2/12/1

* Commands

① ls
⇒ List all folders and file

② date
⇒ display in day month date time IST year format

③ cd
⇒ Enter in directory

④ cd ..
⇒ Exits from a directory

⑤ who

⑥ whoami

⑦ uname

⑧ man cd

⑨ cd --help

⑩ ls -r
⇒ list in reverse order

(11) ls -s
⇒ display sizes

(12) cal
⇒ display calender

(13) pwd
⇒ display present working directory

(14) cp <filename> <path>
⇒ copy a file

(15) cat > <filename> [file ka extension
⇒ create a file ni lakhe to bhi chale]

(16) ctrl + d
⇒ to save and exit file

(17) mkdir
⇒ make directory

(18) help

(19) mv

Page No. :
Date : / /

Outputs

① ls

→ Desktop Documents Downloads examples.desktop Music Pictures Public Templates Videos.

② date

→ Mon Dec 2 12:10:37 IST 2019

③ who

→ 20180610011008O tty7 2019-12-02 12:02 (:0)

④ whoami

→ 20180610011008O

⑤ ls -r

→ Videos Templates Public Pictures Music examples.desktop Downloads Documents Desktop

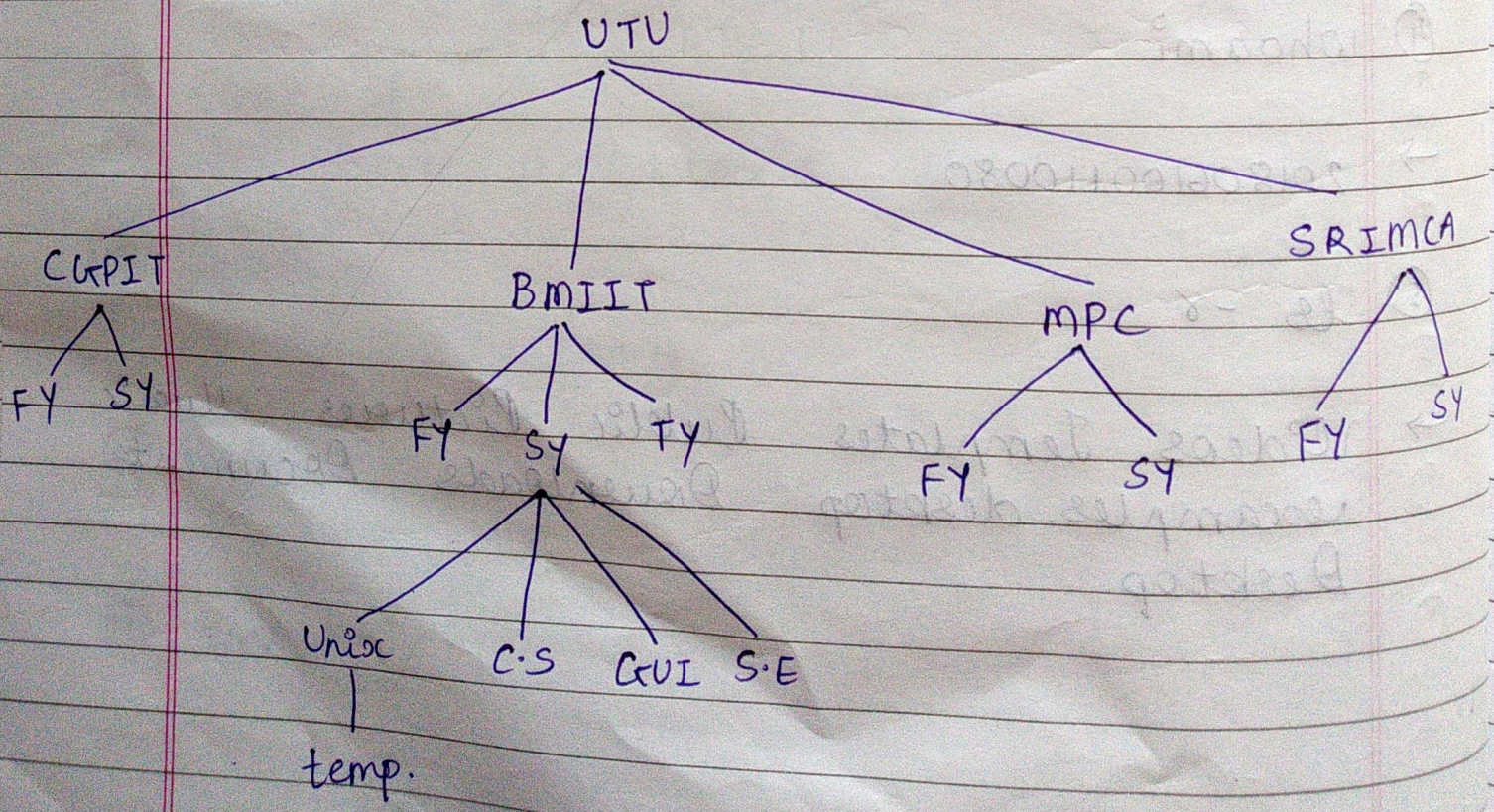
⑥ ls -s

→ total 44

4 Desktop 4 Documents 4 Downloads
12 vsample.desktop 4 Music 4 Pictures 4 Public
4 Templates 4 Videos

⑦ cp oop ~/Desktop/UTU/GPIT/SY

→ copy oop file in SY directory



3/12/19

* Purpose of Unix

① Multiple access [Koi interference in aave]

② Security

• Introduction :-

⇒ Unix is a multi-tasking, multi user operating system. It was developed by Ken Thompson, Dennis Ritche, Rudd canady, Doug Mcllory. It was developed in 1969 - 1970.

(20) ls -R
⇒ shows directory and subdirectories

(21) ls -a
⇒ list all files

Note:
- command -option
  for using options

- command " +%. operator"
  for using %. operators.

Eg: date " +%.s%.r"

(22) cp <filename> <path>\<newfilename>
⇒ To rename the copy file.

(23) cp <source path>\<destination path>\<filename>
new/rename
optional
↓
<file name>\

⇒ To copy from directory. without going into subdirectory

(24) cat >> <file name>
⇒ To append a file

(25) rm <filename>
⇒ To delete a file

(26) rmdir <directoryname>
⇒ To delete a directory

[folder ni andar koi file hoi to e directory delete thse nai]

[agar delete krvu hoi to directory empty hovi joie]

(27) more <filename>
⇒ Show content of file

(28) mv
⇒ rename file name
Eg: mv <old file name> <new file name>

(29) ln

(30) rm/rmdir
⇒ rmdir ⇒ remove directory
rm/⇒remove file

(31) find

(32) ls -l

6/12/19

Practical list 1

Page No. :
Date : / /

① whoami

Output : 20180610011O080

② ~~date "+%R"~~ date "+Today date is - %R"

Output: ~~10:55~~ Today date is 10:55

③ date "+%d/%m/%Y"

Output : 06/12/2019

④ date "+Today's date is %d/%m/%Y. Current time is : %R"

Output: Today's date is 06/12/2019. Current time is : 10:58

⑤ date "+%j"

Output: 340

⑥ date "+Happy %A"

Output : Happy Friday

⑦ cal

Output: December 2019

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

⑧ date "+%-W"

Output: 48

⑨ cal 2019

Output :

⑩ who

Output: 20180610011008O tty7 2019-12-06 08:35 (:0)

⑪ who -uH

Output: NAME LINE TIME IDLE PID COMMENT
20180610011008O tty7 2019-12-06 08:35 02:36 1664

Page No. :
Date : / /

(12) cat --help

Output:

(13) tty

Output: /dev/pts/2

(16) more -V

Output: more from until-linux 2.34

(14)

Work

Linux Java C Sharp

Program Script Program applications

Data

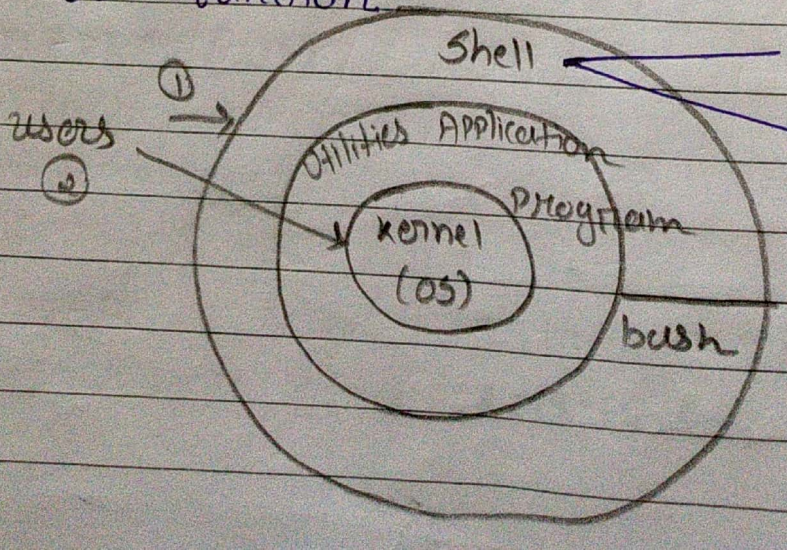
Directory Making → mkdir

Making text file → Cat > friendlist.txt

Copy → cp -r friendlist.txt /bmiit/Stud/2018061001100073/Work/CSharp/Application

* Unix Structure :-

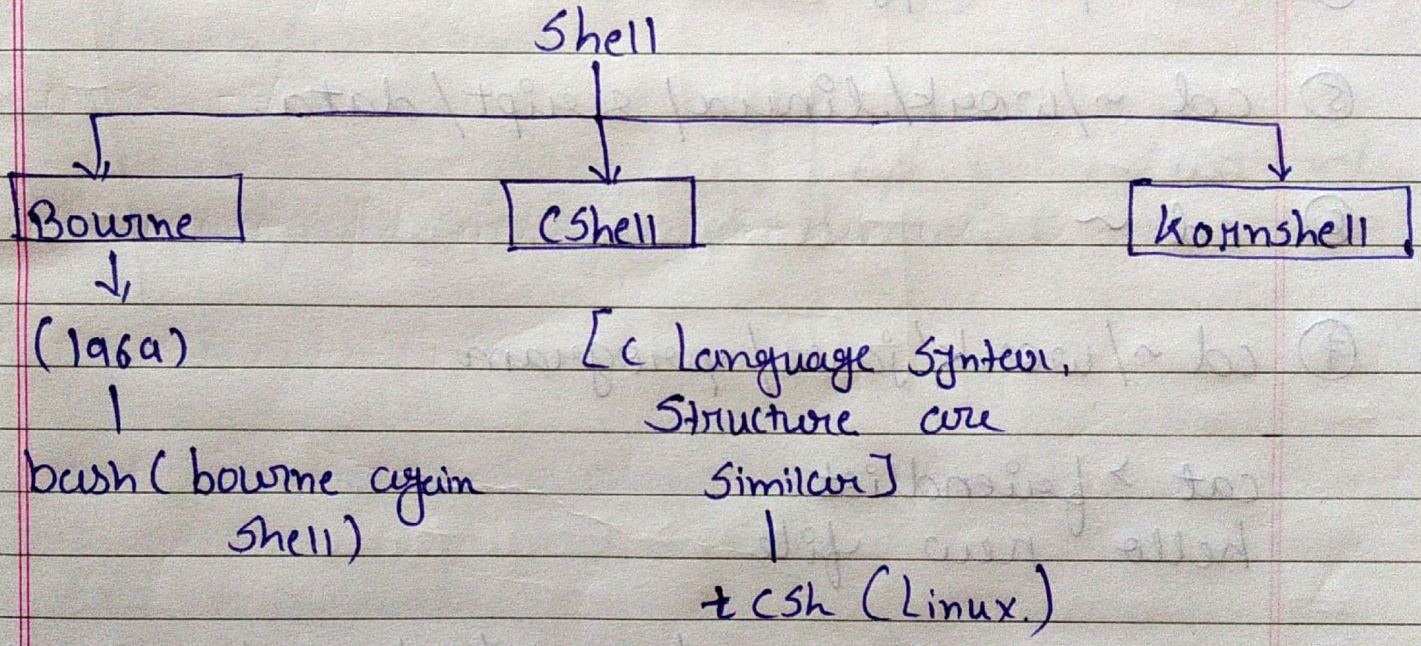
- It reduces the load or cost of CPU in hardware. It is also help hardware to do function.



Request → Response

① Users → Shell → Kernel.

② users → kernel → (directory users can connect with kernel or OS)



16/12/19

Practical List 2

1. cd work
2. cd ~/work/csharp/application
3. cd ..
4. cd ~/work/java
5. cd ~/work/linux/script/data
6. cd ~
7. cd ~/work/java/program

   cat > friendlist
   hello new file
8. cp friendlist ~/work/csharp/application
9. cp friendlist ~/work/java/program/personal
10. cp -r ~/work/linux/script/data ~/work/linux/program
11. cd ~/work
    ls -R

(12) rm friendlist
cd ..
rmdir application

(13) mkdir -p application/code/data

(14) [~/work/csharp/application$] cd ~/work/linux/program

(15) [~/work/java/program$] cp friendlist ~/work/java/program/namelist

(16) ls -l

(17) mv namelist name

(18) cat >.mydata
name: pooja Maldikar
Sem: 4th
enrollement number: 20180610011 0080

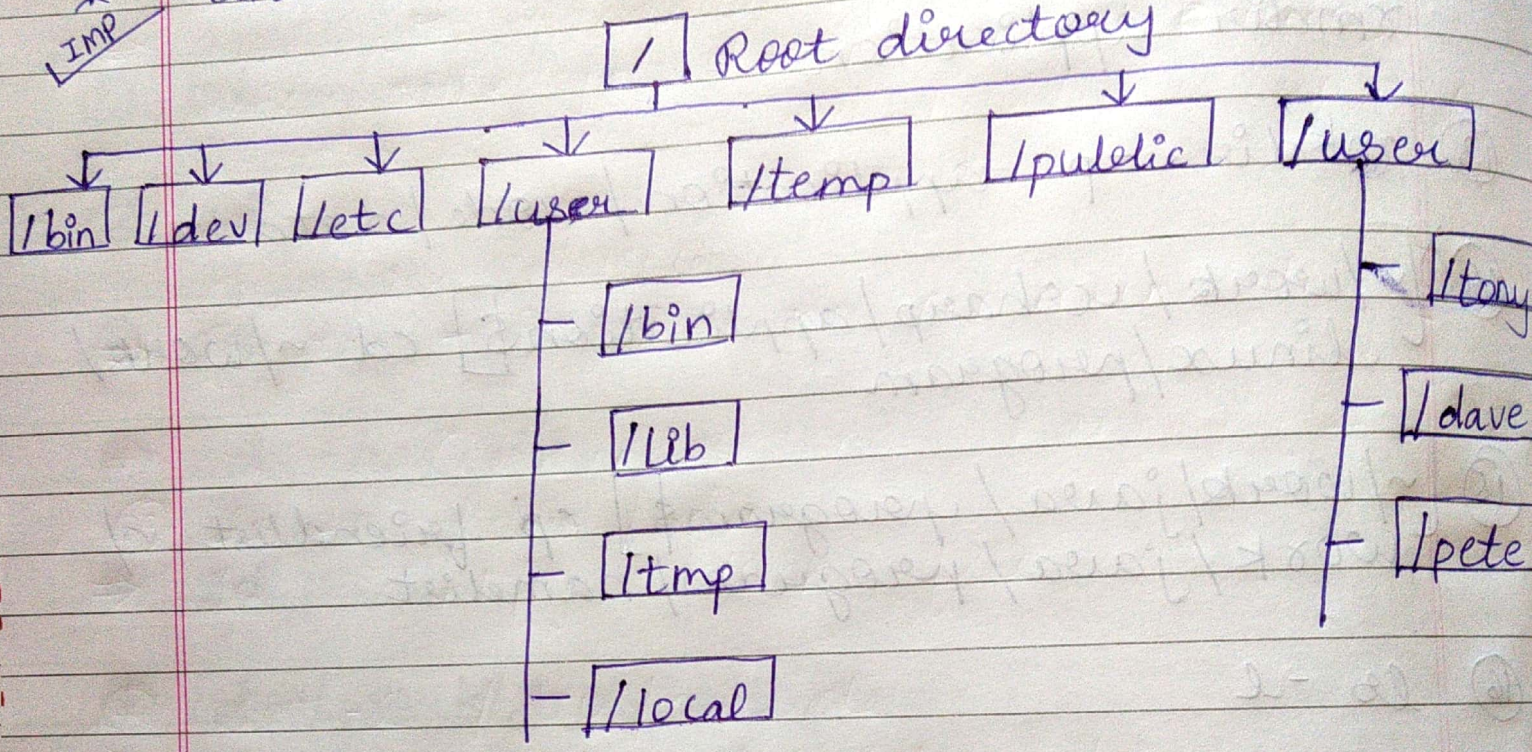
(19) ls .-a

(20) ls --t

(21)

17/12/19

* IMP Unix File System [Diagram of unix library structure]



File structure

[ls -l]

* File types in Unix

- 7 types of file can be created

1. Normal file (-)
2. directory (d)
3. symbolic link (l)
4. named pipe (p)
5. block device (b)

⑥ character device (c)
⑦ socket (s)

- symbolic link (address)
⇒ stores reference path of another file.
like pointer in C.

+ hard link
+ soft link

- named pipe & socket
⇒ used for communication

⇒ named pipe → half duplex only one way communication (ek hi data send thai ka to receive)

resources are not kept on hold

⇒ socket → full duplex two way communication (

resources are on hold until both communication are end

- block device
⇒ use to manage hardware peripherals
(PC sathe je device connect karie)
[laptop ni andar na hardware communi

- character device
⇒ [laptop + Projector nu communication]

Commands

(33) ls -l /dev/sda
⇒ To identify block file

(34) ls -l /dev/console
⇒ To identify character file

(35) mkfifo <file name>
⇒ create a pipe

(36) ln <file name> <filename>
(original file, reference file nu name)
⇒ create symbolic link [hard]

(37) ln -s <file name> <filename>
(reference file)
⇒ create soft symbolic link

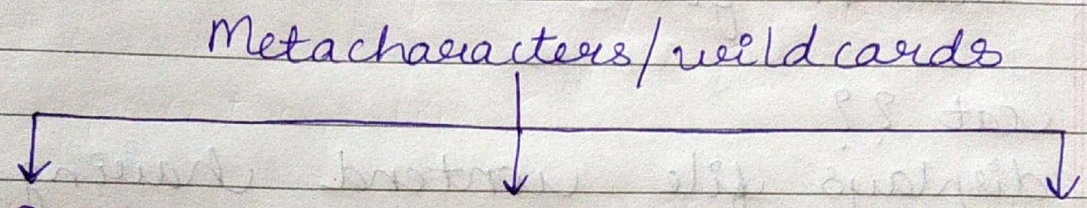
(38) echo
⇒ display whatever provided in argument

19/12/19

Page No. :
Date : / /

⊛ Metacharacters / wildcards

• Metacharacters are used with file name only.



[Single character] [Set] [any]

⊛ Single character

Eg: echo ?

⇒ ama shell single character na file name ne replace kre.

Eg: agar directory ma f and d naam ni file hoi to
echo ?

⇒ f d [baner file nu name replace krse

⇒ pn agar file nai hoi evi single character vali to "?" mark j print thai.

Page No. :
Date : | |

Eg: echo ????
⇒ display 4 character file name from current directory.

Note:
- file name start with . are hidden files.

Eg: cat ??
⇒ displays file content having 2 character file

* Set

Eg: echo f[0123456789]
↓
only one character j replace thai

Eg: echo f[0-9]

Eg: echo [z-A]
aavu lakhe to e z or - or A mathi select krse.
- Sequence maj lakhvanu ASCII na.

Eg: echo [A-Z a-z][A-Z a-z]
⇒ match 2 character file name

(*) any

Eg: echo *
replaces all files

Eg: echo *.*
⇒ file name having . will display

Eg: echo *[0-9]
⇒ all files from current directories where last character is digits.

Commands

(39) touch
⇒ creates all files in one line.

Eg: touch a b c
⇒ It will create 3 files of a, b, c with empty content.

(40) file *
⇒ List all information regarding file (only show current directory files)

Page No.:
Date: / /

Commands

(41) echo
⇒ display current directory path

(42) passwd
⇒ for changing password

(*) Inbuilt calculator in Unix

- bc bench calculator

(*) Two ways of using
(1) pipe
(2) interactive mode
↓
write bc to start
to exit do ctrl d.

- Using pipe
x=10 y=50
write: echo 10 + 50 | bc
write: echo "addition: 10 + 50" | bc

Page No. :
Date : / /

eg: x = 10
y = 50

⇒ echo $x. + $y | bc

⇒ echo $x (*) $y | bc

↓
wildcard tarike na use kre
ethe agad slash karie

- bc is a command for ~~execution~~ starting a calculator.
- It opens the default calculator in unix which is also known as the bench calculator.
- There are two ways to use the bench calculator
  1. Invoke bc as a pipe of some or other command
  2. Invoke bc in interactive mode

Page No. :
Date : | |

Note:

- While using the bench calculator as a pipe a backslash is used before asterick sign to escape the asterick or else it will be treated as a wild card character.

23/12/19

* Shell

Shell act as
① interpreter
② executing command
③ create script

3 shell :

① Bash/(Default/standard shell)
Bourne

- The shell is a command line interpreter it translates command enter by the users and converts them into a langage.

understood by the kernel. Users communicate with the kernel through unix shell.

* Shell script

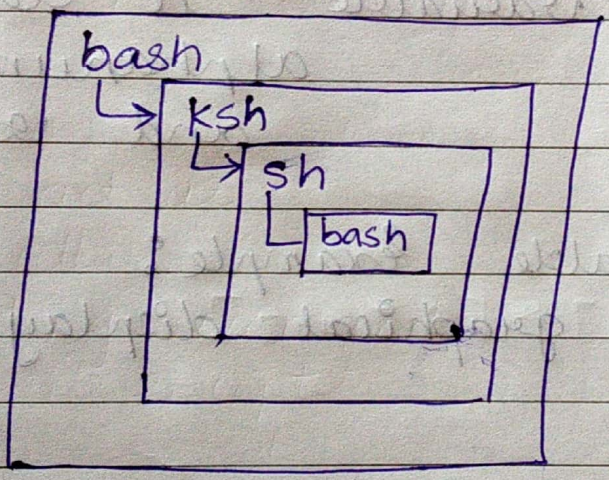
- A shell script means a file which contains a set of commands inside the file. As this file contains command it can be considered as an executable file.

① Bourne shell [$ bash]
- Created by Steve Bourne
- Extension = .sh

② C shell [$csh]
- Created by Bill Joy
- Extension = .csh

③ Korn shell [$ ksh]
- Created by David Korn
- Extension = .ksh

- To exit from shell write exit in command prompt
- To exit from parent directory shell write logout command prompt.
- logout is not used for child shell exit.



Q→ Justify the statement Unix shell follows a hierarchy while working on multiple shells.

Q→ Demonstrate the process of parent-child shell in Unix.

Or

Show how unix shell implements hierarchy while working on multiple shells. [Diagram]

* Shell variables

. Two types of variable
① System variable → always in capital
② Userdefined variable → it can be alphanumeric / and small letter

⊙ System Variable Example:
DISPLAY → The graphical display to use
HOME
HOST
LOGNAME
PS1 [primary Prompt]
PS2 [secondary Prompt]
USER
GROUP
EDITOR
PATH

Eg: echo $ SYSTEMVARIABLE_NAME

Page No. :
Date : / /

- To create a variable:

```
$x=100
$y=50
$x1="hello"
```

- To display:

```
echo $variablename
```

Practical 3

1. echo *
2. echo l*?
3. echo *1
4. echo ???
5. echo l[a,e,i,o,u]st.sh
6. echo [0-9]*
7. echo *[0-9]
8. echo [A-Za-z]*[A-Za-z]
9. echo *.*
10. echo ??st?*

1. cp *.txt ~
2. cp user0?* ~/Desktop/user
3. cat l?st.sh

④ rm *prg*

⑤ rm [A-Za-z]*

⑥ mv data01 data1*1

⑦ mv [A-Za-z]* ~/Desktop/work

⑧ rm [A-Za-z][A-Za-z][0-9]

Page No. :
Date : / /

Commands

(43) echo $ 0
→ show in which shell it is.

(44) ~~echo~~ $ ksh
→ moves to written shell

(45) exit
→ It exits from the current shell.

- Uses of Shell

1 Customizing your work environment
2 Automating your daily task
3 Automating repetitive task
4 Executing important system procedure like shutting down the system.
5 performing same operating on many files.

- Responsibilities of shell

1. Program Execution
2. Variable and file substitution
3. I/O redirection
4. Pipeline hookup
5. Environment control

Page No. :
Date : / /

6. Interpreted Programming Language

(46) export variablename
→ That variable is declared globally. use in all shell

26/12/19

Page No. :
Date : / /

* cat man

① -A, --show-all
⇒ equivalent to -vET

② -b, --number-nonblank
⇒ nonempty output lines, overrides -n

③ -E, --show-ends
⇒ display $ at end of each line

④ -n, --number
⇒ number all output lines

⑤ -s, --squeeze-blank
⇒ suppress repeated empty output lines

⑥ -T, --show-tabs
⇒ display TAB characters as ^I

Output

① cat -A

Output: abc anji [on prompt]
abc anji$

② cat -b

Output: this is my first program [on prompt]
1 this is my first program.

③ cat --show-ends

Output: oop fop ds [on prompt]
oop fop ds$
java [on prompt]
java$

④ cat -n

Output: this is my first file [on prompt]
1 this is my first file

⑤ cat -s

Output: pooja [on prompt]
pooja

⑥ cat -T

Output: p o [on prompt]
p^Io

Page No. :
Date : / /

* Echo man

① -n
⇒ do not output the trailing newline.

② -e
⇒ enables interpretation of backslash escapes

③ -E
⇒ disables interpretation of backslash escapes (default)