

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

# Planification des tâches sous Gnu/Linux

[www.ofppt.info](http://www.ofppt.info)



**DIRECTION RECHERCHE ET INGENIERIE DE FORMATION  
SECTEUR NTIC**

## Sommaire

1.	Introduction .....	2
2.	Crontab .....	2
3.	La commande at .....	4
4.	La commande batch .....	4

# 1.Introduction

Le démon **cron** exécute des commandes pour un utilisateur, selon des critères temporels : date et heure :

- La périodicité (tou(te)s les) quand elles sont soumises par la commande **crontab**.
- L'échéance (à telle heure) quand elles sont soumises par la commande **at**.
- A leur tour et dès que possible quand elles sont soumises par la commande **batch**.

## 2.Crontab

Il est possible de déclencher l'exécution de certaines opérations à intervalles réguliers sous Linux. Ces opérations sont définies pour le système et pour chaque utilisateur. Elles sont enregistrées dans des fichiers de configuration indiquant le moment où elles doivent être déclenchées, et quelle action elles doivent réaliser. Les opérations définies pour le système sont stockées dans le fichier de configuration `/etc/crontab`. Des commandes additionnelles peuvent être définies dans les répertoires `/etc/cron.d/`, `/etc/cron.daily/`, `/etc/cron.weekly/` et `/etc/cron.monthly/`. Par ailleurs, les fichiers de configuration des utilisateurs sont stockés dans le répertoire `/var/cron/tab/`, sous le nom de chaque utilisateur. Il est bien entendu possible d'éditer ces fichiers en tant que root, mais ce n'est pas recommandé. En effet, la commande **crontab** permet d'installer, de supprimer et de consulter les fichiers crontab de chaque utilisateur, et ce de manière sûre.

La commande **crontab** peut être utilisée pour afficher le contenu du fichier de configuration de l'utilisateur qui l'appelle, à l'aide de l'option `-l` :

```
crontab -l
```

Elle permet également de supprimer ce fichier, à l'aide de l'option `-r` :

```
crontab -r
```

Enfin, l'option `-e` permet d'éditer le fichier crontab, à l'aide de l'éditeur spécifié dans la variable d'environnement `VISUAL` ou `EDITOR`. Par défaut, l'éditeur `vi` sera utilisé.

En tant qu'administrateur du système, il est possible de modifier les paramètres pour n'importe quel utilisateur. Pour cela, il faut préciser le login de l'utilisateur avec l'option `-u`. Il est recommandé d'utiliser également l'option `-u` si l'on a effectué un **su**, car la commande **crontab** peut ne pas pouvoir déterminer l'utilisateur qui l'a appelé dans ce cas.

Le format des fichiers crontab est suffisamment riche pour permettre de spécifier avec finesse les conditions d'exécution des opérations programmées. En général, le début du fichier contient la définition de variables d'environnement utilisées

<b>OFPPT @</b>	Document	Millésime	Page
	Planification des tâches sous Gnu/Linux	janvier 15	2 - 5

## Planification des tâches sous Gnu/Linux

par crontab. La suite du fichier est réservée aux commandes programmées. Chaque programmation est réalisée sur une ligne du fichier crontab. Les lignes contiennent 5 champs spécifiant la date et l'heure à laquelle la commande doit être exécutée, un nom d'utilisateur éventuel et la commande elle-même. Le nom d'utilisateur ne doit être spécifié que dans le fichier /etc/crontab, qui définit les commandes du système. Il spécifie alors au nom de quel utilisateur la commande doit être exécutée. Pour les fichiers crontab propres à chaque utilisateur, il n'est bien entendu pas nécessaire d'indiquer ce nom.

Les 5 champs de la partie décrivant la date d'exécution de la commande fournissent respectivement les informations suivantes :

- les minutes (comprises entre 0 et 59) ;
- les heures (comprises entre 0 et 23) ;
- le jour dans le mois (compris entre 0 et 31) ;
- le mois (compris entre 0 et 12, ou indiqué par les trois premières lettres du nom du mois en anglais) ;
- le jour dans la semaine (compris entre 0 et 7, ou indiqué par les trois premières lettres du nom du jour en anglais).

Les numéros de mois 0 et 12 correspondent à Janvier, et les numéros de jours 0 et 7 correspondent au Dimanche.

La commande sera exécutée à chaque fois que le jour, le mois, l'heure et les minutes du système correspondront avec ces 5 champs. Il suffit que l'une des spécifications du jour corresponde pour que la commande soit exécutée (c'est-à-dire qu'elle est exécutée une fois pour le jour du mois et une fois pour le jour de la semaine si ces deux champs sont spécifiés).

Il est possible d'utiliser un intervalle de valeurs pour chacun de ces champs, en indiquant la première et la deuxième valeur, séparées d'un tiret. Il est également possible de faire une liste de valeurs et d'intervalles, en séparant chaque donnée par une virgule. Si l'on veut spécifier toutes les valeurs possibles pour un champ, on peut utiliser le caractère '\*'. Enfin, il est possible d'indiquer que la commande doit être exécutée toutes les n valeurs pour chaque champ. Pour cela, il suffit de faire suivre le champ d'une barre oblique de division ('/') et du nombre n. Ainsi, si l'on trouve l'expression « \*/3 » pour les heures, la commande sera exécutée toutes les trois heures.

La spécification de la commande doit être faite sur une seule ligne. Le caractère de pourcentage ('%') a une signification spéciale, sauf s'il est précédé d'un antislash ('\'). Les données qui suivent le premier pourcentage sont passées telles quelles dans l'entrée standard de la commande. Les caractères pourcentages suivants sont interprétés comme des saut de lignes (donc une validation). Ainsi, la commande suivante :

```
rm -i file.txt%y%
```

permet de supprimer le fichier file.txt et de répondre 'y' à la commande **rm**. Le caractère 'y' est passé ici dans le flux d'entrée standard de **rm**.

Comme vous pouvez le voir, le fichier /etc/crontab du système permet de

OFPPT @	Document	Millésime	Page
	Planification des tâches sous Gnu/Linux	janvier 15	3 - 5

programmer des opérations périodiques, comme les sauvegardes, la destruction des fichiers temporaires, ou toute autre tâche de maintenance. Ne vous étonnez donc pas si votre ordinateur semble s'activer tout seul régulièrement, à heure fixe (par exemple, sur le coup de 11 heures ou minuit). C'est le fonctionnement normal de votre système, qui s'occupe de toutes les tâches ménagères qu'il s'est réservé pour une heure où normalement tout le monde dort...

Les utilisateurs peuvent également définir leur propre crontab pour effectuer les opérations périodiques qu'il désirent. Par exemple, ils peuvent programmer une commande qui leur rappellera un rendez-vous.

### 3. La commande at

La commande at exécute une commande à une date et une heure données. La commande est lue depuis l'entrée standard. Si celle-ci est stockée dans un fichier, il faudra rediriger celui-ci vers la commande.

```
$ at heure [date] < fichier_commandes
```

#### Exemples :

Exécution de fic\_commandes à 15 heures

```
at 1500 < fic_commandes
```

Exécution à minuit

```
at midnight < fic_commandes :
```

Exécution dans trois heures

```
at now + 3 hours < fic_commandes
```

- La liste des travaux est visible avec **at -l**. La liste apparaît sous forme de fichiers **xxxxxxxx.a**
- On supprime un travail avec **at -r xxxxxxxx.a**

### 4. La commande batch

La commande batch fonctionne comme at mais on ne précise pas d'heure ni de date. La tâche est placée dans un file d'attente (FIFO) et est exécutée quand c'est son tour.

```
batch < fic_commandes
```

<b>OFPPT @</b>	Document	Millésime	Page
	Planification des tâches sous Gnu/Linux	janvier 15	4 - 5