

## MACHINE EXERCISE-5

### SHELL PROGRAMMING

Unix shell is a interface b/w the user and the operating system itself and form one of the major components of unix operating system. The shell incorporates a powerful programming language that enables the user to explore the full power and versatility of unix. The shell itself interprets the command in the shell programming and execute them.

Shell script contains a variety of valid unix commands, while writing shell programs we would be using these commands in a slightly different light.

Shell available are integral part of shell programming. They provide the ability to store and manipulate within a shell program. Shell keywords are the words whose meaning has already explained to shell.

## PROCESS BASED COMMANDS IN LINUX

PS:- To see which process are running at any instant  
\$ ps

Unix assign a unique number to every process running in memory. This number is called process ID or PID.

The output of ps shows the PID for the processes being run by us when ps was executed. The output also shows the terminals from which the processes were launched.

PS-a → To find out which processes are running for the other users who have logged in. execute ps command with -a - option -a stands for the processes of all the users.  
\$ ps -a

PID	TTY	TIME	COMMAND
2269	3a	0.05	sh
2269	3a	0.00	ps -a
2100	3b	0.00	sh

ps-f → There is lot of information that UNIX stores about each running processes. This can be obtained by using the option f standing for full string.

\$ ps -f

ps -e → There is a scheduler process running at all times in memory in which it records the CPU times & memory amongst all other process and users

\$ ps -e

ps -e stands for every process running at that instant.

Background Process → Most of the system processes run in the background.

While the user execute their process in the foreground if the user desire then he can also run his process in the background using UNIX facility the user can run time consuming tasks like sorting a large file and save the output in a file in the background.

To run a process, UNIX provides ampersand (&) symbol. If this symbol is placed at the end of the command then the command will be executed in the background.

\$ sort filename 1 > filename 2

When you run a process in the background a number is displayed on the screen i.e. PID of process.

CHANGING PROCESS PRIORITIES: The processes with higher priority would get a time slot earlier. It would be first earlier than other processes in unique.

The priority of a process is decided by number associated with it. This number is called 'Nice' value of process higher the next value of process, lower the priority.  
\$ nice cat filename

The value of process can range from 0 to 19. in which 20 as default nice value of process

SCHEDULING OF PROCESS: Any OS must provide tools to permit scheduling of processes as per user's system requirement.

UNIX can schedule process to get years UNIX manages to remember the processor to be executed and goes about executing time  
\$ ps -e: graph

at command: This command is capable of executing future data as time. Unix can be specified at the command prompt or can be stored in a file and at command can use file to execute the commands  
\$ at

and pressing ctrl+d at command displayed the job-id data.

The Batch Command - The command to be executed at a process moment in time we let system decide the best time for executing our command. The way to achieve this is through command called batch.

\$ batch  
'b' extension given to our job-id signifies that it has been submitted using the batch command.

WHO Command - The who command is more powerful and dialogues into the system currently  
\$ who

**FILE RELATED COMMANDS:** Unix provide a number of command that help us manipulate files are called filters.

**CUT:** The cut picks up a given number of character or fields from the specified file.

```
$ cut
```

```
$ cut -f 277 filename
```

The cut command assumes that fields are separated by tab character.

**GRAPH:** Graph is an acronym for globally search a regular expression and permit it. The command searches the specified input fully for a match with the supplied pattern and displays it.

```
$ grep picture newfile.
```

This would search word 'picture' in newfile.

**PRINTING COMMANDS:** They are as following:

**TOUCH COMMAND:** It does not only create empty files but it also allows you to change the modification and access time of a file. Touch comes into pictures when you want to change these times without really

accessing for modifying the file.

```
$ touch -a my file.
```

**FILE COMMAND:** You have created a bunch of files, but have lost track of what does each file contain.

It provide us the file command.

```
$ file
```

```
a.out: i386 executable NOT stripped
```

```
bin: empty
```

```
bin: directory
```

```
call: out: ascii text
```

```
etc: directory
```

**CAL:** cal command is for calendar. It displays the current date and time. It is capable of printing calendar for any year of the given month.

```
$ cal
```

**SORT COMMAND:** It can be used for sorting the contents of a file. It can also merge multiple sorted files, and store the result in the specified output file.

```
$ sort my file.
```