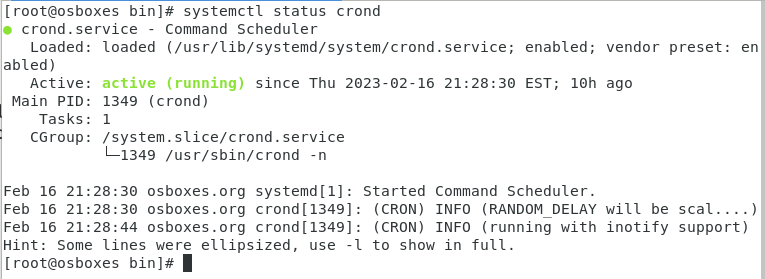
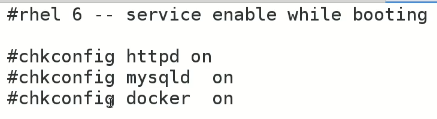
**Lecture 22**

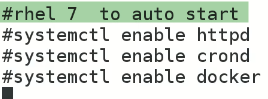
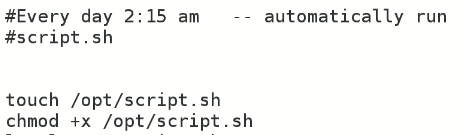
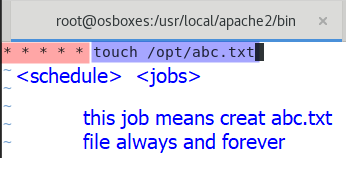
**Cron-Scheduler**

Cron



* Cron jobs
* Works as a service
* $ systemctl status crond 🡪 alternatively $ systemctl is-active crond
* 
* Cron is a time-based job scheduler in Unix-like operating systems. It allows users to schedule commands or scripts to run automatically at specified intervals.
* Cron works by reading a "crontab" (cron table) file, which contains instructions for the scheduling of commands. The crontab file is a simple text file that can be edited using the "crontab" command. Each line in the crontab file specifies a command to be executed and the schedule at which it should be run. The schedule is specified using a set of fields that represent the minutes, hours, days of the month, months, and days of the week.
* For example, the following crontab entry will run the script "/path/to/my/script.sh" every day at 3:30 AM:
* 
* Cron is a powerful tool that can be used for a variety of tasks, such as automatically backing up files, updating software, and sending out automated emails. However, it is important to use it with care and understand the potential security risks involved in granting automatic execution of commands.
* In CentOS 6 / RHEL 6
* 
* 
* It is because services are controlled by “init” 🡪 its PID is always 1
* Now,

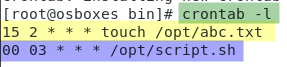
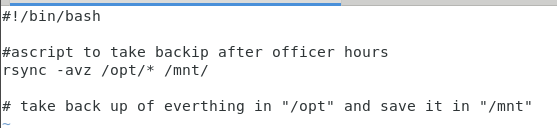
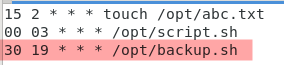
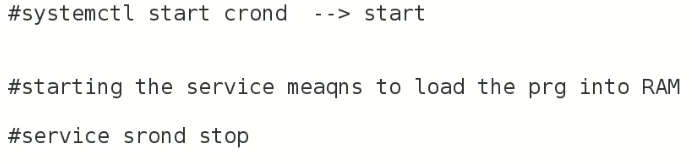
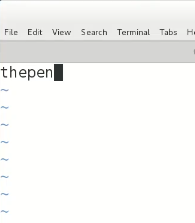
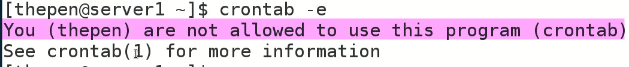
**In RHEL 7/CentOS 7**

* It is changes to “systemd” 🡪 PID 1
* To start “apache” server
* $ systemctl <command> <service\_name>
* $ systemctl start httpd
* 
* 
* How to add cron jobs???
* $ crontab -u <user\_name> -e 🡪 for a particular user 🡪 if logged in as a different use i.e “root”
* $ crontab -e 🡪 open edit to save the jobs schedule.
* it has 5 stars in start 🡪 \* \* \* \* \* 🡪 stars means wildcard or always.
* these stars will be replaced with time a date accordingly.
* 

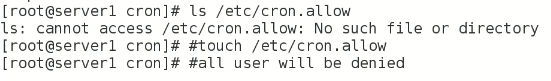
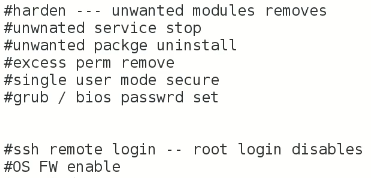
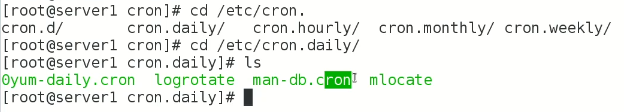
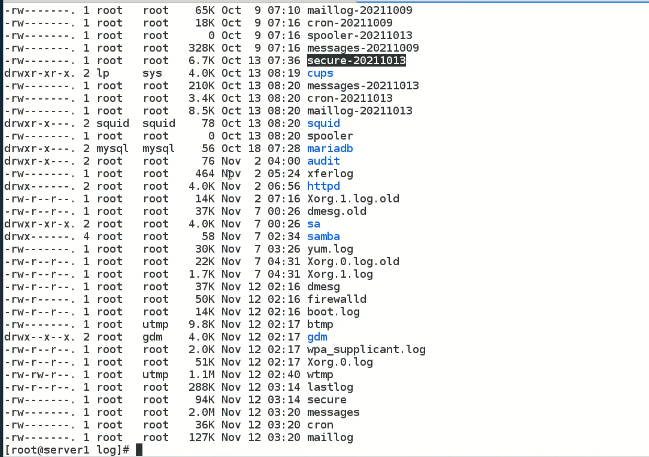
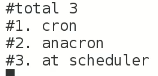
To break down the schedule time, the five asterisks represent the following time intervals:

* The first asterisk represents the minute (0-59).
* The second asterisk represents the hour (0-23).
* The third asterisk represents the day of the month (1-31).
* The fourth asterisk represents the month (1-12).
* The fifth asterisk represents the day of the week (0-6, where Sunday is 0 or 7).

In this case, all five asterisks are set to "\*", which means "any value". Therefore, the cron job will run every minute of every hour, every day of the month, every month, and every day of the week.

* 
* **Note:-** the jobs are scheduled in “vi” editor so save the file as it is saved in “vi or vim editor”
* $ crontab -l 🡪 list the jobs scheduled through cron
* 
* Task
* Make a script which takes backup at 7:00 after office hours
* ”
* Now schedule this job with cron
* 
* To take backup of cron jobs
* /var/spool/cron/root
* $ cp /var/spool/cron/ /mnt/
* How cron works?
* 
* 
* To check if cron scheduled job is done or not?
* $ cat /var/log/cron 🡪 log file keeps the record of all schedules
* How to deny users or any specific user to schedule cron jobs,
* $ ls /etc/cron 🡪 add entry into cron.deny
* 
* $ vi cron.deny
* 
* User “thepen” is entered in cron.deny
* 

There is only cron.deny file

* 
* 
* Operating system (OS) hardening refers to the process of securing an operating system by implementing various measures to reduce its vulnerability to cyber attacks and unauthorized access
* 
* Cron also runs many jobs related to the system.
* 
* System Log rotates by cron
* 
* Another scheduler 🡪 anacron 🡪 /etc/anacron –
* Jobs left by cron are executed by anacron
* 
* “at scheduler” runs for one time job
* 
* Task
* 
* 
* Crontab guru web site to get schedule scripts or jobs scripts