* Interactive keys with top

→ Besides reporting information, top can be utilized interactively for monitoring and controlling Process.

→ while top is running in terminal window, you can enter single-letter command to change it's behavior.

Command	output
h or ?	Display available interactive keys and their function
t	Display or hide summery information (row 2 and 3)
m	Display or hide memory information (row 4 and 5)
l	Show information for each cpu and not just totals
d	change display update interval
A	Sort the Process list by top resource ~~customers~~ consumers.
r	Renicec change priority of) a specific Process
k	Kill the specific Process
f	Enter the top configuration screen
o	Interactively select a new sort order in the Process list.

→ Most of these interactive keys are actually toggles: hitting them a second time reverts to the original display.

→ There are many more interactive options see the man page for top for a comprehensive list.

→ There are a number of alternatives to top with both prettier display and additional capabilities including atop, btop, htop each program has its fans.

* Scheduling Future Process using at.

→ Suppose you need to perform a task on a specific day sometime in the future.

→ However you know you will be away from the machine on that day.

→ How will you perform the task?

→ you can use the at utility program to execute any non-interactive command at a specified time.

* Cron

→ Cron is time-based scheduling utility Program.

→ It can launch routine background jobs at specific times and/or days on ongoing basis.

→ Cron is driven by configuration file called /etc/crontab (cron table) which contains the various shell command that need to run at the Properly scheduled times.

→ Each line of crontab file represents a job, and is composed of so called CRON expression

→ Typing crontab -e will open the crontab editor to edit existing jobs or to create new jobs. Each line of crontab file will contain 6 fields

Field	Description	values
MIN	minutes	0 to 59
HOUR	Hour field	0 to 23
Dom	Day or month	1-31
mon	month field	1-12
Dow	Day or week	0-6 (0 = Sunday)
CMD	Command	Any command to be executed

* Examples

- The entry * * * * * /usr/local/bin/execute/this/script.sh will schedule a job to execute script.sh every minute of every hour of every day of the month. every month and every day in the week.

- The entry 30 08 10 06 * /home/sysadmin/full-backup will schedule a full backup at 8:30 Am, 10-June, irrespective of the day of the week.

* anacron

→ while cron has been used in Unix-like operating system for deade, morden Linux distribution have moved over to newer facility : anacron.

→ This was because cron implicitly assumed the machine was always running.

→ However if the machine was powerd off, sheduled job would not run.

→ anacron will run necessary jobs in a controlled and staggered maner when the the system is up and running.

→ The key configuration file is
/etc/anacrontab

→ Note that anacron still makes use of the cron infrastructure for submitting jobs on a daily, weekly, monthly basis but it defers running them until opportune times when the system is actually alive

* sleep

→ sleep suspends execution for at least the specified period of time, which can be given as number of second (default), minuts, hours, or days.

→ After that time has passed execution will resum

sleep Number [Suffix] ....

→ where suffix may be

- s for second
- m for minutes
- h for hours
- d for days.
-