

TP15 : la planification des tâches

Table des matières

1. Programmer des tâches périodiques : commande crontab.....	2
--------------------------------------------------------------	---

1. Programmer des tâches périodiques : commande crontab.

Nous effectuons un apt-get update avant de faire un apt-get install ntpdate.

```
root@DS1: ~#apt-get update
Réception de :1 http://security.debian.org/debian-security bookworm-security InRelease [47,9 kB]
Réception de :2 http://security.debian.org/debian-security bookworm-security/main Sources [134 kB]
Réception de :3 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [241 kB]
Atteint :4 http://deb.debian.org/debian bookworm InRelease
Réception de :5 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
479 ko réceptionnés en 7s (73,2 ko/s)
Lecture des listes de paquets... Fait
root@DS1: ~#
```

```
root@DS1: ~#apt-get install ntpdate
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ntpsec-ntpdate ntpsec-ntpdig python3-ntp
Les NOUVEAUX paquets suivants seront installés :
  ntpdate ntpsec-ntpdate ntpsec-ntpdig python3-ntp
0 mis à jour, 4 nouvellement installés, 0 à enlever et 50 non mis à jour.
Il est nécessaire de prendre 172 ko dans les archives.
Après cette opération, 599 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```

Nous exécutons la commande ntpdate de synchronisation horaire suivie du serveur de temps public choisi.

```
root@DS1: ~#ntpdate ntp.unice.fr
2025-01-17 15:48:26.945032 (+0100) -1.900896 +/- 0.156003 ntp.unice.fr 134.59.1.5 s2 no-leap
CLOCK: time stepped by -1.900896
root@DS1: ~#
```

Nous programmons une mise à jour périodique de l'horloge système via une crontab avec la commande nano.

```
root@DS1: ~#crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]:
```

```

GNU nano 7.2 /tmp/crontab.51k27o/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
#mise à jour toutes les 3 minutes de l'horloge système
*/3 * * * * /usr/sbin/ntpdate -s ntp.unice.fr

```

Nous lançons une capture de trames en spécifiant le port ntp pour constater les échanges avec le serveur de temps.

```

root@DS1: ~#tcpdump port ntp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:00:01.509785 IP 172.17.101.205.37624 > ntp.unice.fr.ntp: NTPv4, Client, length 48
16:00:01.557990 IP ntp.unice.fr.ntp > 172.17.101.205.37624: NTPv4, Server, length 48
_

```

Nous supprimons la crontab.

```

root@DS1: ~#crontab -r
root@DS1: ~#

```

Nous nous connectons sous le compte noah et vérifions s'il possède une crontab.

```

root@DS1: ~#su - noah
noah@DS1:~$ crontab -l
no crontab for noah
noah@DS1:~$ _

```

Nous créons, sous ce compte, une crontab qui accomplit les tâches suivantes :

- Ecrire la date toutes les minutes à la fin du fichier /tmp/date.log ;
- Ecrire la liste des processus, tous les quarts d'heure de 9h à 16h45 du lundi au vendredi, à la fin du fichier /tmp/process.log

```
noah@DS1:~$ crontab -e
no crontab for noah - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: _
```

```
GNU nano 7.2 /tmp/crontab.aCND0E/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * date >> /tmp/date.log 2>&1
0,15,30,45 9-16 * * 1-5 ps -ef >> /tmp/process.log 2>&1_
```

Nous visualisons le crontab avec la commande crontab -l.

```
noah@DS1:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * date >> /tmp/date.log 2>&1
0,15,30,45 9-16 * * 1-5 ps -ef >> /tmp/process.log 2>&1
noah@DS1:~$
```

NICOLAU Noah

Nous visualisons l'effet de la première commande.

```
noah@DS1:~$ tail -f /tmp/date.log
ven. 17 janv. 2025 16:14:01 CET
ven. 17 janv. 2025 16:15:01 CET
```

Est-ce que le service crontab est actif ?

```
root@DS1: ~# ps -e | grep cron
  511 ?        00:00:00 cron
  481 ?        00:00:00 cron
root@DS1: ~#
```

Le service est actif pour le 511, mais pas pour le 481.

Nous supprimons le crontab de noah et vérifions qu'il soit supprimé.

```
root@DS1: ~# crontab -u noah -r
root@DS1: ~# crontab -u noah -l
no crontab for noah
root@DS1: ~# _
```