

# CRON

Saturday, February 24, 2018 2:34 PM

Cron is a tool for configuring scheduled tasks on Unix systems. It is used to schedule commands or scripts to run periodically and at fixed intervals. Tasks range from backing up the user's home folders every day at midnight, to logging CPU information every hour.

The command `crontab` (cron table) is used to edit the list of scheduled tasks and is done on a per-user basis; each user (including root) has their own crontab.

1. Cron GUI, a graphical application for Cron is available by installing the `gnome-schedule` package:
  - a. Open a **Terminal Window** from the **Main Menu**.
  - b. Enter: **`sudo apt-get install gnome-schedule`** You can then launch the program Scheduled Tasks from the main menu.
2. Editing crontab: Run `crontab` with the `-e` flag to edit the cron table by entering **`crontab -e`**. The first time you run `crontab` you'll be prompted to select an editor; if you are not sure which one to use, choose nano by pressing **Enter**.
3. Add a scheduled task: The layout for a cron entry is made up of six components: minute, hour, day of month, month of year, day of week, and the command to be executed.

```
# m h dom mon dow  command
# * * * * *  command to execute

# T T T T T
# | | | | |
# | | | | |
# | | | | |
# | | | | |  └─ day of week (0-7) (0-6 = Sun-Sat, or use names; 7/0 is Sunday)
# | | | | |  └─ month (1 - 12)
# | | | | |  └─ day of month (1 - 31)
# | | | | |  └─ hour (0 - 23)
# | | | | |  └─ min (0 - 59)
Example: 0 0 * * * /home/pi/backup.sh (run backup.sh every day at midnight)
```

4. View currently saved scheduled tasks: Enter **`crontab -l`**
5. Run a task on reboot
  - a. To run a command every time the Raspberry Pi starts up, write `@reboot` instead of the time and date. For example: **`@reboot python /home/pi/myscript.py`**. This will run the Python script every time the Raspberry Pi reboots.
  - b. If you want the command to be run in the background while the RPi continues starting up, add a space and `&` at the end of the line: **`@reboot python /home/pi/myscript.py &`**.
6. Here's a quote from the manual CRON(8) : cron searches its spool area (`/var/spool/cron/crontabs`) for crontab files (which are named after accounts in `/etc/passwd`). crontabs found are loaded into memory. Note that crontabs in this directory should not be accessed directly - the `crontab` command should be used to access and update them.

Debian Linux **does not activate cron logging automatically**. To activate logging:

```
sudo nano /etc/rsyslog.conf
Find the line # cron and remove the comment (#)
Then restart rsyslog via
/etc/init.d/rsyslog restart
```

The cron logs will be in `/var/log/cron.log`