# Cron Jobs

Reference: <https://www.thegeekdiary.com/centos-rhel-begginners-guide-to-cron/>

Cron is a system daemon used to execute desired tasks (in the background) at designated times.

You can view your current cron table via the command:

* crontab -l

You can remove your current cron table via command:

* crontab -r

You can edit your cron table via the command:

* crontab -e
  + Editor options are displayed the first time your run this command, your choice becomes permanent.

# Interpreting the time and date fields

Each crontab line has either 2 or 3 elements:

1. 5 time and date fields.
2. Username – only seen in the system crontab.
3. A command to be executed when the time specified in the time and date fields matches the current time.

Text

Description automatically generated

If you want to run a job on every day of the week, use the \* in the day of week field.

A picture containing graphical user interface

Description automatically generated

# Examples of setting cron jobs

Here are a few simple examples of crontab usages to understand how to schedule a task:

**Example 1:** Run a job 5 minutes after midnight every day:

5 0 \* \* \* /home/oracle/scan\_asm\_devices.sh

**Example 2:**  Run a job at 5:30pm on the 1st of every month:

30 17 1 \* \* mail -s “It’s 5:30pm”

**Example 3:** Run a job at 4:05am every Monday and Wednesday:

5 4 \* \* Mon,Wed echo “run at 5 after 4 every Monday and Wednesday”

**Example 4:** Run a job every fifteen minutes:

\*/15 \* \* \* \* date >> /tmp/timelog

**Class Example Part 1:**

Write a bash script called log-uptime.sh that, when executed displays the current UTC date and time in the format: 2023/02/15 11:40:26, followed by the output of the “uptime” command (all on one line).

We can use the “date” command to get the current date and time. Because uptime also displays the current time, use the “cut” command so that it’s not displayed.

An example line of output:

2023/02/15 11:40:26 up 42 min, 1 user, load average: 0.04, 0.01, 0.00

Text

Description automatically generated

**Class Example Part 2:**

I would like to execute this command every hour, at the top of the hour. That way, I will have a record of how long my system has been up, and when it has been shutdown. Unfortunately, I’ll have to remember to do this every hour. Perhaps I could a reminder on my phone for 5 minutes to every hour, then I can log into the system and run the command. That will be troublesome overnight, because it’s hard to function if you only get one hour of sleep at a time. (Try having kids!)

Cronjobs to the rescue! With cron, you can schedule commands (including your own programs, scripts, like the bash script we just created) to execute periodically.

The cron daemon looks at the “crontabs” directory **/var/spool/cron/crontabs**; this lists **user** jobs to be run, and when they are to be run.

* YOU DO NOT EDIT THE FILE DIRECTLY! Rather you use the crontab command (man 1 crontab) – it looks after ensuring everything stays in sync with the cron daemon.

**Review:** See “man 5 crontab” for the format. Each line of the crontab file has 6 fields, 5 for the time/date and one for the command to be executed.

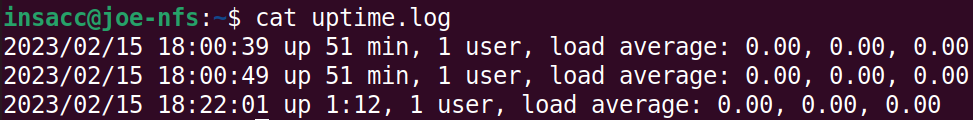
A picture containing graphical user interface

Description automatically generated

**Note:**  You must use absolute paths in crontabs. (Not even the ~ will work!)

Text

Description automatically generated



To view the crontab:

Text

Description automatically generated

On save, cron will parse your entries and let you know anything regarding syntax errors.

* The crontab -e command copies your crontab to a temp file, when you are done editing it makes sure it’s valid.
  + If it is valid it will put the changes into the actual crontab file.
  + If it isn’t valid you will get a explanatory message telling you what to fix.
    - If you choose not to fix it, no changes will be made upon exit.

Try breaking it!

Text

Description automatically generated

Note: The **/etc/crontab** file is usually never edited, instead there’s a directory **/etc/cron.d** where files can be created in the system crontab format with the “user” field included.

\*\*Extra bits:

* If you want to learn more about something related to cron look up anacron!