How to Schedule a Task or Job on Linux Using Crontab

Job scheduling is a feature that is available in all operating systems that allow a user to perform specific tasks at a specific time interval.

The Linux crontab is similar to windows task scheduler.

This reduces the Linux administrator work and regularly executes scheduled jobs at a given time if the system is running.

There are three tools in Linux for planning a job.

All three tools have unique features and choose the one that suits your needs.

* **cron:** cron is used to execute a task repeatedly at a specified time. It can be monthly, weekly, daily, hourly, every minute, even every second.
* **at:** The at command is used to run a task one-time (not recurring) at a specific time.
* **anacron:** anacron command is used to execute commands periodically, which is similar to cron. But this tool is specifically designed for a system that is not running 24×7.

In this tutorial, we are going to teach you about the cron command and then in the upcoming articles we will explain about at and anacron command.

cron uses a 24-hour format, so use the values accordingly.

#### **What is cron Command**

The cron tool allows Linux users to execute commands or scripts at a specific date and time. It is very useful to perform regular tasks such as regular backups, daily scans, /tmp directory cleanup, and restarting the system at a given time.

cron executes scheduled jobs automatically in the backend at a specific time. Each user can have their own ctrontab, and this can be found it in “/var/spool”.

cron jobs can be allowed or disallowed for any users by adding a user in the “cron.allow” and “cron.deny” file.

#### **Linux Crontab Syntax**

The Linux crontab has six fields, the first 5 fields (1-5) indicate the date and time of execution, and the 6’th field is used to execute a command or script.

.-----------------> Minute (0-59)

| .--------------> Hour (0-23)

| | .-----------> Day of Month (1-31)

| | | .--------> Month (1-12) or (jan,feb,mar,apr ...dec)

| | | | .-----> Day of Week (0-6) (Sunday=0 or 7) or (sun,mon,tue,wed,thu,fri,sat)

| | | | | .--> Command to be executed

| | | | | |

\* \* \* \* \* \*

MIN HOUR DOM MON DOW CMD

#### **Important cron Commands**

Each user should be familiar with the following important cron commands that helps the user to manage the cron easily.

Use the following command to add/edit the current user’s crontab entry. This will open crontab file in the editor and allow users to add or remove an entry in it.

# crontab -e

Use the following command to list the logged-in user’s crontab entry.

# crontab -l

Use the following command to remove the logged-in user’s crontab entry.

# crontab -r

Use the following command to add/edit crontab entry to a specific user.

# crontab -u username -e

Use the following command to list a particular user’s crontab entry.

# crontab -u username -l

Use the following command to delete a particular user’s crontab entry.

# crontab -u username -r

#### **Important cron Files and Location**

Below are the important files of the cron daemon.

The crontab configuration file.

/etc/crontab

All user’s crontab files can be found in the below location.

# ls -lh /var/spool/

Predefine crontab directory.

/etc/cron.d

/etc/cron.daily

/etc/cron.hourly

/etc/cron.monthly

/etc/cron.weekly

/etc/cron.deny

Allowed crontab special characters.

+-------+----------------------------------------------------+

| Value | Description |

+-------+----------------------------------------------------+

| \* | Match all possible values in the field |

| - | To define a range |

| / | To repeat an event at a specific interval |

| , | To separate a items |

+-------+----------------------------------------------------+

#### **How to Install cron on Linux**

By default, the crony package is installed on most Linux system. If not, it can be easily installed with the help of the distribution package manager.

For **Fedora** system, use [**DNF Command**](https://www.2daygeek.com/dnf-command-examples-manage-packages-fedora-system/) to install cronie.

$ sudo dnf install cronie

For **Debian/Ubuntu** systems, use [**APT-GET Command**](https://www.2daygeek.com/apt-get-apt-cache-command-examples-manage-packages-debian-ubuntu-systems/) or [**APT Command**](https://www.2daygeek.com/apt-command-examples-manage-packages-debian-ubuntu-systems/) to install cronie.

$ sudo apt install cronie

For **Arch Linux** based systems, use [**Pacman Command**](https://www.2daygeek.com/pacman-command-examples-manage-packages-arch-linux-system/) to install cronie.

$ sudo pacman -S cronie

For **RHEL/CentOS** systems, use [**YUM Command**](https://www.2daygeek.com/yum-command-examples-manage-packages-rhel-centos-systems/) to install cronie.

$ sudo yum install cronie

For **openSUSE Leap** system, use **[Zypper Command](https://www.2daygeek.com/zypper-command-examples-manage-packages-opensuse-system/" \t "_blank)** to install cronie.

$ sudo zypper install cronie

Once you have installed the cronie package, run the following **[systemd command or SysVinit command](https://www.2daygeek.com/sysvinit-vs-systemd-cheatsheet-systemctl-command-usage/" \t "_blank)** to start and enable it.

Use the following command to start the crond Service on SysVinit systems.

# service crond start

or

# /etc/init.d/crond start

Use the following command to enable the crond service on SysVinit systems.

# chkconfig crond on

Use the below command to start the crond service on systemd systems.

# systemctl start crond.service

or

# systemctl start crond

Use the below command to enable the crond service on systemd systems.

# systemctl enable crond.service

or

# systemctl enable crond

#### **1) How to Schedule a cron to Execute a Job Once in a Day**

This example runs the scheduled shell script of the MySQL backup at 6PM every day.

0 18 \* \* \* /opt/scripts/mysql\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 18 | It runs the job at 6PM |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **2) How to Schedule a cron to Execute a Job Twice in a Day**

This example runs the scheduled shell script of the MySQL backup at 9AM and 3PM twice a day.

0 9,15 \* \* \* /opt/scripts/mysql\_backup.sh

**Details:**

+-------+--------------------------------+

| Field | Description |

+-------+--------------------------------+

| 0 | 0'th Minute |

| 9,15 | It runs the job at 9AM and 3PM |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+--------------------------------+

#### **3) How to Schedule a cron to Execute a Job Between Time in a Day**

This example runs the scheduled shell script of the MySQL backup from 10AM to 2PM. It runs 5 times a day.

0 10-14 \* \* \* /opt/scripts/mysql\_backup.sh

**Details:**

+-------+-----------------------------------------------+

| Field | Description |

+-------+-----------------------------------------------+

| 0 | 0'th Minute |

| 10-14 | It runs the job at 10AM,11AM,12PM,1PM and 2PM |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-----------------------------------------------+

#### **4) How to Schedule a cron to Execute a Job Every Minute**

This example runs the scheduled shell script of clear-tmp.sh every minute.

\* \* \* \* \* /opt/scripts/clear-tmp.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| \* | Every Minute |

| \* | Every Hour |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **5) How to Schedule a cron to Execute a Job Every Five Minutes**

This example runs the scheduled shell script of the “clear-tmp.sh” every 5 minutes.

\*/5 \* \* \* \* /opt/scripts/clear-tmp.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| \*/5 | Every 5 Minute |

| \* | Every Hour |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **6) How to Schedule a cron to Execute a Job Every 2 Hours**

This example runs the scheduled shell script of the “clear-tmp.sh” every two hours.

\* \*/2 \* \* \* /opt/scripts/clear-tmp.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| \* | Every Minute |

| \*/2 | Every 2 Hours |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **7) How to Schedule a cron to Execute a Job Once in a Week**

This example runs the scheduled shell script of the website\_backup.sh once a week at 5AM.

0 5 \* \* 6 /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 5 | It runs the job at 5AM |

| \* | Every day |

| \* | Every month |

| 6 | Every week of Saturday |

+-------+-------------------------+

#### **8) How to Schedule a cron to Execute a Job Twice a Week**

This example runs the scheduled shell script of website\_backup.sh every Sunday at 8PM and 10PM.

0 20,22 \* \* sun /opt/scripts/website\_backup.sh

**Details:**

+-------+---------------------------------+

| Field | Description |

+-------+---------------------------------+

| 0 | 0'th Minute |

| 20,22 | It runs the job at 8PM and 10PM |

| \* | Every day |

| \* | Every month |

| sun | Every week of Sunday |

+-------+---------------------------------+

#### **9) How to Schedule a cron to Execute a Job on Specified Days**

This example runs the scheduled shell script of website\_backup.sh every Tuesday and Friday at 3AM.

0 3 \* \* tue,fri /opt/scripts/website\_backup.sh

**Details:**

+-------+------------------------------------------+

| Field | Description |

+-------+------------------------------------------+

| 0 | 0'th Minute |

| 3 | It runs the job at 3AM |

| \* | Every day |

| \* | Every month |

|tue,fri| It runs the job every Tuesday and Friday |

+-------+------------------------------------------+

#### **10) How to Schedule a cron to Execute a Job on Specified Months**

This example runs the scheduled shell script of website\_backup.sh on March, July & December at 11PM.

0 23 \* mar,jul,dec \* /opt/scripts/website\_backup.sh

**Details:**

+-------------+------------------------------------------+

| Field | Description |

+-------------+------------------------------------------+

| 0 | 0'th Minute |

| 23 | It runs the job at 11PM |

| \* | Every day |

| mar,ju,dec | It runs the job March, July & December |

| \* | Every day of the week |

+-------------+------------------------------------------+

#### **11) How to Schedule Multiple Tasks in One cron**

This example runs the programmed shell script of website\_backup.sh & clear-tmp.sh every five hours.

0 \*/5 \* \* \* /opt/scripts/website\_backup.sh; /opt/scripts/clear-tmp.sh

or

0 \*/5 \* \* \* /opt/scripts/website\_backup.sh && /opt/scripts/clear-tmp.sh

**Details:**

+-------+------------------------------------------+

| Field | Description |

+-------+------------------------------------------+

| 0 | 0'th Minute |

| \*/5 | It runs Every 5 Hours |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+------------------------------------------+

#### **12) How to Use a cron Special String in Linux**

You can use @string instead of a long command like (\*\*\*\*\*). It is very simple and straight forward. See the following examples for a better understanding.

|  |  |  |
| --- | --- | --- |
| **String** | **Value** | **Description** |
| @reboot | — | Runs at boot |
| @hourly | 0 \* \* \* \* | Runs once an hour |
| @daily | 0 0 \* \* \* | Runs once a day |
| @midnight | 0 0 \* \* \* | Runs once a day |
| @weekly | 0 0 \* \* 0 | Runs once a week |
| @monthly | 0 0 1 \* \* | Runs once a month |
| @yearly | 0 0 1 1 \* | Runs once a year |
| @annually | 0 0 1 1 \* | Runs once a year |

#### **13) How to Schedule a cron Using the @hourly String**

@hourly string is similar to (0 \* \* \* \*). This example runs the programmed shell script of website\_backup.sh in the first minute of each hour.

@hourly /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| \* | Every Hours |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **14) How to Schedule a cron Using the @daily String**

@daily string is similar to (0 0 \* \* \*). This example runs the scheduled shell script of website\_backup.sh every day for the first minute and for the first hour.

@daily /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | 1'st Hour |

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **15) How to Schedule a cron Using the @midnight String**

@midnight string is similar to (0 0 \* \* \*). This example runs website\_backup.sh’s scheduled shell scripts every day for the first minute at 12 midnight.

@midnight /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | It runs 12AM at midnight|

| \* | Every day |

| \* | Every month |

| \* | Every day of the week |

+-------+-------------------------+

#### **16) How to Schedule a cron Using the @weekly String**

@weekly string is similar to (0 0 \* \* 0). This example will execute the scheduled work of the website\_backup.sh shell script in the first minute of each week and the first hour of Sunday.

@weekly /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | 1'st hour |

| \* | Every Day |

| \* | Every Month |

| 0 | Every Sunday |

+-------+-------------------------+

#### **17) How to Schedule a cron Using the @monthly String**

@monthly string is similar to (0 0 1 \* \*). This example will run the scheduled work of the website\_backup.sh shell script in the first minute and first hour of each month.

@monthly /opt/scripts/website\_backup.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | 1'st Hour |

| 1 | 1'st day of month |

| \* | Every Month |

| \* | Every day of the week |

+-------+-------------------------+

#### **18) How to Schedule a cron Using the @yearly String**

@yearly string is similar to (0 0 1 1 \*). This example will execute the scheduled job of the yearly-maintenance.sh shell script in the first minute, first hour, 1st day of the month, and first month (January) of every year.

@yearly /opt/scripts/yearly-maintenance.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | 1'st Hour |

| 1 | 1'st day of month |

| 1 | Once in a year |

| \* | Every day of the week |

+-------+-------------------------+

#### **19) How to Schedule a cron Using the @annually String**

@annually string is similar to (0 0 1 1 \*). This is similar to the example above and does the same thing, but only the name is different.

This example will execute the scheduled job of the yearly-maintenance.sh shell script in the first minute, first hour, 1st day of the month, and first month (January) of every year.

@annually /opt/scripts/yearly-maintenance.sh

**Details:**

+-------+-------------------------+

| Field | Description |

+-------+-------------------------+

| 0 | 0'th Minute |

| 0 | 1'st Hour |

| 1 | 1'st day of month |

| 1 | Once in a year |

| \* | Every day of the week |

+-------+-------------------------+

#### **20) How to Run a cron job Automatically After Server Reboot**

@reboot is a special string and that allows the user to execute any command or script at startup (boot time).

This example executes the scheduled job of the system-info.sh file after a server restart (start or boot time).

@reboot /opt/scripts/system-info.sh

#### **21) How to Disable a cron eMail Notifications in Linux**

By default a cron will send an email alert when the scheduled job is completed. If you want to disable it for a specific task. To do so, add the below line to the end of the command or script.

/dev/null 2>&1

\*/5 \* \* \* \* /opt/scripts/clear-tmp.sh >/dev/null 2>&1