Schedule tasks using at and cron

– Linux can run tasks automatically, and comes with automated tasks utilities: **cron, anacron, at, batch**.  
– cron jobs can run as often as every minute.  
– A scheduled cron job is skipped if the system is down.  
– **anacron** can run a job only once a day.  
– Scheduled jobs are remembered and run the next time that the system is up.  
– **crond** daemon searches multiple files and directories for scheduled jobs:

1. /var/spool/cron/

2. /etc/anacrontab

3. /etc/cron.d

Configuring cron jobs

cron jobs are defined in /etc/crontab.  
The crontab entries are of the form:

Minutes Hours Date Month Day-of-Week command

where:

Minutes = [0 to 59]

Hours = [0 to 23]

Date = [1 to 31]

Month = [1 to 12]

Day-of-Week = [0 to 6] 0=Sunday - 6=Saturday

command = a script file or a shell command.

Other special characters can be used:

- An asterisk (\*) can be used to specify all valid values.

- A hyphen (-) between integers specifies a range of integers.

- A list of values separated by commas (,) specifies a list.

- A forward slash (/) can be used to specify step values.

Other cron Directories and Files

**/etc/cron.d**  
– Contains files with same syntax as the /etc/crontab – accessible by root privileges only  
– Other cron directories in /etc: –

cron.hourly

cron.daily

cron.weekly

cron.monthly

– Scripts in these directories run hourly, daily, weekly, or monthly, depending on the name of the directory.  
– The **/etc/cron.allow** and **/etc/cron.deny** files restrict user access to cron. If neither file exists, only root can use cron.

Crontab utility

– Users other that root can also configure cron using the crontab utility.  
– user defined crontabs are stored in **/var/spool/cron/[username]**.  
– To create or edit a crontab entry :

# crontab -e

– To list the entries in the user defined crontab :

# crontab -l

Configuring anacron jobs

– anacron jobs are defined in **/etc/anacrontab**.  
– Jobs are defined by :

Period in days : frequency of execution in days

Delay in minutes - Minutes to wait before executing the job

job-identifier - A unique name used in logfiles

command : a shell script or command to execute

example anacron file :

SHELL=/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

# the maximal random delay added to the base delay of the jobs

RANDOM\_DELAY=45

# the jobs will be started during the following hours only

START\_HOURS\_RANGE=3-22

#period in days delay in minutes job-identifier command

1 5 cron.daily nice run-parts /etc/cron.daily

7 25 cron.weekly nice run-parts /etc/cron.weekly

@monthly 45 cron.monthly nice run-parts /etc/cron.monthly

at and batch

– at and batch utilities are used for scheduling one-time tasks.  
– the at command executes a task at a specific time.  
– the batch command executes a task when system load average is below 0.8.  
– the atd service must be running to run at or batch jobs  
– at command syntax :

# at time

– The time argument accept multiple formats :

HH:MM

MMDDYY,MM/DD/YY or MM.DD.YY

month-name day year

midnight: At 12:00 AM

teatime: At 4:00 PM

now + time -- here time can be minutes, hours, days or weeks

– batch command syntax :

# batch (**at>** promp is displayed)

– The **/etc/at.allow** and **/etc/at.deny** files restrict user access to at. If neither file exists, only root can use cron.