

# [Fondamentaux] Cron

## Introduction

Cron est un **planificateur de tâche** disponible sur les distributions Linux.

Il peut permettre l'exécution de binaire ou de script notamment pour vos sauvegardes ou vos programmes.



## Configuration

Cron se configure depuis le fichier **/etc/crontab** où chaque ligne est une tâche planifiée. Voici un pense bête explicatif :

# Cron reference sheet for sysadmins

Min Hour Day Mon Weekday

\* \* \* \* \* script/command to be executed



Day of Week (0=Sun..6=Sat)  
Month (1..12)  
Day of Month (1..31)  
Hour (0..23)  
Minute (0..59)

**Minutes** 0..59 the command/script would be executed at the specified minute.

**Hours** 0..23 the command/script would be executed at the specified hour.

**Days** 1..31 the days of the months in which the script or command would be executed.

**Months** 1..12 the month in which the script would be executed.

**Weekdays** 0..6 the days of the week in which the script gets executed. 0 is Sunday.

## Format

**Examples**

0 \* \* \* \* /opt/backup.sh perform a system backup every hour.

\* / 7 \* \* \* \* /opt/ping.sh check if the remote server is online every 7 minutes.

0 \* / 6 \* \* \* \* /opt/emptytrash.sh empty trash every 6 hours.

20 14 \* \* \* /opt/upgrade upgrade the system at 14:20 PM of every day.

5 9 \* 4 \* /opt/upgrade upgrade the system at 09:05 AM in April.

20 14 \* \* ? /opt/update.sh update system At 14:20 PM of every day.

6 11 \* \* 3 /opt/upgrade.sh upgrade the system at 11:06 AM of every Wednesday.

0 22 \* \* 1-5 /opt/upgrade.sh upgrade the system at 22:00 PM on every day-of-week from Monday through Friday.

0 0 \* \* 2 /opt/upgrade.sh upgrade the system at midnight (00:00) of ever Tuesday.

10 8 \* \* 4L /opt/monitor.sh monitor the system at 08:10 AM on the last Thursday of every month.

15 0 \* \* 4#2 /opt/upgrade upgrade the system at at 00:15 AM on the second Thursday of every month.

0 0 0 1 \* \* /opt/backup.sh perform a sys backup every 1st of month (monthly).

0 0 0 1 1 \* /opt/backup.sh perform a sys backup every 1st of january (yearly).

5 12 \* \* 6 /opt/emptytrash.sh clears the trash at 12:05 PM on Sunday.

@reboot /opt/backup.sh perform a system backup at reboot.

## Special strings

**@reboot** command will be executed once at system startup (non-standard).

**@hourly** command will be executed once an hour, same as ("0 \* \* \* \*") but non-standard.

**@daily** command will be executed once each day, same as ("0 0 \* \* \*") but non-standard.

**@midnight** same as @daily but also non-standard.

**@weekly** command will be executed once every week, same as ("0 0 \* \* 0") but non-standard.

**@monthly** command will be executed once every month, same as ("0 0 1 \* \*") but non-standard.

**@yearly** command will be executed once every year, same as ("0 0 1 1 \*") but non-standard.



**Crontab**

crontab -e Edit or create a crontab file if doesn't already exist.

crontab -l Display the crontab file.

crontab -r Remove the crontab file.

crontab -u username -l Display another user's crontab file.

crontab -u username -e Edit another user's crontab file.

crontab -v Display the last time you edited your crontab file.

## Special characters

**Asterik (\*)** this operator is used to represent all potential values in a field. Write an asterisk "\*" in the Minute column, for example, if you want your cron job to execute every minute.

**Hyphen (-)** to determine a range of values, use this operator. For example, if you want to set up a cron job from Monday through Friday, simply write 1-5 in the weekday column.

**Slash (/)** to split a value, use this operator. For instance, if you want a script to run every 6 hours, enter \*/6 in the Hour field.

**Comma (,)** to list numerous values, use this operator. Writing 1,5 in the Day of the week field, for example, will schedule the task to be executed every Monday and Friday.

**Last (L)** this operator can be used in the month and weekday fields. Writing 6L in the day-of-week field, for example, signifies the last Saturday of the month.

**Weekday (W)** this operator is used to get the closest weekday from a given time. If the 1st of the month is a Saturday, for example, entering 1W in the day-of-month field will execute the command on the following Monday (the 3rd).

**Hash (#)** it is only permitted for the Day Of Week field, which must be followed by a number between 1 and 5. For instance, 5#2 denotes "the second Friday" of a given month.

**Question mark (?)** can be used instead of "\*" in the Day of Month and Day of Week fields. Use this operator to enter "no specified value" for the "day of the month" and "day of the week" fields.

sysxplore.com

Revision #2

Created 7 October 2023 19:54:56 by Elieroc

Updated 10 October 2023 08:47:20 by Elieroc