Filter commands:

1. Cut Command:

The syntax for the cut command is as follows:
cut OPTION... [FILE]...
In the following example a text file exists and is saved as abcd.txt.

```
roehit@LAPTOP-0SIPK43K:~$ cat > abcd.txt
john,smith,34,london
arthur,evans,21,newport
george,jones,32,truro
^Z
[1]+  Stopped                 cat > abcd.txt
```

The delimiter can be set to a comma with -d ','. cut can then pull out the fields of interest with the -f flag. In the following example the first field is cut.

```
roehit@LAPTOP-0SIPK43K:~$ cut -d ',' -f 1 abcd.txt
john
arthur
george
```

Multiple fields can be cut by passing a comma separated list.

```
roehit@LAPTOP-0SIPK43K:~$ cut -d ',' -f 1,4 abcd.txt
john,london
arthur,newport
george,truro
```

Create another file:

```
roehit@LAPTOP-0SIPK43K:~$ cat > state.txt
Tamilnadu
Assam
Bihar
Punjab
Sikkim
```

-b(byte): To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma

```
roehit@LAPTOP-0SIPK43K:~$ cut -b 1,2,3 state.txt
Tam
Ass
Bih
Pun
Sik
```

In this, 1- indicate from 1st byte to end byte of a line

```
roehit@LAPTOP-0SIPK43K:~$ cut -b 1- state.txt
Tamilnadu
Assam
Bihar
Punjab
Sikkim
```

In this, -3 indicate from 1st byte to 3rd byte of a line

```
roehit@LAPTOP-0SIPK43K:~$ cut -b -3 state.txt
Tam
Ass
Bih
Pun
Sik
```

-c (column): To cut by character use the -c option
Tabs and backspaces are treated as a character
$ cut -c 2,5,7 state.txt

```
roehit@LAPTOP-0SIPK43K:~$ cut -c 2,5,7 state.txt
ala
sm
ir
ua
ii
```

$ cut -c 1-7 state.txt

```
roehit@LAPTOP-0SIPK43K:~$ cut -c 1-7 state.txt
Tamilna
Assam
Bihar
Punjab
Sikkim
```

LINUX tr commands:

1. A simple tr command use case is to change all lower-case letters in text to upper case and vice versa, as shown below

```
roehit@LAPTOP-0SIPK43K:~$ cat >linux.txt
linux sucks in everyday use
not for average users
its open source
roehit@LAPTOP-0SIPK43K:~$ cat linux.txt | tr [:lower:] [:upper:]
LINUX SUCKS IN EVERYDAY USE
NOT FOR AVERAGE USERS
ITS OPEN SOURCE
```

Alternatively, you can use the following command to change all lower-case letters to upper case in a file as shown.

```
roehit@LAPTOP-0SIPK43K:~$ cat linux.txt | tr [a-z] [A-Z]
LINUX SUCKS IN EVERYDAY USE
NOT FOR AVERAGE USERS
ITS OPEN SOURCE
roehit@LAPTOP-0SIPK43K:~$
```

Another useful feature is, you can use the -d flag to delete characters, for example to remove the spaces in the domain names using the following command.

```
^[roehit@LAPTOP-0SIPK43K:~$ cat > domains.txt
blah blah
space space
blah blah
roehit@LAPTOP-0SIPK43K:~$ cat domains.txt | tr -d ' '
blahblah
spacespace
blahblah
```

diff commands:

```
roehit@LAPTOP-0SIPK43K:~$ cat > a.txt
Goa
lagos
paris
roehit@LAPTOP-0SIPK43K:~$ cat > b.txt
why
what
how
roehit@LAPTOP-0SIPK43K:~$ cat > c.txt
ear
eye
tongue
roehit@LAPTOP-0SIPK43K:~$ diff a.txt b.txt
1,3c1,3
< Goa
< lagos
< paris
---
> why
> what
> how
roehit@LAPTOP-0SIPK43K:~$
```

Cmp command:

Compare two files, and if they differ, tells the first byte and line number where they differ.

```
roehit@LAPTOP-0SIPK43K:~$ cat > samp_1.txt
this is sample no 1
roehit@LAPTOP-0SIPK43K:~$ cat > samp_2.txt
this is sample no 2
roehit@LAPTOP-0SIPK43K:~$ cmp samp_1.txt samp_2.txt
samp_1.txt samp_2.txt differ: byte 19, line 1
roehit@LAPTOP-0SIPK43K:~$
```

Grep command:

Syntax:
grep [options] [pattern] [file]
The pattern is specified as a regular expression. A regular expression is a string of characters that is used to specify a pattern matching rule. Special characters are used to define the matching rules and positions.
Examples:
(A document is created with the name "assn")

```
roehit@LAPTOP-0SIPK43K:~$ cat > assn.txt
hello
bi
c u
```

Match all lines that start with ‘hello’. E.g: “hello hoomans”
$ grep “^hello” file1

```
roehit@LAPTOP-0SIPK43K:~$ grep "^hello" assn.txt
hello
```

Unix sort command:

Sort syntax:

sort [options] [files]

Sort Options:

Some of the options supported are:

- sort -r: Reverse the sorting order.

sort -n: Use the numerical value to sort.

```
roehit@LAPTOP-0SIPK43K:~$ cat list.txt
01 allan
02 bob
03 tom
```

Sort with default ordering:
$ sort file1.txt

```
roehit@LAPTOP-0SIPK43K:~$ sort list.txt

01 allan
02 bob
03 tom
```

In this example, the sorting is first performed using the first character. Since this is the same for all lines, the sorting then proceeds to the second character. Since the second character is unique for each line, the sorting ends there.

Sort in reverse ordering:
$ sort -r file1.txt

```
roehit@LAPTOP-0SIPK43K:~$ sort -r list.txt
03 tom
02 bob
01 allan
```

sort by the second field:
sort -k POS1, POS2: Specify a key to do the sorting. POS1 and POS2 are optional parameters and are used to indicate the starting field and the ending field indices. Without POS2, only the field

```
roehit@LAPTOP-0SIPK43K:~$ sort -k 2 list.txt

01 allan
02 bob
03 tom
```

Now assume the original list.txt is as below

```
roehit@LAPTOP-0SIPK43K:~$ cat list.txt
01 allan
02 bob
03 tom
04 roger
05 steve
06 eric
```

Sort with default ordering

$ sort file2.txt

Sort suppressing repeated lines

$ sort -u file2.txt

```
roehit@LAPTOP-0SIPK43K:~$ sort -u list.txt
01 allan
02 bob
03 tom
04 roger
05 eric
06 eric
```

The "comm"; command compares two files or streams. By default, 'comm' will always display three columns. First column indicates non-matching items of first file, second column indicates non- matching items of second file, and third column indicates matching items of both the files. Both the files has to be in sorted order for 'comm' ; command to be executed.

Syntax:

1. comm <file1><file2>

```
roehit@LAPTOP-0SIPK43K:~$
roehit@LAPTOP-0SIPK43K:~$ comm list.txt list2.txt
01 allan
        01 bob
02 bob
        02 charles
        03 alan
03 tom
        04 david
04 roger
05 eric
06 eric
roehit@LAPTOP-0SIPK43K:~$
```

Paste Command:

```
roehit@LAPTOP-0SIPK43K:~$ cat > state.txt
tamilnadu
goa
karnataka
roehit@LAPTOP-0SIPK43K:~$ cat > capital.txt
chennai
panaji
bangalore
roehit@LAPTOP-0SIPK43K:~$ cat > num.txt
1
2
3
```

1. –d delimiter

```
roehit@LAPTOP-0SIPK43K:~$ paste -d "|" num.txt state.txt capital.txt
1|tamilnadu|chennai
2|goa|panaji
3|karnataka|bangalore
```

More than one character is specified

$ paste -d "|," number state capital

```
roehit@LAPTOP-0SIPK43K:~$ paste -d "|," num.txt state.txt capital.txt
1|tamilnadu,chennai
2|goa,panaji
3|karnataka,bangalore
```

2. -s (serial):

$ paste -s number state capital

```
roehit@LAPTOP-0SIPK43K:~$ paste -s num.txt state.txt capital.txt
1       2       3
tamilnadu       goa     karnataka
chennai panaji  bangalore
```

$ paste -s -d ":" number state capital

```
roehit@LAPTOP-0SIPK43K:~$ paste -s -d ":" num.txt state.txt capital.txt
1:2:3
tamilnadu:goa:karnataka
chennai:panaji:bangalore
```

Applications of paste command:

1. Combining N consecutive lines: The paste command can also be used to merge N consecutive lines from a file into a single line. Here N can be specified by specifying number hyphens(-) after paste.

with 2 hyphens

```
roehit@LAPTOP-0SIPK43K:~$ cat > capital | paste - -
ItanagarDispur
Hydrebad  Patna
Raipur
```

with 3 hyphens

```
roehit@LAPTOP-0SIPK43K:~$ paste - - - < capital
ItanagarDispur  Hydrebad  Patna Raipur
roehit@LAPTOP-0SIPK43K:~$
```

2. Combination with other commands: Even though paste require at least two files for concatenating lines, but data from one file can be given from shell. Like in our example below, cut command is used with -f option for cutting out first field of state file and output is pipelined with paste command having one file name and instead of second file name hyphen is specified.

Note: If hyphen is not specified then input from shell is not pasted.

without hyphen:

```
roehit@LAPTOP-0SIPK43K:~$ cut -d " " -f 1 state.txt | paste num.txt
1
2
3
```

with hyphen:

```
roehit@LAPTOP-0SIPK43K:~$ cut -d " " -f 1 state.txt | paste num.txt -
1       tamilnadu
2       goa
3       karnataka
```

Ordering of pasting can be changed by altering the location of hyphen:

```
roehit@LAPTOP-0SIPK43K:~$ cut -d " " -f 1 state.txt | paste - num.txt
tamilnadu       1
goa     2
karnataka       3
```

Uniq command:

```
roehit@LAPTOP-0SIPK43K:~$ cat > example.txt
hi
hi
bi
ci
di

roehit@LAPTOP-0SIPK43K:~$ uniq example.txt
hi
bi
ci
di
```