| 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 répertoires **/var/spool/cron/tabs/**.

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

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

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

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
#
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly
#
```

```
-*/15 * * * *   root  test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons >/dev/null 2>&1
```

<note important> Notez ici que le crontab appel un script appelé **run-crons** 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, MAILTO 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)* |

* le 0 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@opensuse:~> 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 **vi**. Tapez ensuite le texte suivant :

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

Sauvegardez et quittez vi.

Vous obtiendrez un résultat similaire à celui-ci :

```
trainee@opensuse:~> 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/tabs/**.

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 :

```
opensuse:~ # cat /var/spool/cron/tabs/trainee
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.5LHBJm installed on Sat Apr  7 15:37:52 2012)
# (Cronie version 4.2)
* * * * * /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.

Sous openSUSE, anacron n'est pas installé par défaut. Il convient donc de l'installer à l'aide de la commande **zypper** :

```
opensuse:~ # zypper install cronie-anacron
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  cronie-anacron

1 new package to install.
Overall download size: 27.0 KiB. After the operation, additional 39.0 KiB will
be used.
Continue? [y/n/?] (y): y
Retrieving package cronie-anacron-1.4.7-9.23.1.i586 (1/1), 27.0 KiB (39.0 KiB unpacked)
Retrieving: cronie-anacron-1.4.7-9.23.1.i586.rpm [done (261.4 KiB/s)]
Installing: cronie-anacron-1.4.7-9.23.1 [done]
```

Ouvrez donc le fichier /etc/anacrontab :

```
opensuse:~ # cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

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

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

```
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1   5    cron.daily        nice /usr/lib/cron/run-crons /etc/cron.daily
7   25    cron.weekly        nice /usr/lib/cron/run-crons /etc/cron.weekly
@monthly 45 cron.monthly        nice /usr/lib/cron/run-crons /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.

```
opensuse:~ # ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly
```

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.

Pour permettre anacron d'automatiser les tâches, il faut qu'il existe un script 0anacron dans le répertoire cron.hourly. Son nom commence avec un 0 afin qu'il soit exécuté en premier.

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

```
opensuse:~ # cat /etc/cron.hourly/0anacron
#!/bin/bash
```

```
#in case file doesn't exist
if test -r /var/spool/anacron/cron.daily; then
    day=`cat /var/spool/anacron/cron.daily`
fi
if [ `date +%Y%m%d` = "$day" ]; then
    exit 0;
fi

# in case anacron is already running,
# there will be log (daemon won't be running twice).
if test -x /usr/bin/on_ac_power; then
    /usr/bin/on_ac_power &> /dev/null
    if test $? -eq 1; then
    exit 0
    fi
fi
/usr/sbin/anacron -s
```

Ce script a pour but d'exécuter la commande **/usr/sbin/anacron -s**. L'option **-s** indique à anacron d'attendre la fin d'exécution d'un job avant d'exécuter le suivant.

# 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 :

```
opensuse:~ # at 13:00 12/31/2015
warning: commands will be executed using /bin/sh
at> pwd > /tmp/test13.atd
at> [^D]
at> <EOT>
job 2 at 2015-12-31 13:00
Can't open /var/run/atd.pid to signal atd. No atd running?
```

Notez que le système vous indique que le daemon **atd** ne s'exécute pas. Vérifier donc le nom du daemon :

```
opensuse:~ # ls /etc/init.d | grep at
atd
boot.multipath
inputattach
multipathd
skeleton.compat
```

Vérifiez ensuite son statut :

```
opensuse:~ # service atd status
Checking for at daemon:                                          unused
```

En effet, le daemon n'est pas démarré. Il convient donc de démarrer atd :

```
opensuse:~ # service atd start
Starting service at daemon                                       done
```

Mettez en place maintenant une deuxième tâche pour le 31/12/2015 à 14h00 :

```
opensuse:~ # at 14:00 12/31/2015
warning: commands will be executed using /bin/sh
at> free > /tmp/test14.atd
at> [^D]
```

```
at> <EOT>
job 3 at 2015-12-31 14:00
```

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

Par exemple :

```
opensuse:~ # ls /var/spool/atjobs
.SEQ  a00002017126d0  a000030171270c
```

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

```
opensuse:~ # cat /var/spool/atjobs/a000030171270c
#!/bin/sh
# atrun uid=0 gid=0
# mail  trainee 0
umask 22
LESSKEY=/etc/lesskey.bin; export LESSKEY
NNTPSERVER=news; export NNTPSERVER
MANPATH=/usr/share/man:/usr/local/man; export MANPATH
HOSTNAME=opensuse; export HOSTNAME
XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB; export XKEYSYMDB
HOST=opensuse; export HOST
PROFILEREAD=true; export PROFILEREAD
HISTSIZE=1000; export HISTSIZE
MORE=-sl; export MORE
JRE_HOME=/usr/lib/jvm/jre; export JRE_HOME
USER=root; export USER
LS_COLORS=no=00:fi=00:di=01\;34:ln=00\;36:pi=40\;33:so=01\;35:do=01\;35:bd=40\;33\;01:cd=40\;33\;01:or=41\;33\;01
:ex=00\;32:\*.cmd=00\;32:\*.exe=01\;32:\*.com=01\;32:\*.bat=01\;32:\*.btm=01\;32:\*.dll=01\;32:\*.tar=00\;31:\*.t
bz=00\;31:\*.tgz=00\;31:\*.rpm=00\;31:\*.deb=00\;31:\*.arj=00\;31:\*.taz=00\;31:\*.lzh=00\;31:\*.lzma=00\;31:\*.z
ip=00\;31:\*.zoo=00\;31:\*.z=00\;31:\*.Z=00\;31:\*.gz=00\;31:\*.bz2=00\;31:\*.tb2=00\;31:\*.tz2=00\;31:\*.tbz2=00
\;31:\*.xz=00\;31:\*.avi=01\;35:\*.bmp=01\;35:\*.fli=01\;35:\*.gif=01\;35:\*.jpg=01\;35:\*.jpeg=01\;35:\*.mng=01\
;35:\*.mov=01\;35:\*.mpg=01\;35:\*.pcx=01\;35:\*.pbm=01\;35:\*.pgm=01\;35:\*.png=01\;35:\*.ppm=01\;35:\*.tga=01\;
```

```
35:\*.tif=01\;35:\*.xbm=01\;35:\*.xpm=01\;35:\*.dl=01\;35:\*.gl=01\;35:\*.wmv=01\;35:\*.aiff=00\;32:\*.au=00\;32:
\*.mid=00\;32:\*.mp3=00\;32:\*.ogg=00\;32:\*.voc=00\;32:\*.wav=00\;32:; export LS_COLORS
XNLSPATH=/usr/share/X11/nls; export XNLSPATH
HOSTTYPE=i386; export HOSTTYPE
FROM_HEADER=; export FROM_HEADER
PAGER=less; export PAGER
CSHEDIT=emacs; export CSHEDIT
XDG_CONFIG_DIRS=/etc/xdg; export XDG_CONFIG_DIRS
MINICOM=-c\ on; export MINICOM
MAIL=/var/spool/mail/root; export MAIL
PATH=/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11R6/bin:/usr/game
s:/usr/lib/jvm/jre/bin; export PATH
CPU=i686; export CPU
JAVA_BINDIR=/usr/lib/jvm/jre/bin; export JAVA_BINDIR
INPUTRC=/etc/inputrc; export INPUTRC
PWD=/root; export PWD
JAVA_HOME=/usr/lib/jvm/jre; export JAVA_HOME
LANG=POSIX; export LANG
PYTHONSTARTUP=/etc/pythonstart; export PYTHONSTARTUP
GPG_TTY=/dev/pts/0; export GPG_TTY
QT_SYSTEM_DIR=/usr/share/desktop-data; export QT_SYSTEM_DIR
SHLVL=1; export SHLVL
HOME=/root; export HOME
SDL_AUDIODRIVER=pulse; export SDL_AUDIODRIVER
ALSA_CONFIG_PATH=/etc/alsa-pulse.conf; export ALSA_CONFIG_PATH
LESS_ADVANCED_PREPROCESSOR=no; export LESS_ADVANCED_PREPROCESSOR
OSTYPE=linux; export OSTYPE
LS_OPTIONS=-A\ -N\ --color=tty\ -T\ 0; export LS_OPTIONS
XCURSOR_THEME=DMZ; export XCURSOR_THEME
WINDOWMANAGER=/usr/bin/gnome; export WINDOWMANAGER
LESS=-M\ -I\ -R; export LESS
MACHTYPE=i686-suse-linux; export MACHTYPE
LOGNAME=root; export LOGNAME
XDG_DATA_DIRS=/usr/share; export XDG_DATA_DIRS
```

```
LC_CTYPE=fr_FR.UTF-8; export LC_CTYPE
LESSOPEN=lessopen.sh\ %s; export LESSOPEN
XAUTHLOCALHOSTNAME=localhost; export XAUTHLOCALHOSTNAME
LESSCLOSE=lessclose.sh\ %s\ %s; export LESSCLOSE
G_BROKEN_FILENAMES=1; export G_BROKEN_FILENAMES
JAVA_ROOT=/usr/lib/jvm/jre; export JAVA_ROOT
COLORTERM=1; export COLORTERM
XAUTHORITY=/root/.xauthwrrVC8; export XAUTHORITY
mc=\(\)\ {\ \ .\ /usr/share/mc/mc-wrapper.sh"
"}; export mc
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 :

```
opensuse:~ # at -l
2   2015-12-31 13:00 a root
3   2015-12-31 14:00 a root
opensuse:~ # at -d 2
opensuse:~ # at -l
3   2015-12-31 14:00 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 :

```
opensuse:~ # touch todo.list
opensuse:~ # echo pwd > todo.list
opensuse:~ # echo free >> todo.list
opensuse:~ # echo who >> todo.list
opensuse:~ # cat todo.list
pwd
free
who
opensuse:~ # at 14:30 12/31/2015 < todo.list
job 4 at 2015-12-31 14:30
```

~~DISCUSSION:off~~

# Donner votre Avis

{(rater>id=opensuse_11_l110|name=cette page|type=rate|trace=user|tracedetails=1)}