ASSIGNMENTS / UNIT TESTS / QUIZ BOOK

Name of the student CHANDANA · MANDAVA

Branch : CSE SEM : 5 Section : A

USN : 1 K N 1 9 C S 0 1 7

Subject Name: UP Subject Code : 18CS56

Assignments	Date of Submission	Title & Description	C O No.	Marks		Signature of Student
				Maximum	Marks Obtain	
I	4/12/21	Assignment -01	1			C.Mandava
II						
III						

Unit tests/ QUIZ	Date	Title & Description	C O No.	Maximum Marks	Marks Obtain	Signature of Student
I	29/11/21	CT-01				C.Mandava
II						
III						
IV						
V						

Name of faculty SRUJANA ·

Marks Obtain / 10

Signature of faculty

Signature Of H.O.D

UP -

3/12/21 ASSIGNMENT-01

PAGE NO

B) 09) What is chmod? Explain in detail 2 ways of changing file permission with examples.

→ Using chmod, we can change the file permissions and allow the owner to execute this file. The command can be used in 2 ways: • In relative manner by specifying the changes to current permissions
• In absolute manner by specifying final permissions.

⇒ Relative permissions: Syntax: chmod category operation permission filename(s)

Eg: If I/P is rw- r-- r--
Resultant output: rw- rw- rw-
we can change in following manner

$ chmod go + w kns

⇒ Absolute manner: Here, we need not to know the current file permissions. We can set all 9 permissions explicitly. A string of 3 octal digits is used as a expression. The permission can be represented by one octal digit for directory for each category, we add octal digits.
Read permissions - 4, Write permissions - 2, Execute permissions - 1
Eg: rw- r-- r--
If the resultant should be in form of
rw- rw- rw-
Then the following could be

$ chmod 666 kns

B08) Explain the following Unix commands with Examples
1) mkdir 2) rmdir 3) pwd 4) cd.

→ 1) mkdir: making directories.
directories are created using mkdir command. The command is followed by names of the directories to

be created A directory patch is created under current directory like this: $ mkdir patch.

You can create subdirectories with a mkdir command:

$ mkdir patch dba doc

for instance the following command creates a directory tree

$ mkdir progs progs/c progs progs/java progs

This creates 3 subdirectories - progs, cprogs, javaprogs under progs. You can't create subdirectories before creation of parent directory. System refuses to create a directory if directory is already exists and user doesn't have permission to create directory. The order of specifying arguments is important.

2) rmdir: Removing directories :- The rmdir command removes the directories. You have to do this to remove progs

$ rmdir progs. If progs is empty directory, then it will be removed from system. following command works with rmdir

$ rmdir progs/c progs progs/javaprogs progs.

You can't remove a directory if it is not empty, which doesn't exists in system and if you don't have permission to do.

3) pwd: checking your current directory

Anytime user can know the current working directory using pwd command.

$ Pwd /home/kumar. Like HOME it displays the absolute path

4) cd: Changing the current directory :- User can move around file system using cd command, when used with the argument, it changes the current directory to the directory specified as argument, progs:

PAGE NO.

$Pwd /home/kumar
$ cd progs
$ Pwd /home/kumar/progs Here we are using the relative pathname of progs directory.

Q07.) Discuss ls command with any 4 major options with examples

→ ls command is used to obtain a list of all filenames in the current directory. Syntax : ls [option]... [file]...

Option :- a) $ls -l.

To display all information about files / directories.

Eg :- $ ls -l

total 72

-rw-r--r-- 1 kumar metal 19514 may 10 13:45 ch

(ii) -x :- This option is used with the ls command to print the list of the filenames in the current working directory in the form at of multiple columns

Eg :- $ ls -x

kit student USN ATC.

(iii) -F (Fuppercase) : Previously when we used ls command -F option with ls command helps us to print the name of the files with identification of directory and executable files. The filenames which preceed with '*' are executable files and those who precede with '/' are directory files Eg :- $ ls -F

* kit / ef * USN.

(iv) -a :- This option when used with ls command prints all the files from the current directory. filenames which start with '.' are hidden files of the current directory while which start with '..' are hidden files

of the parent directory of the current directory.

Eg:- $ ls -a
-hello .. hidefile abc.

Q6) Explain the working of man command in detail

→ man is the system's manual viewer, it can be used to display manual pages, scroll up and down, search for occurrences of specific text, and other useful functions. Each argument given to man is normally the name of a program, utility or function. The default action is to search in all of the available sections, following a pre-defined order and to show only the first page found, even if page exists in several sections.

A man page is divided into a number of compulsory Optional sections. Every command doesn't have all sections, first 3 are seen in all man pages

(NAME, SYNOPSIS and DESCRIPTION)

↳ presents the online intro to command ↳ Shows syntax ↳ provides detailed info

Eg:- man syntax

$man [option] command name

Eg:- $man man
view the manual page for the man command.

Q5) Discuss in detail Absolute and Relative path name with Example

→ Absolute pathname: Directories are arranged in a hierarchy with root (/) at top level. positions of any file within the hierarchy is described by its pathnames. Elements of pathnames are separated by (/). A pathname is

absolute if it is described in relation to root. Thus absolute pathnames begin with /.

Eg:- •/etc/passwd •/home/kumar/progs/cprogs.

- Relative pathname: A pathname can be relative to your current working directory. Relative pathnames never begin with '/'. Users can move from working directory /home/kumar/progs/cprogs to home directory /home/kumar by using 'cd' command like:

```
$ Pwd /home/kumar/progs/cprogs
$ cd /home/kumar
$ Pwd /home/kumar
```

Q04. Explain the following:-
i) /bin (ii) /dev (iii) /etc iv) /home

→ (i) /bin: •/bin and /usr/bin: these are the directories where all the commonly used UNIX commands are found

(ii) /dev: This directory contains all device files. These files don't occupy space on disk there could be more sub directories like dsk, rdsk etc

(iii) /etc: this directory contains the configuration files of the system. You can change a very important aspect of system functioning by editing a text file in this directory. Your login name and password are stored in files /etc/passwd and etc/shadow

(iv) /home: •/home:- on many systems users are housed here. Kumar would have his home directory in /home/kumar.

Q03) Write a short notes on
i) Command Structure ii) Command Options

→ i) Command Structure.
Commands are entered at shell prompt. The components of the command line are:
- The verb;
- any options required by the command
- the command's arguments (if required)

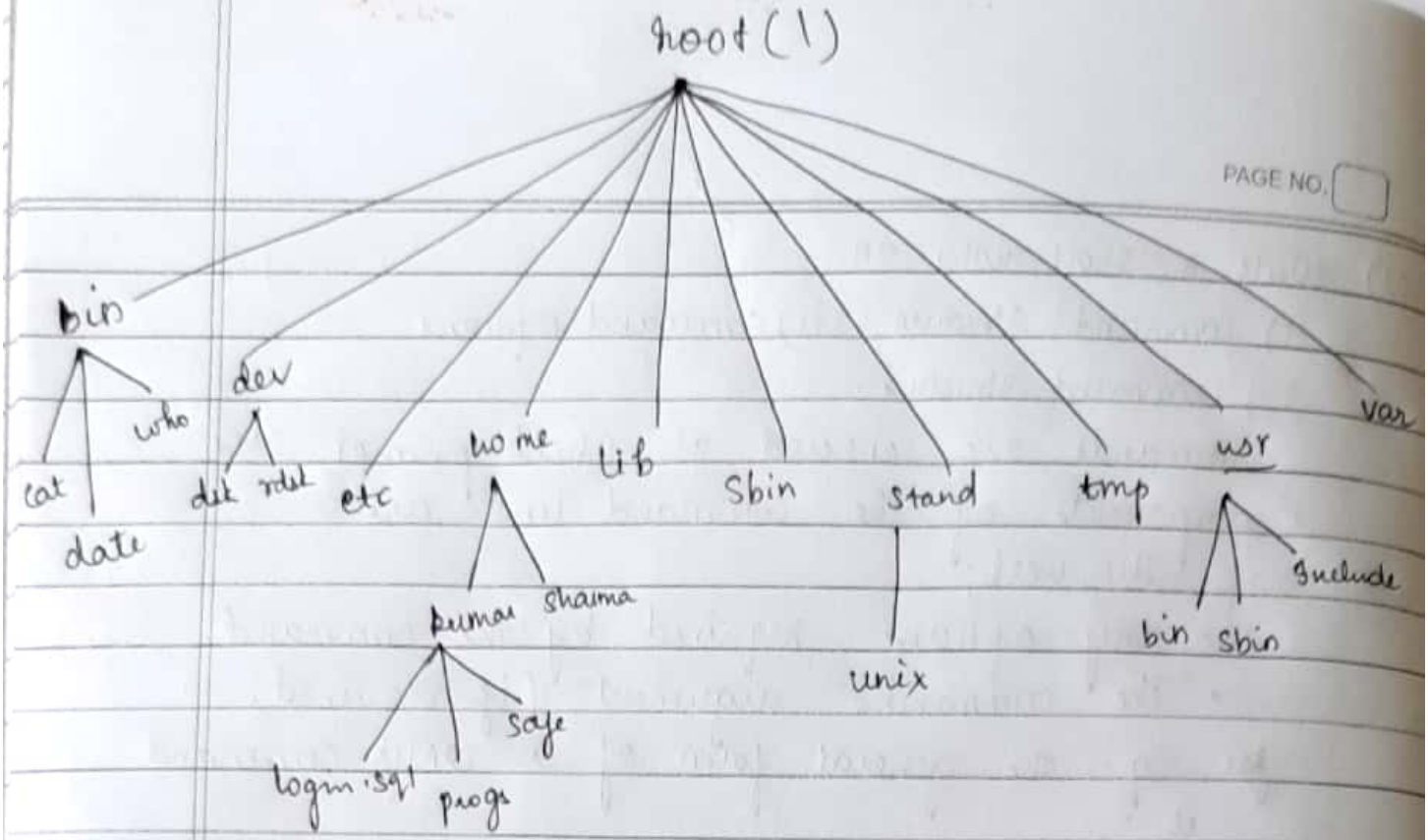
for Eg:- the general form of a UNIX command is:

$ verb [noption(s)] [argument ([5)]

ii) command option:-
verb: is the command name. The command Indicates what action is applied
argument: provides additional information to the command
Note: option MUST come after the command and before any command arguments. Options should not appear after the main arguments.
if options are enclosed within the [] then options are not mandatory else it is compulsory
if arguments are enclosed within the [] then options are not mandatory else it is compulsory

Q02) Explain with neat diagram of the UNIX file System tree.

→ Parent Child Relationship / organisation of files.
* The files in unix are related to one another. The file system in unix is a collection of all these files organisation in hierarchial structure.



→ The root is actually a directory and has number of subdirectories under it. Every file apart from root, must have a parent. In parent-child relationship, the parent is always a directory. Thus the home directory is the parent of kumar. If you create a file login.sql under the kumar directory, then kumar will be parent of this file.

Q.01) Explain the following :-
a) PATH b) HOME c) su d) ~~type~~ passwd.

a) PATH : The sequence of directories that the shell searches to look for a command is specified in its own path variable

Eg:- $echo $PATH
/bin :/usr/ bin :/ usr / local /bin :

b) HOME: when you login to the system, Unix places you in a directory called home directory. It is created by the system when the user

account is created. The shell variable HOME knows the home directory

$echo $HOME
/home/kumar.

c) su: (Substitute user): to execute commands with the privileges of another user account. When executed it invokes a shell without changing the current working directory

d) passwd: The passwd command is used to change the password of a user account.
syntax: passwd [options] [LOGIN]
The /etc/passwd file is the file to keep track of every user on the system.