

Niveau : Admin Junior	Numéro de la Leçon	Dernière Modification
2/4	<progrecss 5/12 style=inline />	2020/01/30 03:28

Gestion des Tâches

cron

Le service crond est normalement lancé au démarrage de la machine. Ce service est chargé de faire exécuter des tâches (commandes et scripts) à des moments précis. Le service crond présume que la machine est allumée en permanence.

Le service crond lit toutes les minutes le fichier **/etc/crontab**, les crontabs se trouvant dans **/etc/cron.d** ainsi que les fichiers propres à chaque utilisateur.

Les crontabs des utilisateurs ont comme nom, le nom de l'utilisateur qui les crée et se trouvent dans le répertoire **/var/spool/cron/crontabs/**.

Par exemple, le fichier propre à l'utilisateur fenestros est le fichier **/var/spool/cron/crontabs/fenestros**.

Le service crond exécute des tâches en rajoutant une ligne dans son fichier de journalisation **/var/log/syslog**.

Si une commande produit une sortie, celle-ci est dirigée vers la messagerie.

L'utilisation de crond est réservé à root. Cependant, vous pouvez établir une liste d'utilisateurs qui ont la permission d'utiliser crond en créant un fichier nommé **cron.allow** dans **/etc**. A l'inverse, un fichier **cron.deny** peut contenir une liste d'utilisateurs qui n'ont pas la permission d'utiliser crond.

Sous Debian **/etc/crontab** prend la forme suivante :

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
```

```

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#

```

<note important> Notez ici que le crontab appelle un binaire, nommé **run-parts**, afin d'exécuter les scripts et commandes se trouvant dans /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly et /etc/cron.monthly. </note>

Dans ce fichier, on peut constater la définition de variables dont SHELL et PATH. Ces variables sont différentes des variables système.

Chaque ligne contient 5 champs temporels qui décrivent la périodicité de l'exécution de la tâche concernée.

Les 5 champs sont :

Minutes	Heures	Jour du mois	Mois de l'année	Jour de la sem.
(0-59)	(0-23)	(1-31)	(1-12)	(0-6 7)*

* le 0 ou le 7 correspond à dimanche.

Les champs temporels peuvent contenir des valeurs différentes :

Exemple	Description
Une valeur absolue telle 10	Dans le champs minutes = 10 minutes après l'heure
Une série de valeurs telle 2,6,8	Dans le champs mois = février, juin et août
Une intervalle telle 1-5	Dans le champs Jour de la Semaine = du lundi au vendredi
Le joker *	Dans le champs minutes = toutes les minutes
Une périodicité telle 0-23/2	Dans le champs heures = toutes les deux heures

Chaque utilisateur peut avoir sa propre version d'un fichier crontab. Pour vérifier s'il existe une version de crontab pour lui, l'utilisateur lancera la

commande suivante :

```
trainee@debian:~$ crontab -l  
no crontab for trainee
```

Afin de créer votre fichier crontab, lancez la commande crontab avec l'option -e en tant que votre nom d'utilisateur et non en tant que root :

```
$ crontab -e [Entrée]
```

Cette commande lance l'interface de l'éditeur **nano**. Tapez ensuite le texte suivant :

```
* * * * * /bin/pwd > pwd.txt
```

Sauvegardez et quittez nano.

Vous obtiendrez un résultat similaire à celui-ci :

```
trainee@debian:~$ crontab -e  
no crontab for trainee - using an empty one  
crontab: installing new crontab
```

<note important> Ce fichier crontab va être exécuté toutes les minutes. La commande va inscrire le répertoire courant dans le fichier pwd.txt.
</note>

Le fichier crontab que vous venez de créer, au nom de votre utilisateur, a été écrit sur le disque dans le répertoire **/var/spool/cron/crontabs/**.

Malgré le fait que vous venez de l'éditer, vous ne pouvez pas visualiser ce fichier en tant que simple utilisateur.

Passez donc en mode super-utilisateur et visualisez le fichier en utilisant la commande cat :

```
root@debian:~# cat /var/spool/cron/crontabs/trainee  
# DO NOT EDIT THIS FILE - edit the master and reinstall.  
# (/tmp/crontab.ZIDAqR/crontab installed on Sat Apr  7 13:40:00 2012)  
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
```

```
# 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
* * * * * /bin/pwd > pwd.txt
```

Afin d'accorder ou non le droit d'éditer son propre fichier crontab, root a la possibilité d'inscrire les noms d'utilisateurs dans un des deux fichiers :

- **/etc/cron.allow** pour accorder le droit d'éditer son propre fichier crontab
- **/etc/cron.deny** pour interdire le droit d'éditer son propre fichier crontab

Par contre, si root inscrit le nom d'un utilisateur dans **/etc/cron.deny** et cet utilisateur a déjà planifié des tâches avec son propre fichier crontab, les tâches seront exécutées. Pour détruire les tâches planifiées de l'utilisateur, il faut détruire le fichier crontab de l'utilisateur.

anacron

L'inconvénient avec crond est qu'afin que celui-ci puisse exécuter des tâches automatiquement, la machine doit rester allumée en permanence.

Il existe un alternatif très efficace pour des machines qui ne sont pas constamment allumées. Cette solution est le logiciel Anacron.

Ouvrez donc le fichier `/etc/anacrontab` :

```
root@debian:~# cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# These replace cron's entries
1 5 cron.daily nice run-parts --report /etc/cron.daily
7 10 cron.weekly nice run-parts --report /etc/cron.weekly
@monthly 15 cron.monthly nice run-parts --report /etc/cron.monthly
```

Vous noterez que ce fichier fait référence aux répertoires `cron.daily`, `cron.weekly` et `cron.monthly`.

Les deux premiers champs sont des champs temporels. Dans le cas d'anacron ceux-ci correspondent à:

Période	Délai
La périodicité en jours	Le délai en minutes avant l'exécution

Anacron mémorise dans les fichiers qui se trouvent dans **`/var/spool/anacron/`** la date à laquelle il a exécuté les commandes qui se trouvent dans `anacrontab` pour la dernière fois. Ces fichiers sont appelés `cron.daily`, `cron.weekly` et `cron.monthly`.

```
root@debian:~# ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly
```

Par exemple :

```
root@debian:~# cat /var/spool/anacron/cron.daily
20120407
root@debian:~# cat /var/spool/anacron/cron.weekly
20120403
root@debian:~# cat /var/spool/anacron/cron.monthly
20120403
```

A chaque allumage de la machine anacron consulte ces fichiers et en fonction de la date inscrite et la périodicité spécifiée dans anacrontab, décide si oui ou non il doit exécuter la ligne de commande.

Sous Debian, pour permettre anacron d'automatiser les tâches, il faut qu'il existe un script 0anacron dans chacun des répertoires cron.daily, cron.weekly et cron.monthly. Son nom commence avec un 0 afin qu'il soit exécuté en premier.

Voici un exemple du script /etc/cron.daily/0anacron :

```
root@debian:~# cat /etc/cron.daily/0anacron
#!/bin/sh
#
# anacron's cron script
#
# This script updates anacron time stamps. It is called through run-parts
# either by anacron itself or by cron.
#
# The script is called "0anacron" to assure that it will be executed
# _before_ all other scripts.

test -x /usr/sbin/anacron || exit 0
anacron -u cron.daily
```

Cette commande, grace à l'opérateur **-u**, met à jour la date se trouvant dans le fichier **/var/spool/anacron/cron.daily**.

Le même principe est utilisé pour les fichiers **/var/spool/anacron/cron.weekly** et **/var/spool/anacron/cron.monthly**.

at

Tout comme avec la commande cron, root a la possibilité d'accorder ou d'interdire aux utilisateurs le droit d'exécuter des tâches avec at en utilisant les deux fichiers suivants :

- **/etc/at.allow**
- **/etc/at.deny**

Si le fichier at.allow existe, seuls les utilisateurs dans ce fichier pourront exécuter at.

En tant que root, mettez en place maintenant deux tâches pour le 31/12/2015 à 13h00 et 14h00 respectivement :

```
root@debian:~# at 13:00 12/31/2015
warning: commands will be executed using /bin/sh
at> pwd > /tmp/test13.atd
at> [^D]
at> <EOT>
job 1 at Thu Dec 31 13:00:00 2015
root@debian:~# at 14:00 12/31/2015
warning: commands will be executed using /bin/sh
at> free > /tmp/test14.atd
at> [^D]
at> <EOT>
job 2 at Thu Dec 31 14:00:00 2015
```

Les fichiers concernant ces deux tâches sont stockés dans le répertoire suivants **/var/spool/cron/atjobs/**.

Par exemple :

```
root@debian:~# ls /var/spool/cron/atjobs/
a00001017126d0  a000020171270c
```

A l'examen du deuxième fichier, vous constaterez un résultat similaire à celui-ci :

```
root@debian:~# cat /var/spool/cron/atjobs/a000020171270c
#!/bin/sh
# atrun uid=0 gid=0
# mail trainee 0
umask 22
XDG_SESSION_COOKIE=9dbc42206eca490459754e5100000008-1333799322.10135-21419052; export XDG_SESSION_COOKIE
USER=root; export USER
MAIL=/var/mail/root; export MAIL
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin; export PATH
PWD=/root; export PWD
LANG=fr_FR.UTF-8; export LANG
SHLVL=1; export SHLVL
HOME=/root; export HOME
LOGNAME=root; export LOGNAME
XAUTHORITY=/var/run/gdm3/auth-for-trainee-UWBRbn/database; export XAUTHORITY
COLORTERM=gnome-terminal; export COLORTERM
OLDPWD=/etc/cron.daily; export OLDPWD
cd /root || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
free > /tmp/test14.atd
```

Pour supprimer cette tâche il convient d'utiliser la commande suivante :

```
# at -d N° de tâche [Entrée]
```

Par exemple :

```
root@debian:~# at -l
1 Thu Dec 31 13:00:00 2015 a root
2 Thu Dec 31 14:00:00 2015 a root
root@debian:~# at -d 1
root@debian:~# at -l
```



```
2 Thu Dec 31 14:00:00 2015 a root
```

Finalement, pour exécuter plusieurs commandes à la même heure d'une manière séquentielle, vous pouvez les insérer dans un fichier :

```
# at 14:30 12/31/2015 < todo.list [Entrée]
```

Par exemple :

```
root@debian:~# touch todo.list
root@debian:~# echo pwd > todo.list
root@debian:~# echo free >> todo.list
root@debian:~# echo who >> todo.list
root@debian:~# cat todo.list
pwd
free
who
root@debian:~# at 14:30 12/31/2015 < todo.list
job 3 at 2015-12-31 14:30
```

~~DISCUSSION:off~~

Donner votre Avis

{(rater>id=debian_6_l110|name=cette page|type=rate|trace=user|tracedetails=1)}

From:
<https://www.ittraining.team/> - **www.ittraining.team**

Permanent link:
<https://www.ittraining.team/doku.php?id=elearning:workbooks:debian:6:l110>

Last update: **2020/01/30 03:28**

