Dernière mise-à-jour : 2020/01/30 03:27

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 Ubuntu **/etc/crontab** prend la forme suivante :

/etc/crontab

```
# /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 )
```

Notez ici que le crontab appel 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.

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 le 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

Exemple	Description
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 :

```
root@ubuntu:~# exit
déconnexion
trainee@ubuntu:~$ 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 :

```
trainee@ubuntu:~$ crontab -e
no crontab for trainee - using an empty one

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

Choose 1-4 [2]: 4
```

Choisissez l'option **4. /usr/bin/vim.tiny** en tant qu'éditeur de fichiers.

Ajoutez le texte * * * * * **/bin/pwd > pwd.txt** au fichier :

```
# 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
~
~
```

Sauvegardez et quittez vim.

Vous obtiendrez un résultat similaire à celui-ci :

```
trainee@ubuntu:~$ crontab -e
no crontab for trainee - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
```

```
  2. /bin/nano        <---- easiest
  3. /usr/bin/mcedit
  4. /usr/bin/vim.tiny

Choose 1-4 [2]: 4
crontab: installing new crontab
```

Ce fichier crontab va être exécuté toutes les minutes. La commande va inscrire le répertoire courant dans le fichier pwd.txt.

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 :

```
trainee@ubuntu:~$ sudo su -
[sudo] password for trainee:
root@ubuntu:~# cat /var/spool/cron/crontabs/trainee
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.HcfPxN/crontab installed on Fri Oct 17 14:47:34 2014)
# (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 allumée. Cette solution est le logiciel Anacron.

Ouvrez donc le fichier /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
HOME=/root
LOGNAME=root

# These replace cron's entries
1   5    cron.daily    run-parts --report /etc/cron.daily
7   10    cron.weekly    run-parts --report /etc/cron.weekly
@monthly    15    cron.monthly    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@ubuntu:~# ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly
```

Par exemple :

```
root@ubuntu:~# cat /var/spool/anacron/cron.daily
20141014
root@ubuntu:~# cat /var/spool/anacron/cron.weekly
20141014
root@ubuntu:~# cat /var/spool/anacron/cron.monthly
20140929
```

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

```
root@ubuntu:~# ls /etc/cron.daily/
0anacron  apt           cracklib-runtime  logrotate  mlocate  popularity-contest      upstart
apport    bsdmainutils  dpkg              man-db     passwd   update-notifier-common
root@ubuntu:~# ls /etc/cron.weekly
0anacron  apt-xapian-index  fstrim  man-db  update-notifier-common
root@ubuntu:~# ls /etc/cron.monthly
0anacron
```

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

/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.

Le binaire at n'est pas installé par défaut sous Ubuntu :

```
root@ubuntu:~# apt-get install at
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Paquets suggérés :
  default-mta mail-transport-agent
Les NOUVEAUX paquets suivants seront installés :
  at
0 mis à jour, 1 nouvellement installés, 0 à enlever et 171 non mis à jour.
Il est nécessaire de prendre 36,0 ko dans les archives.
Après cette opération, 194 ko d'espace disque supplémentaires seront utilisés.
Réception de : 1 http://fr.archive.ubuntu.com/ubuntu/ trusty/main at i386 3.1.14-1ubuntu1 [36,0 kB]
36,0 ko réceptionnés en 0s (311 ko/s)
Sélection du paquet at précédemment désélectionné.
(Lecture de la base de données... 168323 fichiers et répertoires déjà installés.)
Préparation du décompactage de .../at_3.1.14-1ubuntu1_i386.deb ...
Décompactage de at (3.1.14-1ubuntu1) ...
Traitement déclenché pour  ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Traitement déclenché pour  man-db (2.6.7.1-1) ...
Paramétrage de at (3.1.14-1ubuntu1) ...
```

```
atd start/running, process 12200
Traitement déclenché pour  ureadahead (0.100.0-16) ...
```

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

```
root@ubuntu:~# at 13:00 12/31/2015
warning: commands will be executed using /bin/sh
at> pwd > /tmp/test13.atd
at> <EOT>
job 1 at Thu Dec 31 13:00:00 2015

root@ubuntu:~# at 14:00 12/31/2015
warning: commands will be executed using /bin/sh
at> free > /tmp/test14.atd
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@ubuntu:~# ls /var/spool/cron/atjobs/
a00001017126d0  a000020171270c
```

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

```
root@ubuntu:~# cat /var/spool/cron/atjobs/a000020171270c
#!/bin/sh
# atrun uid=0 gid=0
# mail trainee 0
umask 22
XDG_VTNR=7; export XDG_VTNR
XDG_SESSION_ID=c2; export XDG_SESSION_ID
USER=root; export USER
```

```
LS_COLORS=rs=0:di=01\;34:ln=01\;36:mh=00:pi=40\;33:so=01\;35:do=01\;35:bd=40\;33\;01:cd=40\;33\;01:or=40\;31\;01:
su=37\;41:sg=30\;43:ca=30\;41:tw=30\;42:ow=34\;42:st=37\;44:ex=01\;32:\*.tar=01\;31:\*.tgz=01\;31:\*.arj=01\;31:\
*.taz=01\;31:\*.lzh=01\;31:\*.lzma=01\;31:\*.tlz=01\;31:\*.txz=01\;31:\*.zip=01\;31:\*.z=01\;31:\*.Z=01\;31:\*.dz
=01\;31:\*.gz=01\;31:\*.lz=01\;31:\*.xz=01\;31:\*.bz2=01\;31:\*.bz=01\;31:\*.tbz=01\;31:\*.tbz2=01\;31:\*.tz=01\;
31:\*.deb=01\;31:\*.rpm=01\;31:\*.jar=01\;31:\*.war=01\;31:\*.ear=01\;31:\*.sar=01\;31:\*.rar=01\;31:\*.ace=01\;3
1:\*.zoo=01\;31:\*.cpio=01\;31:\*.7z=01\;31:\*.rz=01\;31:\*.jpg=01\;35:\*.jpeg=01\;35:\*.gif=01\;35:\*.bmp=01\;35
:\*.pbm=01\;35:\*.pgm=01\;35:\*.ppm=01\;35:\*.tga=01\;35:\*.xbm=01\;35:\*.xpm=01\;35:\*.tif=01\;35:\*.tiff=01\;35
:\*.png=01\;35:\*.svg=01\;35:\*.svgz=01\;35:\*.mng=01\;35:\*.pcx=01\;35:\*.mov=01\;35:\*.mpg=01\;35:\*.mpeg=01\;3
5:\*.m2v=01\;35:\*.mkv=01\;35:\*.webm=01\;35:\*.ogm=01\;35:\*.mp4=01\;35:\*.m4v=01\;35:\*.mp4v=01\;35:\*.vob=01\;
35:\*.qt=01\;35:\*.nuv=01\;35:\*.wmv=01\;35:\*.asf=01\;35:\*.rm=01\;35:\*.rmvb=01\;35:\*.flc=01\;35:\*.avi=01\;35
:\*.fli=01\;35:\*.flv=01\;35:\*.gl=01\;35:\*.dl=01\;35:\*.xcf=01\;35:\*.xwd=01\;35:\*.yuv=01\;35:\*.cgm=01\;35:\*
.emf=01\;35:\*.axv=01\;35:\*.anx=01\;35:\*.ogv=01\;35:\*.ogx=01\;35:\*.aac=00\;36:\*.au=00\;36:\*.flac=00\;36:\*.
mid=00\;36:\*.midi=00\;36:\*.mka=00\;36:\*.mp3=00\;36:\*.mpc=00\;36:\*.ogg=00\;36:\*.ra=00\;36:\*.wav=00\;36:\*.a
xa=00\;36:\*.oga=00\;36:\*.spx=00\;36:\*.xspf=00\;36:; export LS_COLORS
MAIL=/var/mail/root; export MAIL
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin; export PATH
QT_QPA_PLATFORMTHEME=appmenu-qt5; export QT_QPA_PLATFORMTHEME
PWD=/root; export PWD
LANG=fr_FR.UTF-8; export LANG
SHLVL=1; export SHLVL
HOME=/root; export HOME
XDG_SEAT=seat0; export XDG_SEAT
LOGNAME=root; export LOGNAME
LESSOPEN=\|\ /usr/bin/lesspipe\ %s; export LESSOPEN
LESSCLOSE=/usr/bin/lesspipe\ %s\ %s; export LESSCLOSE
XAUTHORITY=/home/trainee/.Xauthority; export XAUTHORITY
COLORTERM=gnome-terminal; export COLORTERM
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@ubuntu:~# at -l
2   Thu Dec 31 14:00:00 2015 a root
1   Thu Dec 31 13:00:00 2015 a root
root@ubuntu:~# atq
2   Thu Dec 31 14:00:00 2015 a root
1   Thu Dec 31 13:00:00 2015 a root
root@ubuntu:~# at -d 1
root@ubuntu:~# 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@ubuntu:~# touch todo.list
root@ubuntu:~# echo pwd > todo.list
root@ubuntu:~# echo free >> todo.list
root@ubuntu:~# echo who >> todo.list
root@ubuntu:~# cat todo.list
pwd
free
who
root@ubuntu:~# at 14:30 12/31/2015 < todo.list
warning: commands will be executed using /bin/sh
job 3 at Thu Dec 31 14:30:00 2015
```

<html>

Copyright © 2004-2016 Hugh Norris.

Ce(tte) oeuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.

</html>