

\$ passwd geek

③ Dealing with a user configuration file.

\$ cat /etc/passwd

This contains prints the data of the configuration file. This file contains information about the users in the format.

\$ username : x : user id : user group id : /home/username : /bin/bash. (for modify information)

④ The command to change the user ID for a user.

\$ sudo usermod -u 100 test

⑤ Command to modify the group ID of an user.

usermod -g new-group-id username.

\$ sudo usermod -g 100 test.

⑥ We can change the user login name using usermod command.

Sudo usermod -l new-login-name old-login-name

~~Sudo usermod -C John-Pi John-Roe~~

⑦ The command to change the home directory.

\$ usermod -d new-home-directory-Path username

⑧ You can also delete a username. The below command deletes the user whose username is provided. make sure that the user is not part of a group. If the user is part of a group then it will not be deleted directly, hence we will have to first remove him from the group and then we can delete him.

\$ userdel -r username

~~\$ sudo userdel -r Pi~~

Day - 3

4. With a Process transition diagram, explain different Process States.

→ A process is not always ready to run. Occasionally, it has to wait for events from external sources beyond its control - for keyboard input in a text editor. For example, until the event occurs, the process cannot run.

The scheduler must know the status of every process in the system when switching between tasks, it obviously doesn't make sense to assign CPU time to processes that have nothing to do.

For example, if a process is waiting for data from a peripheral device, it is the responsibility of the scheduler to change the state of the process from waiting to runnable once the data have arrived.

A process may be in one of the following states -

① New:- The process is newly created and is in a transition state; the process exists, but is not ready to run nor is it sleeping. This state is the start state for all processes except process 0.

② Ready (in memory):- Process is not executing but is ready to run as soon as the kernel schedules it.

③ Ready (Swapped):- Process is ready to run but the Swapper must swap the process into main memory before the kernel can schedule it to execute.

④ Sleeping (in memory) :- Process is sleeping and resides in main memory.

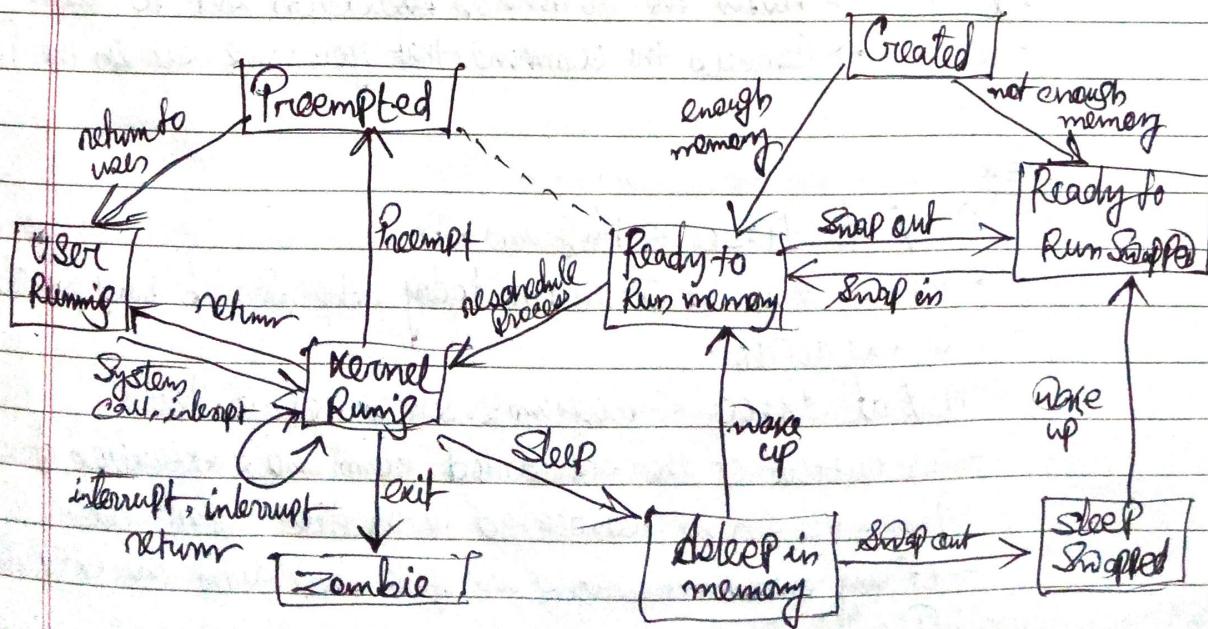
⑤ Sleeping (Snapshot) :- The process is sleeping, and the snapshot has swapped the process to secondary storage to make room for other processes in main memory.

⑥ Running (User) :- The process is executing in user mode.

⑦ Running (Kernel) :- The process is executing in kernel mode.

⑧ Preempted :- The process is returning from the kernel to user mode but the kernel preempts it and does a context switch to schedule another process.

⑨ Terminated :- The process executed the exit system call and is in the zombie state. The process no longer exists but leaves a record containing an exit code and some time statistics for its parent process to collect.



Process State Transitions Diagram

2. Explain Process Scheduling Commands.

- Unix Provides Special facilities to Schedule a Job to be at a Specified time of day, rather than executing now.
This is very useful because the load on any Unix System varies greatly through out the day. So, it makes sense to schedule less urgent jobs at a time when the system overhead are low.

Unix Provides three commands for scheduling execution of process later:-

(a) at Command - Delaying of Processing.

(b) batch command.

(c) Cron Daemon - To execute Jobs Periodically

(a) at [-option] Time :

- at Command is used to execute a specified command at a later date and time.

Option	Purpose
-f filename	→ Specifies the file that contains the command to run.
-m	→ Sends mail once the command has been run.
-l	→ Lists the commands ^{have} been set to run.
-r	→ Cancels the command that you have set in the past.

Ex:

i) \$ at 17:30 < Program.sh

- means that at 5:30 PM today, the script Program.sh will be executed.

ii) \$ at 18:00 < Program2.sh > report.lst

- The output of the scheduled command, execution of Program2.sh is redirected to the file report.lst.

- if any error messages are generated during the execution of Program

iii) at -r 87 4130881-a

- The job which is specified after -r is deleted from the at queue.

~~#~~(b) batch Command:

- The batch command too schedules jobs later execution, but unlike at, scheduling is done in such a way that jobs are executed as soon as the system load permits, in other words, when the load average drops below 1.5, or the value specified in the invocation of atrun.
- When we submit our jobs using this command, Unix executes our ~~at~~ job when it is relatively free and the system load is light.

Eg: \$batch sort emp.date | grep bangalore > add.out

- The file emp.date is sorted (sort), searches for the string "bangalore" (grep) and redirects the output of the file add.out.
- The command is not executed immediately, but gets executed when the processes load is relatively lesser.

② Cron Daemon - To execute jobs Periodically

- Cron is a Periodic command scheduler. It enables users to schedule jobs (commands or shell scripts) to run periodically at certain times or dates.

- It is commonly used to automate systems maintenance or administration.

- It executes commands automatically without prompting from the user.

- Unlike at and batch that are meant for one-time execution, cron executes programs at regular intervals.

① Setting up a Crontab File:

A Crontab file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five are integers that specify the following:

(i) minute (0-59)

(ii) hour (0-23)

(iii) day of the month (1-31)

(iv) month of the year (1-12)

(v) Day of the week (0-6 with 0=Sunday)

(vi) The sixth field of a line in a crontab file is a string to be executed by the shell at the specified times by the first five field.

② Crontab command:

To list and edit Crontab file we should use Crontab Command which copies the specified command - file into a directory that holds all users Crontab.

Options

	Purpose
-l	used to view command(s) that has been submitted.
-r	used to remove a job
-e	Edits the crontab file to be used.

→ \$ crontab cron.txt

- the file cron.txt contains Commands that need to be executed periodically at the specified date and time.

→ 0 0 1 15 1 sort employee.lst

(will run the sort command on the first and fifteenth of each month, as well as on every Monday at exactly midnight).

→ 0 23 * * 1 echo "Time to take backup"

(will run a command only on Mondays at eleven pm)

(3)

User Management in Linux. Various operation Perform With user in terms of assigning & Committing. Briefly explain.

Soln →

- A user is an entity, in a Linux operating system, that can manipulate files and perform several other operations.
- Such user is assigned an ID that is unique for each user in the operating system.
- In a single directory, we can create 60,000 users. Now we will discuss the important commands to manage users in Linux.

① To list out all the users in Linux use the awk command with -F option. Here we are accessing a file and printing only first column with help of Print \$1 and awk.

Pi@Picoder:~\$ awk -F ':' '{print \$1}' /etc/passwd

Pi

Pi

② Using id command, we can get the ID of any username.

Every user has an id assigned to it and the user is identified with the help of this id. By default, this id is also the group id of the user.

Pi@Picoder:~\$ id -u

③ The command to add a user, adduser command adds a new user to the directory.

\$ sudo adduser username

④ Using Passwd command to assign a Password to a user. After using this command we have to enter the new password for the user, and then the password gets updated to the new password.