

Sessions - 20

* I/O Redirection -

→ Cond

↳ Output

(Pipeline)

Cond (Input)

* \rightarrow pipeline symbol

(Output)

date | to 'a-z' 'A-Z' (Input)

date | to 'a-z' 'A-Z'

* # rpm -qa (Show all Software installed)

* put something in f.txt (File)

* # wc f.txt (Count data → Word, bytes)

(Again we don't know how to read; actual command)

wc < f.txt

wc (Take input like \$ point file (cat))

↓ (Ctrl + d) → End of file sign

while taking input control like (wc, cat) fly
on file as they use Enter & Ctrl+d
Like Control end of file → End of file

--- ↳ Enter → 10 → 802

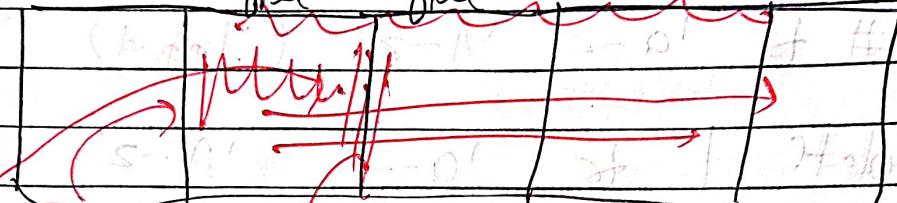
--- ↳ Eof (End of file)

(EOP → End of file)

* One Interesting Concept -

* In our system we have different files stored in different locn.
 But the thing is that when we open any file they open one file
 file bcz of EOF concept (End of
 line), other file doesn't open bcz they
 have End of line.

file 1 file 2 file 3



(Cat)

(EOF)

(Popen/pend)

Some things are gone to

(mailto:gmail.com) > cat H.D of
 = mail@ gmail.com

(date)
 date

gmail
 8/8/15

→ Cat (password)

* If we find a way we can bypass
 this EOF concept.

→ OR, remove EOF

* ~~Ctrl + q~~ → tell End of file.

`wc -l f.txt` (show no of line)

`wc -w f.txt` (show words)

`wc -c f.txt` (show char)

`bc < f.txt`

`bc -l`

→ 3

* we have lots of cmd need input, Cat, WC.

→ we can use first cmd of Cmd1 → give output
as the output of Cmd give input to Cmd2

Cmd 1

→ pipeline

Cmd 2

* If we create one file (lots of words/long) then
we want to ret give only one word →

`vim f1.txt`

`Cat f1.txt`

`grep linux f1.txt`

→ only part (Linux) (line)

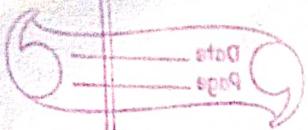
`grep Linux`

→ tell all Linux in top

* # `open -qa | grep python`

remove all the list
of software

show all see
python names of software



man

reduce (Cont)

Page

open -qa | grep Python | wc -l

→ Count word count of all python in log.

(Example of man reduce)

* grep → searching word

open -qa | grep python | wc -l > f.txt

→ save file Content/output of f.txt.

ls -l

* Navigation we see the list but how we know that is file or folder?

* Most of files in Linux we see the color, then we determine the color of folders/file.

→ But in some place in CLI we won't have any color & Extension.

ls -F

→ any forward slash at directory.

* #ls -l (long list)

(cd → directory)

-> file

\$ drop

DP → auto it

* If any start with d those are directory.

ls -l | grep ^d

(This will take/show all d, those whose
heads also d)

→ do for this we have →

ls -l | grep ^d

1. C Regex → Regular expression

(List of files)

ls -l | grep ^d | wc -2

ls -l | grep -n ^d

* If we want to print, Linux command →

grep ^d /etc/fstab

* If we want to print line where line begin,

grep -n /etc/fstab

* For grep and -V means → reverse (Ultra)

grep -V Linux /etc/fstab

→ want to see with line

grep -V -n /etc/fstab

* So, like this except → All file lines with

ls -l | grep ^d | sort & then reverse (ultra)
(cdirectory print)

ls -l | grep -V ^d

(Opposite → All files show)

* If you want to send output to screen (STDOUT) as well as file

date | tee fz.txt =

cat fz.txt

* Want to send outputs to screen & other end also →

date | tee file | nc -w

(one end to other end)

* tee → it will collect data & send file