



Administration système et sécurité sous LINUX



Planifier des tâches Avec CRON

Equipe Système

CRON

Cron est un *daemon* utilisé pour programmer des tâches devant être exécutées à un moment précis.

Chaque utilisateur a un fichier **crontab**, lui permettant d'indiquer les actions et à quelles périodes, elles devront être exécutées.

Il y a également une **crontab** pour le système, permettant les tâches techniques, pour la mise à jour des différents programmes ou autres besoins périodiques.

System CRON jobs

System CRON jobs

Le fichier de configuration du **cron** est **/etc/crontab**,
il est conseillé de mettre les scripts à lancer dans les dossiers
préconfigurer :

/etc/cron.hourly : les scripts qui existent dans ce dossier ce
lancent chaque heure.

/etc/cron.daily

/etc/cron.weekly

/etc/cron.mounthly

System CRON jobs

Pour utiliser **cron**, ajoutez simplement les entrées à votre fichier **/etc/crontab**, Chaque ligne représente une commande à exécuter. La table de configuration doit être remplie de la façon suivante :

```
mm hh jj MM JJ user commande > log
```

mm : les minutes 0-59	JJ : le jour de la semaine 0-7 0 et 7 représentent le dimanche
hh : les heures 00-23	user : le nom de l'utilisateur
jj : le numéro de jour du mois 1-31	Commande : commande à lancer
MM : le numéro du mois 1-12	Log : le nom du fichier log

System CRON jobs

Exemples

```
1 2 3 4 * root /usr/bin/apt-get update
```

Cela va exécuter la commande **/usr/bin/apt-get update** tous les Avril (4), qui tombent un 3 Avril (3), à 2h01.

une étoile (toutes les valeurs) :

```
* 2 * 4 5 root /usr/bin/apt-get update
```

tous les vendredis (5), d'Avril (4), à 2h (2), et toutes les minutes (*).

System CRON jobs

Une plage:

```
1 2 * 4 5-7 root /usr/bin/apt-get update
```

tous les vendredis, samedis et dimanches (5-7), d'Avril (4), à 2h01

un multiple :

```
*/10 2 3 4 * root /usr/bin/apt-get update
```

tous les Avril (4), qui tombent un 3 Avril (3), à 2h et toutes les 10 minutes (*/10).

System CRON jobs

Une plage:

```
1 2 3 4 3,5,7 root /usr/bin/apt-get update
```

tous les mercredis, vendredis et dimanches (4,5,7), d'Avril (4), qui tombent un 3 Avril (3), à 2h01.

```
1 2 3 4 * root run-parts /home/arafet/mes_taches
```

L'option run-parts est utilisée pour lancer un ensemble de commandes dans un répertoire bien déterminé.

System CRON jobs

Il existe des raccourcis intéressants :

@reboot # se lance au reboot avec les droits utilisateurs, bien commode

@yearly

@annually

@monthly

@weekly

@daily

@midnight

@hourly

User CRON jobs

User CRON jobs

Tous les utilisateurs peuvent planifier l'exécution de tâches. c'est pourquoi chacun dispose de sa propre **crontab**.

Un utilisateur peut éditer les commandes à planifier :
crontab -e

Afficher la liste des tâches programmées
crontab -l

00 * * * * /usr/bin/xclock -display :0.0

Supprimer la table cron
crontab -r

User CRON jobs

ses informations sont stockées dans :

- /var/spool/cron/crontabs/<utilisateur>
- /var/spool/cron/<utilisateur>
- ou /var/spool/cron/tabs/<utilisateur>

Selon les distributions

Pour l'administrateur root :

Afficher la liste des tâches programmées d'un utilisateur :

#crontab -u salah -l

00 * * * * /usr/bin/xclock -display :0.0

Supprimer la table cron d'un utilisateur :

crontab -u salah -r

User CRON jobs

La sécurité dans CRON :

Les fichiers cron.allow et cron.deny contrôlent l'utilisation de la commande crontab.

/etc/cron.allow

/etc/cron.deny

cron.allow	cron.deny	Utilisateurs autorisés
Présent	Présent	Ceux explicitement dans cron.allow
Présent	-	Ceux explicitement dans cron.allow
-	Présent	Tous sauf ceux dans cron.deny
-	-	Uniquement root



ANACRON jobs

anachronistic command scheduler

ANACRON jobs

l'inconvénient du système **CRON** est si le système est stoppé au moment de l'exécution de la tâche, celle-ci est ignorée.

Anacron est donc né pour permettre l'exécution de tâches sans nécessité de laisser la machine allumée 24h/24.

Notez bien qu'anacron n'est pas du tout destiné à remplacer cron, il est complémentaire et il coexiste avec lui...

ANACRON jobs

anacron utilise des indications de temps relatives (« une fois par jour / par semaine / par mois ») au lieu de références temporelles absolues (« le 14 janvier 2008 à 15h 30 »).

De la sorte, même si vous « manquez » un moment ou une date particulière où l'exécution d'un « job » était prévue, celui-ci sera tout de même exécuté peu de temps après le prochain démarrage du système.

ANACRON jobs

Anacron est un **programme** « standard » (c'est pas un démon/service), il est lancé :

- au **démarrage du système** via un script d'init (/etc/init.d/anacron)
- via le **crontab système**, on verra ce dernier point lors de la description des interaction cron/anacron

Le fichier de configuration d'anacron est **/etc/anacrontab**

ANACRON jobs

Le fichier de configuration d'anacron est **/etc/anacrontab**

```
#####  
#périodicité (jours),                                #  
#|   délai (minutes),                                #  
#|   |   nom de la tâche,                             #  
#|   |   |   commandes                                #  
#####  
1 5  Montre  /usr/bin/xclock -display :0.0
```

Dans l'exemple ci-dessus la commande xclock sera exécutée tous les jours 5 minutes après le démarrage d'anacron.

ANACRON jobs

Si la machine reboot plusieurs fois dans la journée, anacron est exécuté plusieurs fois, afin de ne pas lancer une tâche quotidienne à chaque reboot anacron stock la date (format AAAAMMJJ) de sa dernière exécution dans un **fichier de « log »** portant le nom de la tâche et présent dans `/var/spool/anacron`.

Lorsqu'il parcourt les tâches qui lui sont affecté, anacron vient donc lire le fichier de « log » correspondant, compare la date du jour, la date de dernière exécution et l'intervalle de la tâche et la lance le cas échéant.

The background of the slide is a collage of educational scenes. At the top, there is a banner with a globe on the left and a classroom of students on the right. Below this, there is a large, faded image of a classroom with students at their desks. In the center, the text 'At jobs' is displayed. At the bottom, there is another faded image of a classroom with students at their desks and a teacher at the front.

At jobs

at et le demon atd

Atd est un démon qui s'occupe des commandes à exécuter une seule fois à un instant précis et futur

De nombreuses tâches sont régulièrement planifiées :

- la rotation des logs;
- mise à jour de la base de données du programme locate;
- les sauvegardes;
- des scripts d'entretien (comme le nettoyage des fichiers temporaires).

at et le demon atd

Je souhaite supprimer tous les fichiers *.tmp du répertoire tmp :
dans 2 minutes :

at now + 2 minutes

Entrée

rm /tmp/*.tmp *Entrée* "Ctrl+D"

à 13h37 :

at 13:37

Entrée

rm /tmp/*.tmp *Entrée* "Ctrl+D"

at et le demon atd

La commande :

atq pour voir la liste de taches

atrm -V num_tache : pour supprimer une tache