Crontab - Scheduled Tasks

From AudioWiki

Crontab is a Linux tool which allows you schedule a task. You don't need to be in your computer in a specific moment to to execute a script or whatever.

Contents

Quick use instructions

To schedule a task you must execute next command:

```
$ crontab -e
```

And it will opened crontab file. Its format is:

```
# m h  dom mon dow   command
```

Where:

- m: point to minute (from 0 to 59) that the command will be executed.
- h: point to hour (in 24 hour format) that the command will be executed.
- dom: point to day of the month that the command will be executed.
- mon: point to month of the year that the command will be executed.
- dow: point to day of the week that the command will be executed. It will be indicated with a number o first three letters of the day:
  - **0** (and in some cases **7**) or **sun** means Sunday.
  - **1** or **mon** means Monday.
  - **2** or **tue** means Tuesday.
  - **3** or **wed** means Wednesday.
  - **4** or **thu** means Thursday.
  - **5** or **fri** means Friday.
  - **6** or **sat** means Saturday.
- command: command to execute.

An example of a crontab file line is:

```
05 7 20 11 4 cp -Rp /home/user1/scripts/ /home/user1/tasks
```

This command will copy **/home/user1/scripts/** folder into **/home/user1/tasks/** folder on November 20th at 7:05 a.m. even if the day of the week isn't Thursday.

Or when crontab file was opened you can find next also format:

```
# m h dom mon dow user command
```

In this case you must indicate the user's name who want to execute the task. An example is:

```
05 7 20 11 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

Wildcards

The symbol ***** means that always the command will be executed. Examples:

```
* 7 20 11 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

The command will be executed on November 20th at 7:00 a.m., 7:01 a.m., 7:02 a.m. and so on until 7:59 a.m. even if the day of the week isn't Thursday.

```
05 * 20 11 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

The command will be executed on November 20th at 0:05 a.m., 1:05 a.m., 2:05 a.m. and so on until 23:05 a.m. even if the day of the week isn't Thursday.

```
* * 20 11 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

The command will be executed on November 20th at every minute of the day even if the day of the week isn't Thursday.

```
05 7 * * 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

This command will be executed every Thursday of the year at 7:05 a.m.

```
05 7 20 11 * user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

This command will be executed November 20th at 7:05 a.m.

Changing default editor

By default, when you execute **crontab -e** command it could be modified with nano editor, but if you prefer use another editor, **vim** for example, you can use this instruction before **crontab -e** command:

```
export EDITOR=vim
```

and crontab file will be opened with **vim**.

Important

About preferences execution time in commands

Once set a day and a month don't matter if the day of the week is correct or wrong, the command will be executed. For example:

```
05 7 20 11 4 cp -Rp /home/user1/scripts/ /home/user1/tasks
```

This command will be executed on November 20th at 7:05 a.m. even if the day of the week isn't Thursday. It isn't matter the day of the week, it wolud be executed per force on November 20th.

In the other hand, if you need to execute a command weekly in a specific day one solution would be:

```
05 7 * 11 4 user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

Don't set a day, because the day of the month you put, for example 20, isn't Thursday, command won't be execute. In this case command will be executed every Thursday of November. Another possibility is:

```
05 7 20 * * user cp -Rp /home/user1/scripts/ /home/user1/tasks
```

That will be executed every Thursday of the year.

About the user who is responsible of scheduled task

There is a crontab file for every user. That means if you execute **crontab -e** command as user user_name, scheduled tasks will be run with user_name's privileges. If you execute the command as root, scheduled task will be executed with root privileges. So, if you have in crontab file next line:

```
30 19 30 10 * mv /etc/hosts.allow .
```

data will be moved to the user's home folder (/home/user_name) on October 30th at 19:30.