

Sessions - 07

* Session 06 → Basic Python, Introduction of String.

* I/O Redirection & Piping

* When we run some command they show output & some command does not give output on screen.

Eg:-

date

cat

touch file.txt

ls

* We see the output command showing list by our screen.

*

date, # cat

(output)

(Customize file
Output
your way)

(why you choose
of terminal file
we can also easily
both output. of file output)

① what is file

② Email

③ text msg

* In any OS → we know only full file screen id some kind of devcl.

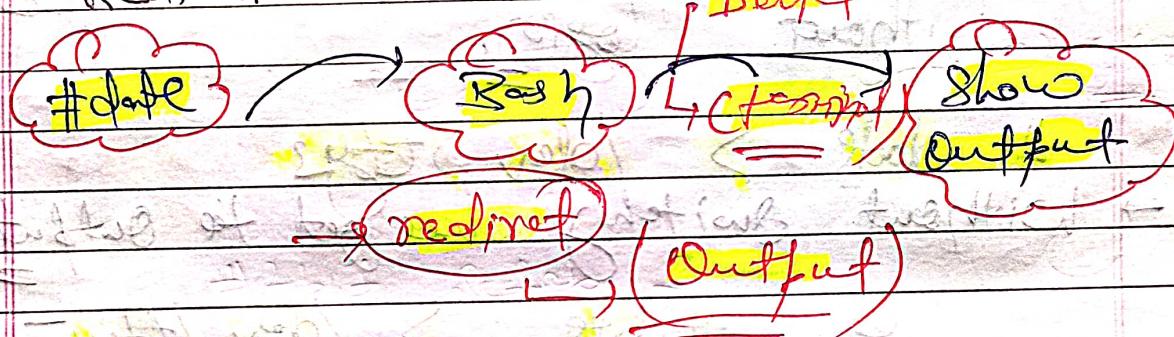
tty /dev/ttys → device name

↳ All file device in Linux managed by /dev folder.

* When you run any command, somebody take your command interact with developmental program show output for you.

* When you run any command bash shell say & does Show output.

* The locat^o/where to show the output of command on display as any other screen i.e. Redirect.



* bash run #tty → bash inform all to send all output of date command, they redirect output to the screen (terminal).

* for sending the output >, symbol to send the output anywhere.

#tty (to find the name of terminal)
date > /dev/ttym (device name of console)
= Go internally by bash & redirect the output to terminal.

* date date internally use tty command to get file & device information.

- * Whenever you run any shell, you can use a redirect redirection to redirect to current terminal.
- * Now, we know the concept now we can run our own command output anywhere, as per need.
- * Now, we are running our command output to different screen in RHEL-8, without login.

date > /dev/stderr
 → Without switching we get the output.

date > /dev/fd/65
 /dev/fd/65 # ffy

Ctrl + Alt + F5

(Now, without login see the output)

Box-Cafe

* without login, without going to that place or without sharing the password, we can show them a specific command by running from any screen
 # date > /dev/fd/66

After running it will show them to see their screen & show them that file.

* without password we can monitor system.

* When we run the cmd, shell take this & run the program is by using symbol \rightarrow symbol \rightarrow command from user.

* \rightarrow Redirection symbol.

=

Output \rightarrow command [Screen] \rightarrow By default goes to (STDOUT) put

* Cmd \rightarrow Shell \rightarrow \rightarrow (other process)

+ redirect

date \rightarrow date.txt

cat \rightarrow date.txt

\rightarrow They run the cmd & store file output in date file.

* \rightarrow feature of shell to create file.

\rightarrow (help us to redirect file output)

* \rightarrow file.txt (first they create file, then they redirect to file.txt)

Bash symbol \rightarrow

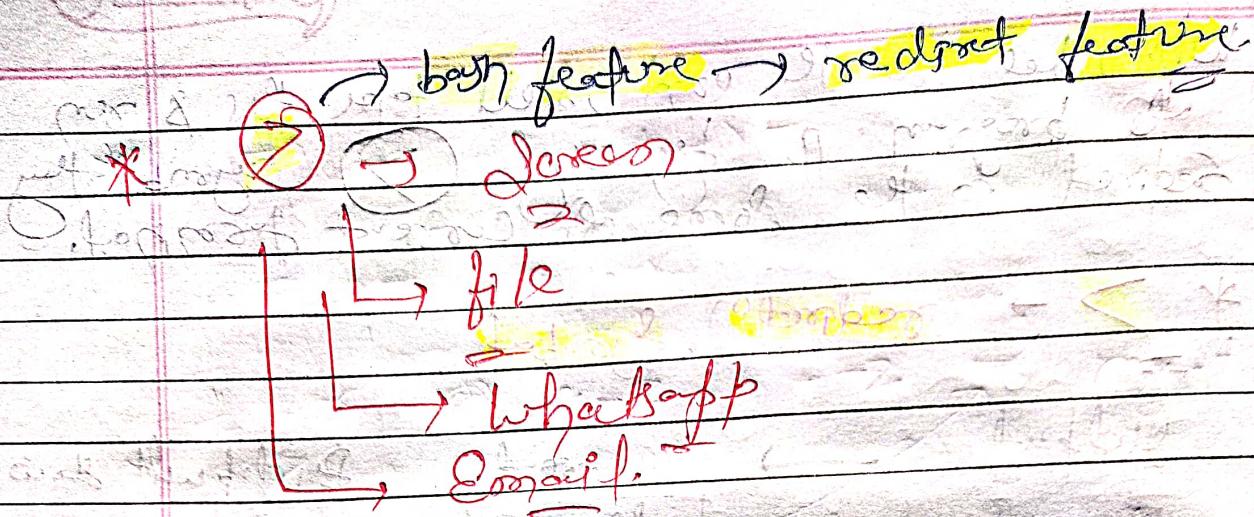
This file created by one program both program/shell.

bash \rightarrow file.txt

on

\rightarrow file.txt

\rightarrow This is file created by cmd, create file (S) - feature of bash.



* H.10 | Command output to mail / whatsover.

Logic used → resolved

* # cat > z.txt

→ If file already
exist
remove old data
new data

→ If the file not exist flag
create one if file exist flag
but copy / output of
file and by 2.

* # cat >> z.txt >> Append

→ If file already exist flag append with
old data.

* Use of cat cmd?

* Before flag, It is all about Output
redirect.

Redirect

Input

Output

* Input Redirection

→ Some cmd give output on console & date, # cat & some cmd doesn't give output but they are # touch, # rm -r.

* Some cmd need to input to run.

type 'A-Z' 'A-Z'

↳ (translates)

Small letters

↳ By letters

→ (By running type command screen stop)

* STDIN (Keyboard)

* STDOUT (Screen)

Type & they after hit Enter they convert
into Capital.

(↳ Enter) → They convert.

* Enter → (Ctrl + C), (Ctrl + D)

They send signal to cmd, they
stop i.e. end of files.

* You can take input from different location,
like from m/c, what's & tee they
run.

* for this they have different input redirect
Symbol → <

* ~~first~~ #redit my.txt

Cat my.txt

ls @ 'a-z' 'A-Z' < my.txt

→ (for all files in directory) symbol
↳ (by default) type it to see this file

B Change/Convert file letters

* All the command give output we can send anywhere.

* All the command file we take input from anywhere.

* Now, we can know the right we can use
of cat command = (Cat command need)

Cat > (input)

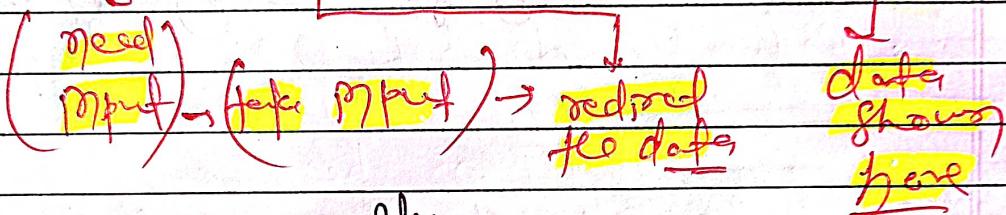
→ Cat is file command, whatever we give file data to Cat command in prompt they directly print file some e.g.

* Cat just print on screen.

* Stop → Ctrl + C
Ctrl + D

- * So far they are using file and to put file of data.
- * First they execute the file if not there, then they read file input & the output that they print they are redirected to file file by (2) redirect symbol.

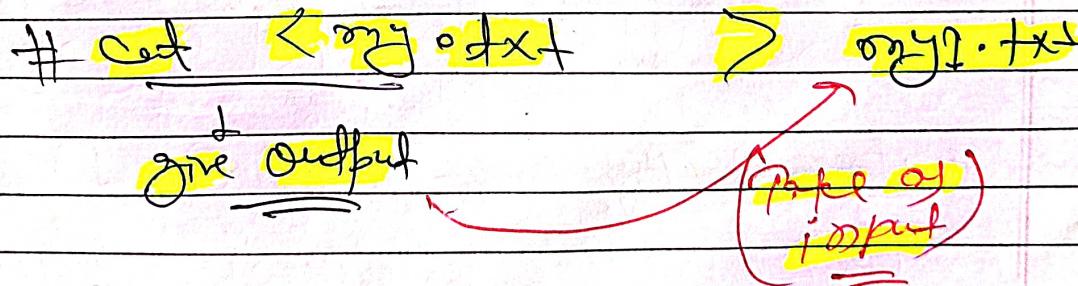
Cat < my.txt



* Cat is not really read the file.

* One more myth Cat create file?

Cat < my.txt
(Show data)



Cat my1.txt

↳ (here Cat cmd also able to copy file)

* The file created by < symbol.

Cat > my2.txt
(take input)

↳ (here can copy file see New window)

Cat my2.txt

↳ (read file data stored)

* If we want to close that file, Ctrl+F, Ctrl+C, Close terminal, exit q.

They won't forget on data =

* ⑧ → file will **everyday**.

Cat → only for 1 file handling.