Theory

Note: Bash shell Bourne + Korn

1. Bourne shell

It was created by Steve Bourne. This shell is present in every UNIX operating system. The default prompt of Bourne shell is â€œ**$**".

2. C Shell

It was created by Bill Joy at the University of California. It has two advantages over the Bourne shell.

i. It allows aliasing of commands i.e. the user can decide the name.

ii. Instead of typing the entire command, you can simply use the short alias name.

One more advantage of this shell is that it maintains history of the previously executed files. The default prompt of C shell is â€œ**%**".

3. Korn Shell

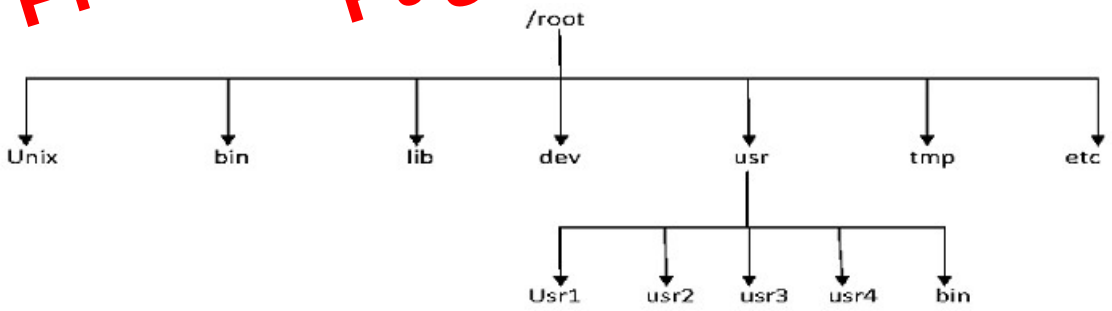
It was created by David Korn at AT & T Bell laboratory. It is a superset of Bourne shell. It is widely used for any types of commands to execute because it is very powerful. The default prompt of korn shell is **"$**".

Bash: This is another type of shell which added some extra features of Bourne shell. Its named as Bash.

Write a short note on UNIX file system.

In UNIX, everything including hardware devices is treated as a file. The file maybe a program, file, a directory or a sub-directory. It reassembles a tree structure with its root at the top. The UNIX file system begins with a directory called root. The root directory is denoted as slash (/). Branching from the root there are several other directories called bin, lib, usr, etc, tmp, dev and unix. These directories are called sub-directories and their parent is the root directory. Each of these sub-directories contains several files as well as other directories (sub-sub-directory). Following diagram describes the structure of UNIX file system.

Preview from Notesale.co.uk
Page 3 of 21



The following table describes all these sub-directories

Directory	**Contains**
unix	UNIX kernel is present in this directory
bin	Binary executable file
lib	Library function
dev	Device related files

b. tty: Terminal number where the process is running.

c. TIME: Total time for the process to complete its execution (Burst time).

Â

ps command with different options.

a. Ps "f"

This command will display the full information about the process that is running at present. The informations are:

User id, PPID (present PID), PID, tty, stime (starting time), etc.

b. Ps "e"

This command shows all the processes including the system processes running on the system.

c. Ps "a"

This command shows all the user processes but not the system processes.

Explain nohup.

When a user logs out with one or more background processes running which is an unnatural death. In such case, when the user does not want to hang up the background process and the user logs out, the nohup command is used. Therefore nohup command lets the user to run a program continuously even after the user has logged out.

Preview from Notesale.co.uk
Page 10 of 21

Syntax: $nohup sort -0 filename1 filename2 &

Example: $nohup sort -0 abc.dat xyz.dat &

Write a brief note about kill().

Kill command is used to terminate the process forcefully.

A background process can be terminated using kill command.

A particular process can be killed by giving its PID (process id).

More than one process also can be terminated by using one kill command.

Write a brief note about fork().

The fork() system call is involved when a parent process wants to create a child process.

When fork() is executed it will return one unique PID for the child process i.e. when a process is forked, the child gets a new PID.

Write a brief note about exec().

The fork() is used to create a child process only. It is not possible to run the child process by using fork().

To start execution for a child process, parent process calls another system call i.e. exec().

Write a brief note about wait().

Sometimes it is required to stop a process execution temporarily.

This is possible by invoking another system call wait().

To start a process which is stopped temporarily the parent process will invoke another system call i.e. init() or exec().

```
do
fact=`expr $fact \* $i`
i=`expr $i + 1`
done
echo "The factorial of $num is:"
echo $fact
```

5. To print a string in reverse order.

```
clear
echo "Enter the string"
cat>str
len=`cat str | wc -c`
len=`expr $len - 1`
row=2
col=0
while [ $len -gt 0 ]
do
ch=`cut -c $len str`
tput cup $row $col
echo $ch
len=`expr $len - 1`
col=`expr $col + 1`
done
echo
```

Preview from Notesale.co.uk
Page 17 of 21

6. To count the no of vowels in a given string.

```
clear
echo "Enter the string"
cat>str
len=`cat str | wc -c`
len=`expr $len - 1`
count=0
while [ $len -gt 0 ]
do
ch=`cut -c $len str`
case $ch in
A|a)count=`expr $count + 1`
;;
E|e)count=`expr $count + 1`
;;
I|i)count=`expr $count + 1`
;;
O|o)count=`expr $count + 1`
```