IT Scripting and Automation

Periodic Processes / Scheduling Tasks

Lecturer: Art Ó Coileáin

Cron: Schedule Commands

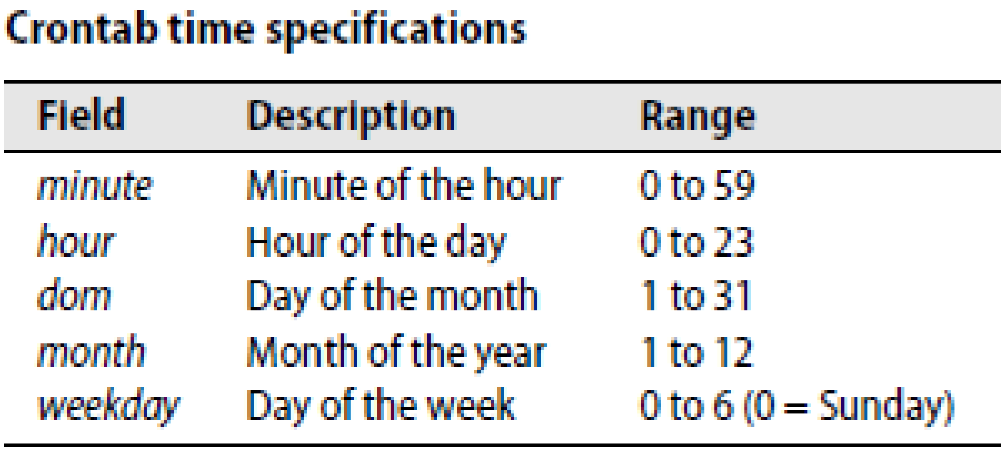
- The ***cron*** name derives from *chronos*, Greek for time.
- The **cron** daemon is the standard tool for running command on a predetermined schedule. It starts when the system boots and run as long as the system is up.
- **cron** reads configuration files that contain lists of command lines and the times at which they are to be invoked. The command lines are executed by **sh**, so almost anything you can do by hand from the shell can also be done with **cron**.
- The **cron** configuration file is called a "crontab", short for "cron table". Crontabs for individual users are stored under **/var/spool/cron**. There is one crontab file per user: one for root, one for benand so on.

Cron: Schedule Commands

- Crontab files are named with the login names of the users to whom they belong, and cron uses these filenames to figure out which UID to use when running the commands contained in each file.
- The **crontab** command transfers files to and from **/var/spool/cron** directory
- The **crontab** helps maintain cron's efficiency by notifying cron when the **crontabs** change.
- Crontab files should **not** be edited directly.
- Cron normally does its work silently, but most versions can keep a log file (usually **/var/cron/log** or **/var/adm/cron/log**). It contains a list of commands that were executed and the rimes at which they ran.

The format of crontabfiles

- The **crontab** on a system share a similar format. Comments are introduced with a # in the first column of a line.
- Each non commented line contains six fields and represents one command:
- ***minute hour dom month weekday command***



Crontab time specifications

Field	Description	Range
minute	Minute of the hour	0 to 59
hour	Hour of the day	0 to 23
dom	Day of the month	1 to 31
month	Month of the year	1 to 12
weekday	Day of the week	0 to 6 (0 = Sunday)

The format of crontab files

- Each of the time-related fields may contain:
  - A star, which matches everything
  - A single integer, which matches exactly
  - Two integers separated by a dash, matching a range of values
  - A range followed by a slash and a step value, e.g. 2-5/3 (Linux only)
  - A comma-separated list of integers or ranges, matching any value

Examples:

45 10 * * 1-5 script_name

- which means “10:45 am, Monday through Friday”. Never put a star in the first field unless the command is wanted to be run every minute

0,30 * 13 * 5 script_name

- Means “every half-hour on Friday, and every half-hour on the 13th of the month”

The format of crontab files

- The command can be any valid shell command and should not be quoted. The command is considered to continue to the end of the line and may contain blanks or tabs.

Examples:

- **30 2 * * 1   (cd /home/ben/project; make)**
- This entry runs make command in the directory /home/ben/project every Monday morning at 2:30 a.m.

- **20 1 * * 0-3,6   find /tmp -atime +3 -type f -exec rm -f {} ';'**
- This line runs at 1:20 each morning (Sun-Wed and Sat) and removes all files in the /tmp directory that have not been accessed in 3 days.

The crontabfiles

What are the meanings of the following crontab entries?

- 0 1 * * 5 /bin/execute/this/script.sh
- 0 1 * * 1-5 /bin/execute/this/script.sh
- 10 * 1 * * /bin/execute/this/script.sh
- 0,10,20,30,40,50 * * * * /bin/execute/this/script.sh

Crontab

- To see what contabs are currently running on your system (as root):

```
$ crontab -l
```

- To edit the list of cronjobs:

```
$ contab -e
```