SIMPLE FILTERS

pr:Paginating Files

- pr command prepares a file for printing by adding suitable headers, footers and formatted text

2016-08-18 12:33   d2.txt   Page 1

saturday

sunday

tuesday

- Header shows the date and time of last modification of the file along with the filename and page number

pr options

- **-d** Double spaces input
- **-n** Number lines
- **-o n** offsets lines by n spaces, increases left margin of page

head command

- This command displays the top the file
- When used this command without option, displays the first ten lines of the file
- **$ head filename** //default displays first ten lines

head command with option

- **$ head -n 3 emp.lst** //displays first three lines of emp.lst

The tail command in unix or linux system

- It is used to print the last N lines from the file on the terminal

- The syntax of tail command is

  tail [options] [files]

The tail command options are:

c : Prints the last N bytes of file; With leading +, prints the characters from the N byte in the file.

n : Prints last N lines; With leading + prints lines from the Nth line in the file.

f : Prints the appended lines on the terminal as the file grows.

tail Command Examples

- **cat example.txt**

virtual storage

oracle virtual instance

mysql backup

dedicated hosting server

cloud servers

By default, the tail command prints the last 10 lines from the file.

- **tail example.txt**

Display last N lines

- **tail -n2 example.txt**

dedicated hosting server

cloud servers

Print lines from the Nth line

- **tail -n+2 example.txt**

oracle virtual instance

mysql backup

dedicated hosting server

cloud servers

Print the last n bytes

the -c option to print the last N bytes from the file

- **tail -c8 example.txt**

servers

Cut Command in Unix (Linux) Examples

- Cut command in unix (or linux) is used to select sections of text from each line of files

- **cat file.txt**

unix or linux os

is **u**nix good os

is **l**inux good os

- **To print the characters in a line, use the -c option in cut command**

  cut -c4 file.txt

x

u

l

- **cut -c4,6 file.txt**

Output:?

- **cut -c4-7 file.txt**

x or

unix

linu

- **cut -c-6 file.txt**

unix o

is uni

is lin

- cut -c- file.txt

Output:?

cut command to print the fields using the delimiter

- **cut -d ' ' -f2 file.txt**

or

unix

linux

- **The -d option in cut command can be used to specify the delimiter and -f option is used to specify the field position.**

- **cut -d' ' -f2,3 file.txt**

A unix/linux cut command to display range of fields?

- **cut -d' ' -f1-3 file.txt**
- **cut -d' ' -f-3 file.txt**
- **cut -d' ' -f2- file.txt**

paste command

- The paste command merges the lines from multiple files.

- **$paste file1 file2**

- **cat addressbook**

anup kumar
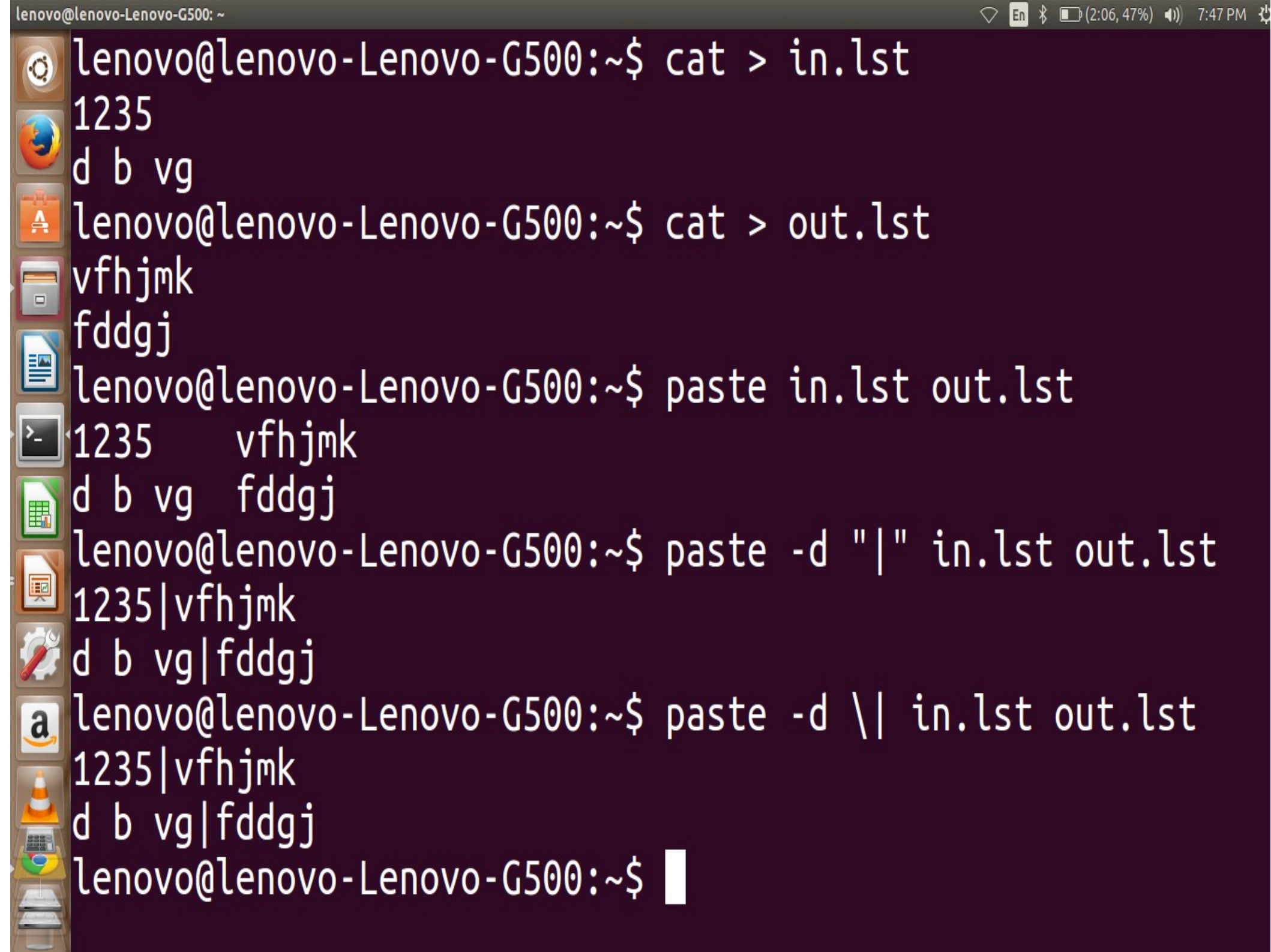
anup_k@yahoo.com

24569083

vinod sharma

vinod_sharma@hotmail.com

34586532

```
lenovo@lenovo-Lenovo-G500:~$ cat > in.lst
1235
d b vg
lenovo@lenovo-Lenovo-G500:~$ cat > out.lst
vfhjmk
fddgj
lenovo@lenovo-Lenovo-G500:~$ paste in.lst out.lst
1235     vfhjmk
d b vg   fddgj
lenovo@lenovo-Lenovo-G500:~$ paste -d "|" in.lst out.lst
1235|vfhjmk
d b vg|fddgj
lenovo@lenovo-Lenovo-G500:~$ paste -d \| in.lst out.lst
1235|vfhjmk
d b vg|fddgj
lenovo@lenovo-Lenovo-G500:~$
```
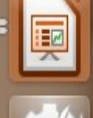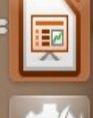
```
lenovo@lenovo-Lenovo-G500:~$ paste -s -d "||\n" a
ddressbook
anup kumar|anup_k@yahoo.com|24569083
vinod sharma|vinod_sharma@hotmail.com|34586532
lenovo@lenovo-Lenovo-G500:~$
```

```
lenovo@lenovo-Lenovo-G500:~$ paste -s -d "||" add
ressbook
anup kumar|anup_k@yahoo.com|24569083|vinod sharma
|vinod_sharma@hotmail.com|34586532
lenovo@lenovo-Lenovo-G500:~$
```

```
lenovo@lenovo-Lenovo-G500:~$ paste -s -d "|\n" ad
dressbook
anup kumar|anup_k@yahoo.com
24569083|vinod sharma
vinod_sharma@hotmail.com|34586532
lenovo@lenovo-Lenovo-G500:~$
```

tr command

- tr is an UNIX utility for translating, or deleting, or squeezing repeated characters. It will read from STDIN and write to STDOUT.

- tr stands for translate.

Translation

- The syntax of tr command is:

  $ tr [OPTION] SET1 [SET2]

- If both the SET1 and SET2 are specified and '-d' OPTION is not specified, then tr command will replace each characters in SET1 with each character in same position in SET2.

- **$ tr a-z A-Z or tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ**

- $ tr '{}' '()' < inputfile > outputfile

  The above command will read each character from "inputfile", translate if it is a brace, and write the output in "outputfile".

Translate white-space to tabs

- **$ echo "This is for testing" | tr [:space:] '\t'**

Delete specified characters using -d option

- **$ echo "the geek stuff" | tr -d 't'**
- **$ echo "my username is 432234" | tr -d [:digit:]**