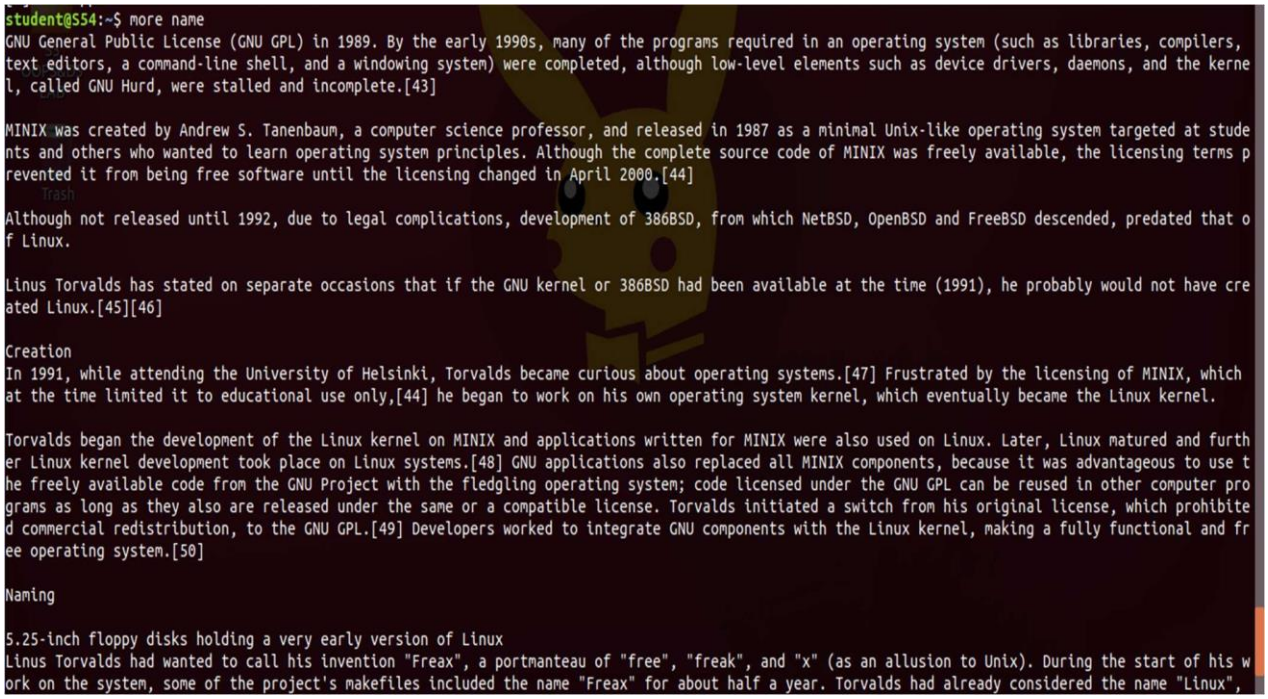
Syntax : $ paste -s file 1 file2

Output:

34. **more name :**

The more command displays the first section of the file. By pressingthe “ENTER” key, we can scroll line by line, all the way to the bottom of the file.

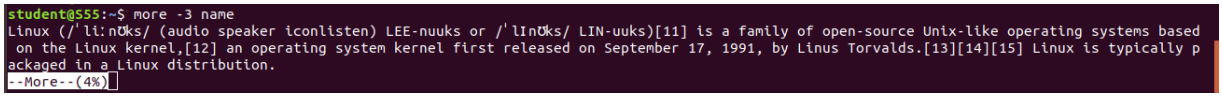
Syntax :$ more filename



35. **more -3 name :**

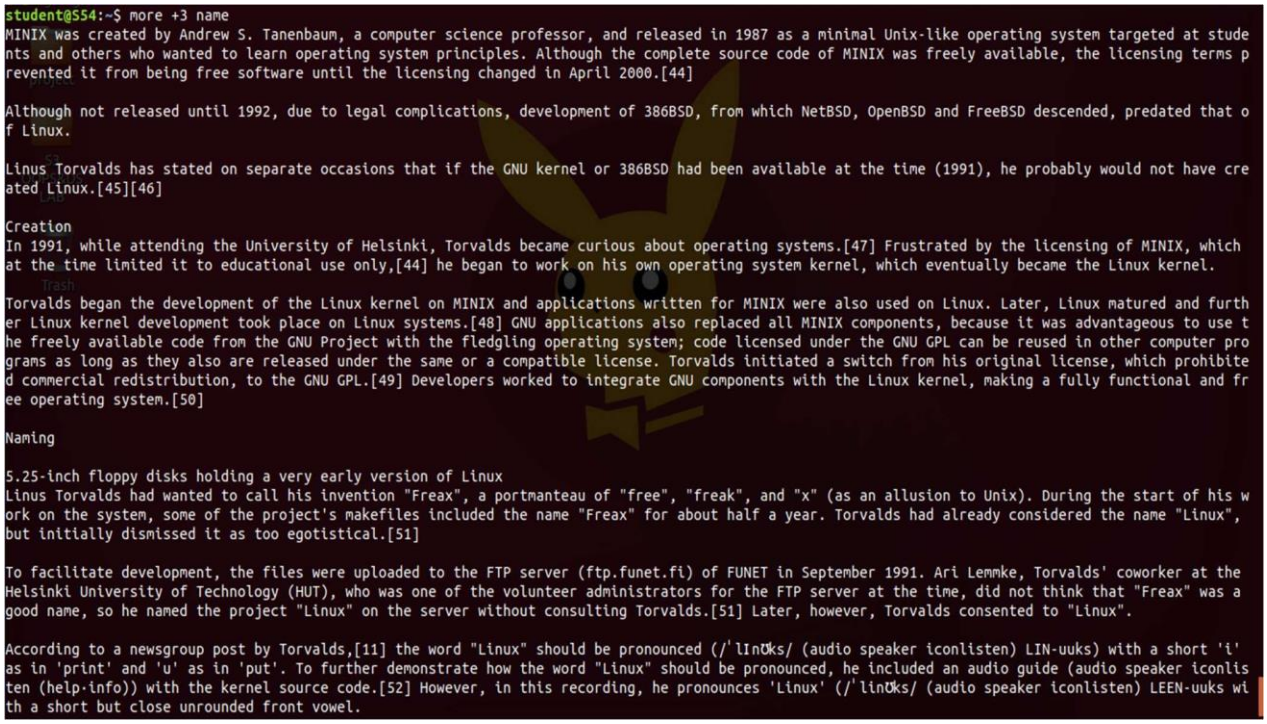
Used to type the number of lines(num) to display per screen.

Syntax : $more -num filename



36. more +3 name :

This option displays the text after the specified number of linesof the document.
Syntax : $ more + num filename

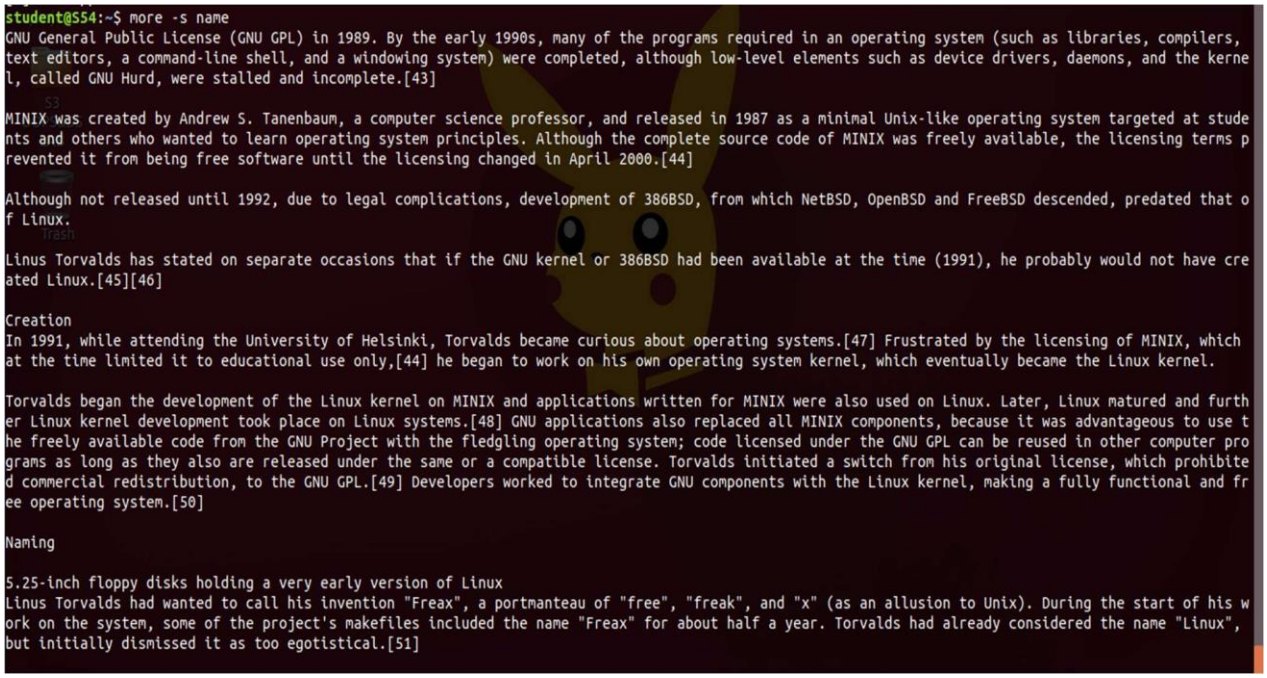


37. more -s name :

This option squeezes multiple blank lines into one single blankline.

Syntax : $ more -s filename

Output:



```
student@S54:~$ more -s name
GNU General Public License (GNU GPL) in 1989. By the early 1990s, many of the programs required in an operating system (such as libraries, compilers,
text editors, a command-line shell, and a windowing system) were completed, although low-level elements such as device drivers, daemons, and the kerne
l, called GNU Hurd, were stalled and incomplete.[43]

MINIX was created by Andrew S. Tanenbaum, a computer science professor, and released in 1987 as a minimal Unix-like operating system targeted at stude
nts and others who wanted to learn operating system principles. Although the complete source code of MINIX was freely available, the licensing terms p
revented it from being free software until the licensing changed in April 2000.[44]

Although not released until 1992, due to legal complications, development of 386BSD, from which NetBSD, OpenBSD and FreeBSD descended, predated that o
f Linux.

Linus Torvalds has stated on separate occasions that if the GNU kernel or 386BSD had been available at the time (1991), he probably would not have cre
ated Linux.[45][46]

Creation
In 1991, while attending the University of Helsinki, Torvalds became curious about operating systems.[47] Frustrated by the licensing of MINIX, which
at the time limited it to educational use only,[44] he began to work on his own operating system kernel, which eventually became the Linux kernel.

Torvalds began the development of the Linux kernel on MINIX and applications written for MINIX were also used on Linux. Later, Linux matured and furth
er Linux kernel development took place on Linux systems.[48] GNU applications also replaced all MINIX components, because it was advantageous to use t
he freely available code from the GNU Project with the fledgling operating system; code licensed under the GNU GPL can be reused in other computer pro
grams as long as they also are released under the same or a compatible license. Torvalds initiated a switch from his original license, which prohibite
d commercial redistribution, to the GNU GPL.[49] Developers worked to integrate GNU components with the Linux kernel, making a fully functional and fr
ee operating system.[50]

Naming

5.25-inch floppy disks holding a very early version of Linux
Linus Torvalds had wanted to call his invention "Freax", a portmanteau of "free", "freak", and "x" (as an allusion to Unix). During the start of his w
ork on the system, some of the project's makefiles included the name "Freax" for about half a year. Torvalds had already considered the name "Linux",
but initially dismissed it as too egotistical.[51]
```