

Ecole Nationale des Sciences Appliquées
Khouribga

Unix/Linux Administration Centrale

Med AMNAI
2016-2017

Plan

1. Introduction

2. Cron

1. Configuration d'une tâche cron
2. Cron table système
3. Création d'un crontab
4. Lancement et interruption du service
5. Journalisation
6. Sécurité
7. Commande crontab

3. Anacron

1. Fichier de configuration
2. Journalisation
3. Lancement et interruption du service

4. AT

1. Ajout d'un travail
2. Contrôle des travaux
3. Sécurité
4. Sorties et erreurs

1-Introduction

Sous Linux, des tâches peuvent être configurées pour **s'exécuter automatiquement** pendant une **période de temps donnée** et à **des dates données**.

Un administrateur système peut utiliser des **tâches automatisées** pour effectuer des copies de **sauvegarde périodiques**, **surveiller le système**, **exécuter des scripts personnalisés**, etc.

2- Cron

- ❑ **Cron** est un outil qui permet d'automatiser l'exécution des tâches répétitives.
- ❑ Il se présente sous la forme d'un **démon crond** utilisant un ensemble de fichiers consignant **les travaux à effectuer avec la période d'exécution associée** : les fichiers "**crontab**" (**/var/spool/cron**).
- ❑ La période d'exécution est une combinaison de **l'heure**, du **jour** du mois, du **mois**, du jour de la **semaine** et de la **semaine**.
- ❑ **Cron suppose que le système est allumé en continu**. Si le système n'est pas allumé au moment où une tâche doit être exécutée, l'exécution n'a pas lieu.
- ❑ Pour configurer des tâches **basées sur des périodes** au lieu d'heures précises, utiliser **anacron**.

2- Cron

Le démon **crond** de **CRON** est lancé par les scripts de démarrage du **boot** :

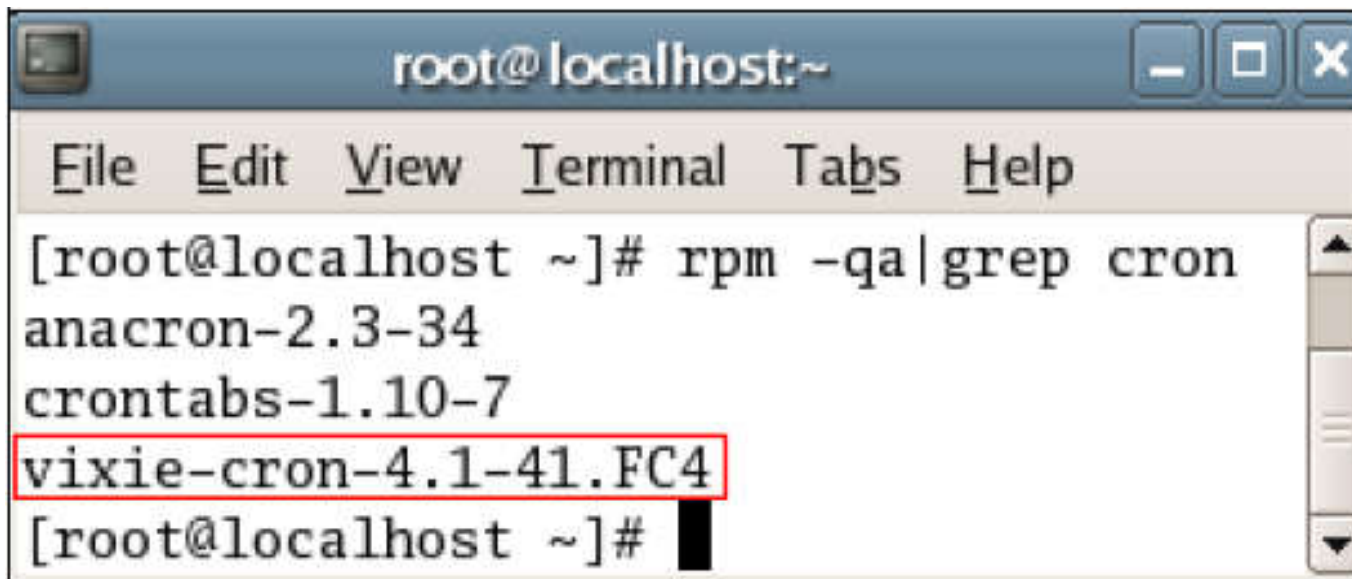
A terminal window titled 'amnai : bash' with a menu bar containing 'Fichier', 'Édition', 'Affichage', 'Signets', 'Configuration', and 'Aide'. The terminal shows the user 'amnai' at 'localhost' in the home directory. The command 'ls -l /etc/rc.d/init.d/crond' is entered and executed, resulting in the output: '-rwxr-xr-x. 1 root root 2793 13 août 2010 /etc/rc.d/init.d/crond'. The path is highlighted in green. The prompt returns to the user.

```
amnai : bash
Fichier Édition Affichage Signets Configuration Aide
[amnai@localhost ~]$
[amnai@localhost ~]$ ls -l /etc/rc.d/init.d/crond
-rwxr-xr-x. 1 root root 2793 13 août 2010 /etc/rc.d/init.d/crond
[amnai@localhost ~]$
```

2- Cron

Pour utiliser le service **cron**, le **paquetage RPM cronie** doit être installé.

Pour savoir si le paquetage est installé, utilisez la commande :

A terminal window titled 'root@localhost:~' with standard window controls. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The command '[root@localhost ~]# rpm -qa | grep cron' has been executed, resulting in three lines of output: 'anacron-2.3-34', 'crontabs-1.10-7', and 'vixie-cron-4.1-41.FC4'. The last line is highlighted with a red rectangular box. The prompt '[root@localhost ~]#' is visible at the bottom.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# rpm -qa | grep cron  
anacron-2.3-34  
crontabs-1.10-7  
vixie-cron-4.1-41.FC4  
[root@localhost ~]#
```

2- Cron

- ❑ **Cron** recherche dans le répertoire `/var/spool/cron`, des **fichiers de crontab** ayant des noms existants dans `/etc/passwd`. Les fichiers trouvés sont chargés en mémoire.
- ❑ Ensuite, **cron** se réveille toutes les **minutes**, examine les **crontab mémorisés**, et vérifie chaque commande pour savoir s'il doit la lancer dans la minute à venir.
- ❑ Lors de **l'exécution d'une commande**, toute **sortie est envoyée par mail** au **propriétaire** de la **crontab** (ou à l'utilisateur dont le **nom est mentionné** dans la variable d'environnement **MAILTO** si elle existe).

2- Cron

- ☐ De plus, **cron vérifie chaque minute si la date de modification de son répertoire de stockage a changé.**
- ☐ **Si c'est le cas, cron examinera les dates de modifications de chaque fichier crontab, et rechargera ceux qui ont été changés.**
- ☐ **Ainsi, cron n'a pas besoin d'être redémarré après la modification d'un fichier crontab**

2.1- Configuration D'une Tâche Cron

Prenons comme exemple le **fichier crontab** suivant :

Les lignes d'initialisation des variables d'environnement. Elles permettent de définir l'environnement dans lequel les travaux **cron doivent être exécutés**.

```
MAILTO=hakimb
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
HOME=/home/hakimb
# Commentaire

55 23 31 12 * /etc/jobs/year.sh
00 20 1 * * /etc/jobs/month.sh
30 8 * * 1 /etc/jobs/week.sh
20 12 * * * /etc/jobs/day.sh
40 * * * * /etc/jobs/hour.sh
* * * * * /etc/jobs/minute.sh    %chaque minute
```

2.1- Configuration d'une tâche Cron

Définition des variables

- ☐ Les variables **LOGNAME**, **HOME** et **SHELL** sont prédéfinies et contiennent respectivement **le nom du compte**, **le répertoire personnel** indiqués dans **/etc/passwd** pour le propriétaire de la cron table, et **/bin/sh** comme interpréteur **des commandes**.
- ☐ Les variables **HOME** et **SHELL** peuvent être modifiées, ce qui n'est pas le cas de **LOGNAME**.
- ☐ **MAILTO** est une autre variable utilisée par **cron**. Si celle-ci n'est pas définie, les sorties standard (**stdout**) et d'erreur (**stderr**) des commandes de la cron table seront envoyées par mail au propriétaire de la crontab. Il est possible d'affecter le nom d'un autre utilisateur à cette variable pour envoyer le message à celui-ci.
- ☐ Pour qu'aucun message ne soit envoyé, il faudra spécifier **MAILTO=""** ou rediriger les sorties des commandes.

2.1- Configuration d'une tâche Cron

Définition des commandes

Les lignes de commandes pour le démon **crond** définissent les travaux à **lancer périodiquement**.

Une ligne de commandes **cron** est constituée de deux informations :

- ☐ La période d'exécution de la tâche.
- ☐ La ligne de commandes Shell à interpréter pour lancer la tâche.

2.1- Configuration d'une tâche Cron

Définition des commandes

Chacune des lignes du fichier **crontab** a le format :

minute	hour	day	month	dayofweek	command
minute	nombre entier entre 0 et 59				
hour	nombre entier entre 0 et 23				
day	nombre entier entre 1 et 31				
month	nombre entier entre 1 et 12 (ou le nom ANSI sur trois lettres).				
	1	jan	janvier		
	2	feb	février		
	3	mar	mars		

2.1- Configuration d'une tâche Cron

Définition des commandes

Chacune des lignes du fichier **crontab** a le format :

minute	hour	day	month	dayofweek	command
dayofweek		nombre entier entre 0 et 7, 0 ou 7 désigne dimanche (ou le nom ANSI sur trois lettres)			
		0	sun	dimanche	
		1	mon	lundi	
		2	tue	mardi	
		7	sun	dimanche	
command		commande ou script à exécuter.			

2.1- Configuration d'une tâche Cron

Comme vous pouvez le voir dans le fichier crontab précédent, le service **cron** est utilisé pour **exécuter les jobs des scripts suivants** :

minute	hour	day	month	dayofweek	command
55	23	31	12	*	/etc/jobs/year.sh
00	20	1	*	*	/etc/jobs/month.sh
30	8	*	*	1	/etc/jobs/week.sh
20	12	*	*	*	/etc/jobs/day.sh
40	*	*	*	*	/etc/jobs/hour.sh
*	*	*	*	*	/etc/jobs/minute.sh %chaque minute

/etc/jobs/year.sh	le 31 décembre de chaque année à 23h55
/etc/jobs/month.sh	le 1 ^{ier} de chaque mois à 20h
/etc/jobs/week.sh	chaque lundi à 8h30
/etc/jobs/day.sh	chaque jour à midi 12h20 (midi 20)
/etc/jobs/hour.sh	toutes les heures 40 min (8h40, 9h40, 10h40...)
/etc/jobs/minute.sh	toutes les minutes

2.1- Configuration d'une tâche Cron

Pour les valeurs ci-dessus :

- ☐ Un astérisque (*) peut être utilisé pour indiquer **toutes les valeurs valides. (avec respect des restrictions des autres valeurs).**
- ☐ Un trait d'union (-) placé entre deux nombres entiers indique **une gamme de nombres entiers.** Par exemple **1-4** correspond aux nombres entiers **1, 2, 3, 4.**
- ☐ Une liste de valeurs séparées par des virgules (,) **correspond à une liste.** Par exemple, **3, 4, 6, 8-12** correspond aux valeurs : **3, 4, 6, 8, 9, 10, 11, 12.**

2.1- Configuration d'une tâche Cron

Pour les valeurs ci-dessus :

- ☐ La barre oblique (/) peut être utilisée pour spécifier des **valeurs échelonnées**. Pour passer un nombre entier dans une gamme, faites-la suivre de /<nombre entier>.
- ☐ Par exemple, **0-59/2** permet de définir **une minute sur deux dans le champ des minutes (0,2,4,...58)**. Ces valeurs échelonnées peuvent également être utilisées avec un astérisque.
 - ***/3** peut être utilisée dans le champ des mois pour passer **un mois sur trois**.
 - ***/10** --> **valeurs 0, 10, 20, 30, 40, etc.**
 - **1/2** --> **valeurs 1, 3, 5, 7, 9, 11, etc.**
- ☐ Les lignes commençant par un signe dièse (#) correspondent à des **commentaires** et ne sont pas traitées.

2.2- Cron Table Système

- ❑ En plus des cron tables utilisateur, il existe la notion de **crontab système**. Celle-ci **permet de planifier des tâches d'administration**.
- ❑ La **différence** avec la **cron table de root**, qui permet d'obtenir les mêmes résultats, **est la possibilité de spécifier** dans celle-ci **le nom de l'utilisateur** sous lequel **doit exécuter la tâche**.
- ❑ Le fichier **crontab système** est **/etc/crontab**. Sa **syntaxe est la même** que les **cron tables utilisateur**, avec la **possibilité de spécifier le nom de l'utilisateur** juste **avant la ligne** de commandes à lancer.

2.2- Cron Table Système

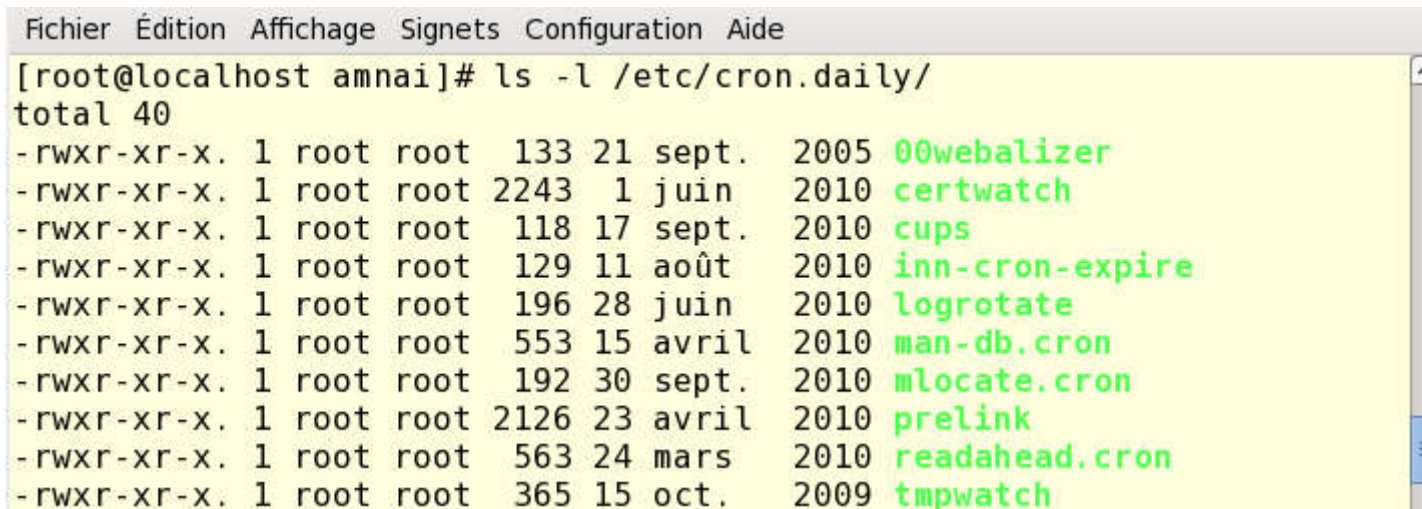
- ❑ La commande **run-parts** exécute tous les scripts et programmes présents dans le répertoire passé en **argument**.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
SHELL=/bin/bash  
PATH=/sbin:/bin:/usr/sbin:/usr/bin  
MAILTO=root  
HOME=/  
  
# run-parts  
01 * * * * root run-parts /etc/cron.hourly  
02 4 * * * root run-parts /etc/cron.daily  
22 4 * * 0 root run-parts /etc/cron.weekly
```

2.2- Cron Table Système

Parmi les programmes exécutés, remarquez **logrotate** qui permet d'effectuer des sauvegardes et de renommer des fichiers logs et des journaux du système afin que ceux ci ne deviennent pas inexploitable à cause de leur taille. Le programme **tmpwatch** est chargé de nettoyer le système des fichiers inutilisés (dans /tmp par exemple).



```
Fichier  Édition  Affichage  Signets  Configuration  Aide
[root@localhost amnai]# ls -l /etc/cron.daily/
total 40
-rwxr-xr-x. 1 root root 133 21 sept. 2005 00webalizer
-rwxr-xr-x. 1 root root 2243 1 juin 2010 certwatch
-rwxr-xr-x. 1 root root 118 17 sept. 2010 cups
-rwxr-xr-x. 1 root root 129 11 août 2010 inn-cron-expire
-rwxr-xr-x. 1 root root 196 28 juin 2010 logrotate
-rwxr-xr-x. 1 root root 553 15 avril 2010 man-db.cron
-rwxr-xr-x. 1 root root 192 30 sept. 2010 mlocate.cron
-rwxr-xr-x. 1 root root 2126 23 avril 2010 prelink
-rwxr-xr-x. 1 root root 563 24 mars 2010 readahead.cron
-rwxr-xr-x. 1 root root 365 15 oct. 2009 tmpwatch
```

2.3- Création D'un Crontab

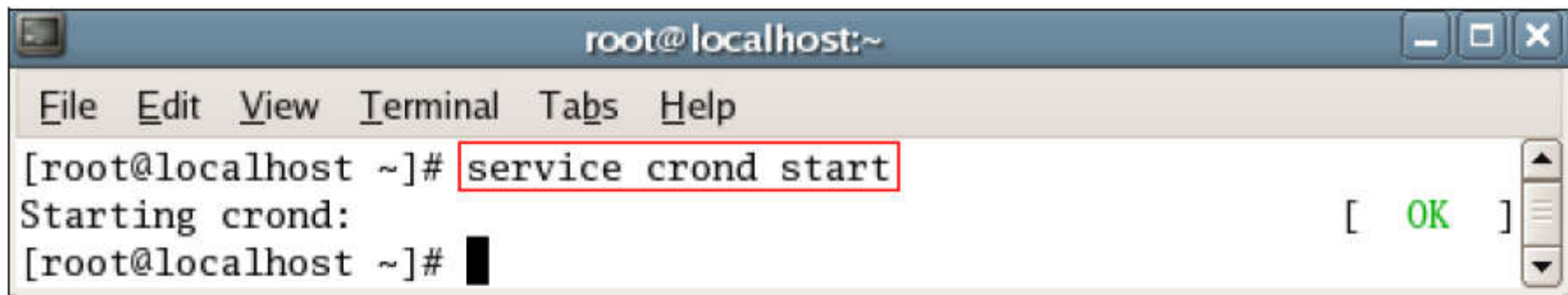
- ❑ Les usagers peuvent configurer des tâches cron à l'aide de l'utilitaire **crontab**.
- ❑ Le fichier **crontab** de chaque usager est stocké dans le répertoire **/var/spool/cron** et porte le même nom que l'usager.
- ❑ Par exemple le fichier crontab de l'usager **user1** est **/var/spool/cron/user1**
- ❑ La commande suivante permet à l'usager **user1** d'éditer son fichier crontab :

```
crontab -e
```

- ❑ Le fichier **crontab** est édité à l'aide de l'éditeur déterminé par la variable d'environnement **VISUAL** ou **EDITOR**.
- ❑ Chaque utilisateur dispose de sa propre table **crontab**, et bien que celles-ci se trouvent dans **/var/spool/cron**, elles ne sont pas conçues pour être éditées directement.

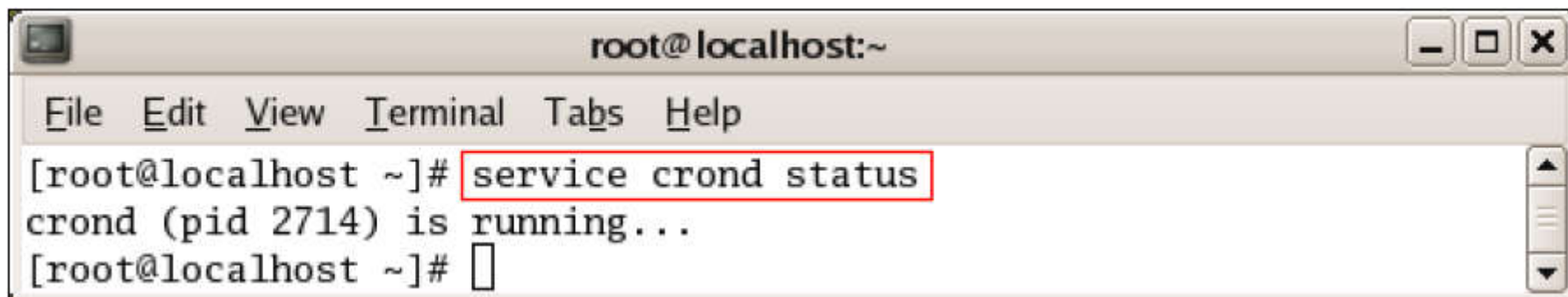
2.4- Lancement et Interruption du Service

- ❑ Pour lancer le service **cron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service crond start  
Starting crond: [ OK ]  
[root@localhost ~]#
```

- ❑ Pour vérifier le statut du service **cron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service crond status  
crond (pid 2714) is running...  
[root@localhost ~]#
```

2.4- Lancement et Interruption du Service

- ❑ Pour arrêter le service **cron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service crond stop  
Stopping crond: [ OK ]  
[root@localhost ~]#
```

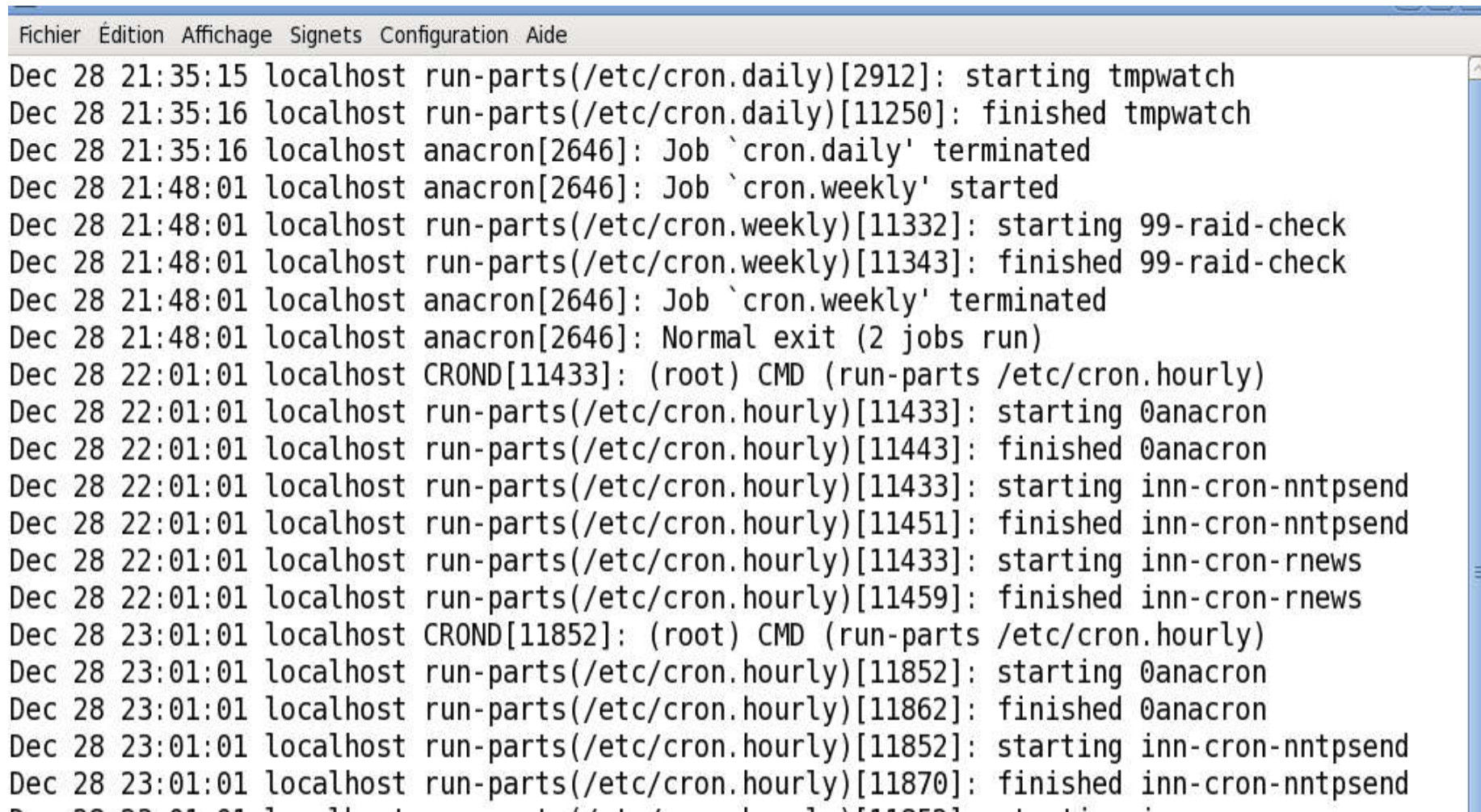
- ❑ Pour redémarrer le service **cron**, utilisez la commande



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service crond restart  
Stopping crond: [ OK ]  
Starting crond: [ OK ]  
[root@localhost ~]#
```


2.5- Journalisation

- ❑ Le fichier `/var/log/cron` est le fichier journal du cron.



The screenshot shows a terminal window with a menu bar at the top containing 'Fichier', 'Édition', 'Affichage', 'Signets', 'Configuration', and 'Aide'. The terminal displays a series of log entries from the cron daemon, each line representing a job execution. The logs include timestamps, the hostname 'localhost', the user 'root', and the specific command being executed. The jobs shown include 'tmpwatch', 'cron.daily', 'cron.weekly', '99-raid-check', and 'inn-cron-nntpsend'.

```
Fichier Édition Affichage Signets Configuration Aide
Dec 28 21:35:15 localhost run-parts(/etc/cron.daily)[2912]: starting tmpwatch
Dec 28 21:35:16 localhost run-parts(/etc/cron.daily)[11250]: finished tmpwatch
Dec 28 21:35:16 localhost anacron[2646]: Job `cron.daily' terminated
Dec 28 21:48:01 localhost anacron[2646]: Job `cron.weekly' started
Dec 28 21:48:01 localhost run-parts(/etc/cron.weekly)[11332]: starting 99-raid-check
Dec 28 21:48:01 localhost run-parts(/etc/cron.weekly)[11343]: finished 99-raid-check
Dec 28 21:48:01 localhost anacron[2646]: Job `cron.weekly' terminated
Dec 28 21:48:01 localhost anacron[2646]: Normal exit (2 jobs run)
Dec 28 22:01:01 localhost CROND[11433]: (root) CMD (run-parts /etc/cron.hourly)
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11433]: starting 0anacron
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11443]: finished 0anacron
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11433]: starting inn-cron-nntpsend
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11451]: finished inn-cron-nntpsend
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11433]: starting inn-cron-rnews
Dec 28 22:01:01 localhost run-parts(/etc/cron.hourly)[11459]: finished inn-cron-rnews
Dec 28 23:01:01 localhost CROND[11852]: (root) CMD (run-parts /etc/cron.hourly)
Dec 28 23:01:01 localhost run-parts(/etc/cron.hourly)[11852]: starting 0anacron
Dec 28 23:01:01 localhost run-parts(/etc/cron.hourly)[11862]: finished 0anacron
Dec 28 23:01:01 localhost run-parts(/etc/cron.hourly)[11852]: starting inn-cron-nntpsend
Dec 28 23:01:01 localhost run-parts(/etc/cron.hourly)[11870]: finished inn-cron-nntpsend
```

2.6- Sécurité

- ❑ Si le fichier **/etc/cron.allow** existe, alors vous devez être mentionnés dans celui-ci pour pouvoir utiliser le cron même si vous êtes root.
- ❑ Si le fichier **/etc/cron.allow** n'existe pas, mais que le fichier **/etc/cron.deny** existe, alors vous ne devez pas être mentionnés dans celui-ci, si vous désirez utiliser le cron.
- ❑ Si les deux fichiers existent, seul **/etc/cron.allow** sera considéré.

2.7- Commande crontab

- ❑ L'option **-l** permet **d'imprimer** la table **crontab en cours** sur la sortie standard

```
# crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.15383 installed on Fri Nov 19 13:12:47 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
* * * * * touch hakim
* * * * * touch /root/$$
```

- ❑ Si l'option **-u** est indiquée, elle **permet de préciser le nom de l'utilisateur** dont la crontab doit être manipulée.

```
# crontab -lu hakim
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.4590 installed on Wed Nov 24 17:39:15 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
```

2.7- Commande crontab

- ☐ L'option **-r** supprime la table crontab en cours.

```
# crontab -r
```

- ☐ L'option **-e** permet d'éditer la table en cours, une fois que vous sortez de l'éditeur, la table modifiée sera installée automatiquement.

```
# crontab -e
```

2.7- Exemples crontab

- ❑ Lister le contenu de /usr dans le fichier /home/amnai/usr.txt tous les jours à midi:

```
0 12 * * * ls /usr > /home/amnai/usr.txt
```

- ❑ Exécuter un script de nettoyage des logs tous les lundi à 10 h 15 :

```
15 10 * * 1 /sbin/supercleaner
```

- ❑ Faire exécuter du script backup tout les lundi à 11h30.

```
30 11 * * 1 /root/backup
```

- ❑ Exécuter un script à tous les jours à minuit

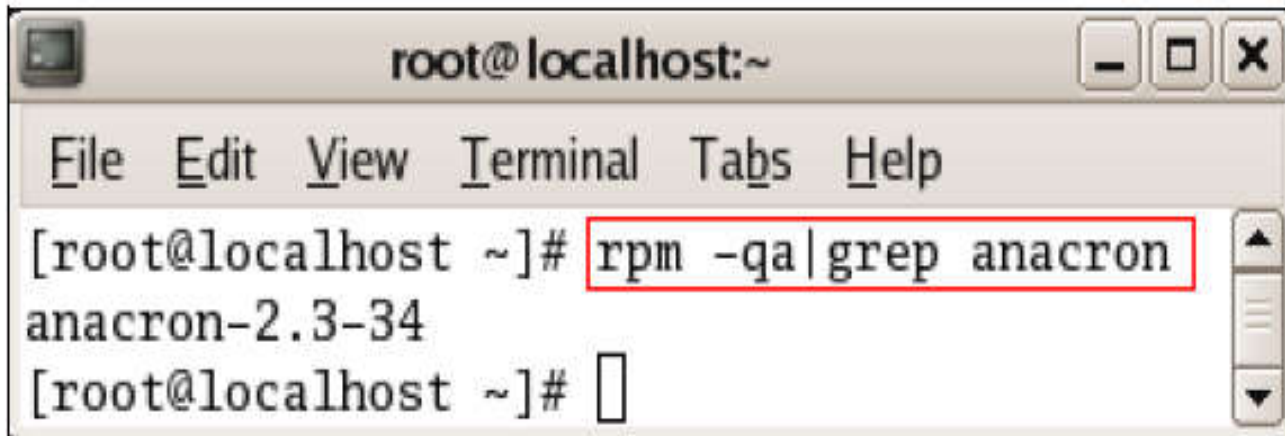
```
0 0 * * * /home/amnai/scripts/script_a_executer.sh
```

3- ANACRON

- ❑ Les tâches devant être exécutées périodiquement sont planifiées aux heures creuses de l'activité, généralement la nuit. Ces machines peuvent être des postes de travail, donc être éteintes la nuit.
- ❑ anacron exécute les tâches planifiées qui n'ont pas été effectuées depuis un certain temps. En lançant cet outil au démarrage du système, on évite que certaines tâches ne soient oubliées à cause d'un arrêt de la machine.
- ❑ anacron est un planificateur de tâches similaire à cron, à la différence près qu'il ne nécessite pas l'exécution continue du système. anacron peut être utilisé pour l'exécution quotidienne, hebdomadaire et mensuelle des tâches généralement exécutées par cron.

3- ANACRON

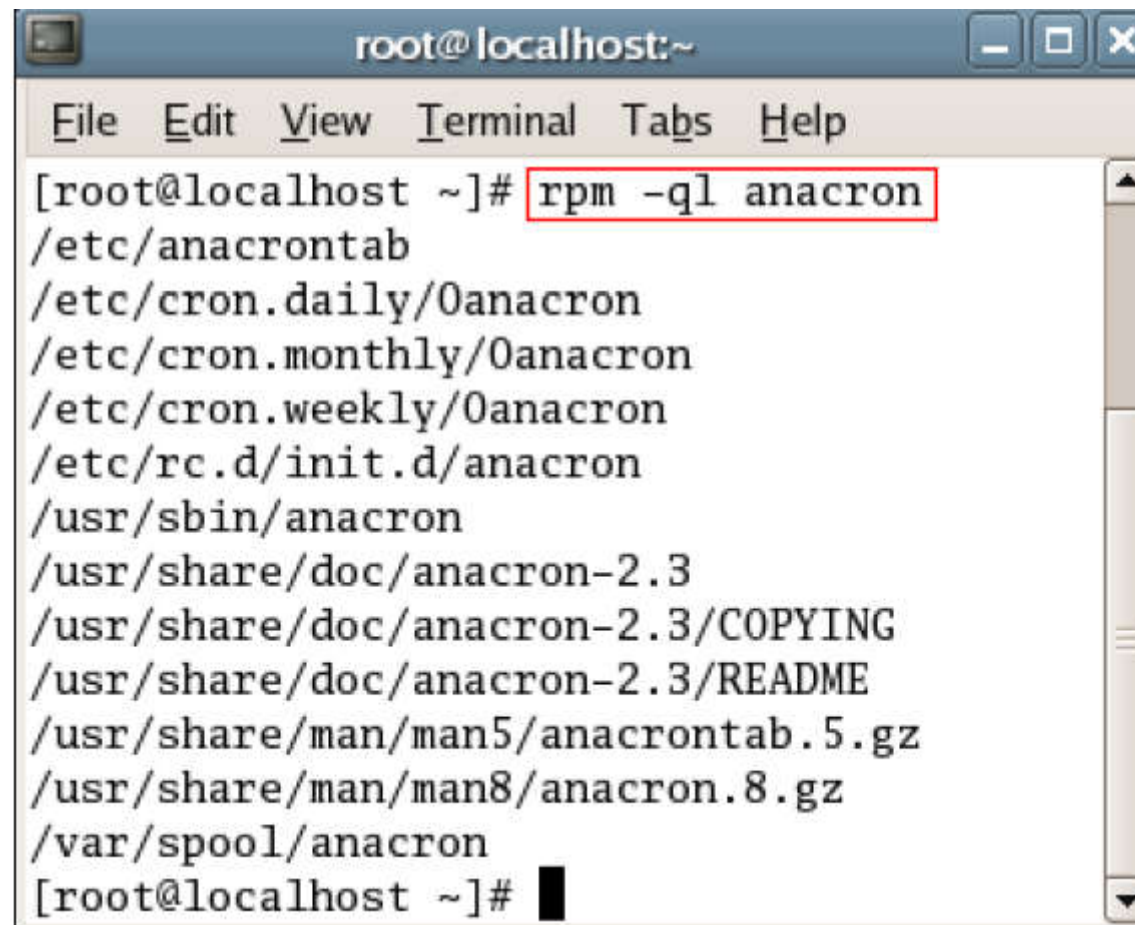
- ❑ Pour pouvoir utiliser le service **anacron**, le **paquetage RPM anacron** doit être installé. Pour savoir si le paquetage est installé, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# rpm -qa | grep anacron  
anacron-2.3-34  
[root@localhost ~]#
```

3- ANACRON

❑ Contenu du paquetage anacron



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# rpm -ql anacron  
/etc/anacrontab  
/etc/cron.daily/0anacron  
/etc/cron.monthly/0anacron  
/etc/cron.weekly/0anacron  
/etc/rc.d/init.d/anacron  
/usr/sbin/anacron  
/usr/share/doc/anacron-2.3  
/usr/share/doc/anacron-2.3/COPYING  
/usr/share/doc/anacron-2.3/README  
/usr/share/man/man5/anacrontab.5.gz  
/usr/share/man/man8/anacron.8.gz  
/var/spool/anacron  
[root@localhost ~]#
```

3.1- Fichier De Configuration

- ❑ Les tâches **anacron** **sont répertoriées** dans le fichier de configuration **/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 entries are useful for a Red Hat Linux system.
1      5      cron.daily      run-parts/etc/cron.daily
7      10     cron.weekly   run-parts/etc/cron.weekly
30     15     cron.monthly  run-parts/etc/cron.monthly
```

- ❑ Chaque ligne du fichier de configuration correspond à une tâche sous le format:

Période	Delai	job-id	commande
---------	-------	--------	----------

3.1- Fichier De Configuration

- ❑ **Période**, durée minimale en jours entre deux exécution de la tâche.
- ❑ **Delai**, durée en minutes avant le lancement du travail au redémarrage de la machine ; on spécifie des délais différents pour chaque tâche afin de ne pas alourdir le démarrage du système.
- ❑ **job-id**, chaîne de caractères ne contenant ni espace, slash '/', permettant d'identifier de façon unique le travail.
- ❑ **Commande**, commande permettant de lancer la tâche.

Période	Delai	job-id	commande
---------	-------	--------	----------

3.1- Fichier De Configuration

- ❑ Pour chaque tâche, **anacron** détermine si la tâche **a été exécutée** dans la période spécifiée dans **le champ période** du fichier de configuration.
- ❑ Si ce n'est pas le cas, **anacron exécute** la commande spécifiée dans le **champ commande** après **avoir respecté** le délai d'attente spécifié dans **le champ delai**.
- ❑ Une fois la tâche terminée, **anacron enregistre la date dans un fichier daté** dans le répertoire **/var/spool/anacron**.
- ❑ Seule **la date est utilisée** (et pas l'heure) **et la valeur de job-id est utilisée** comme nom de fichier pour le fichier daté.

```
# more cron.weekly  
20041118
```

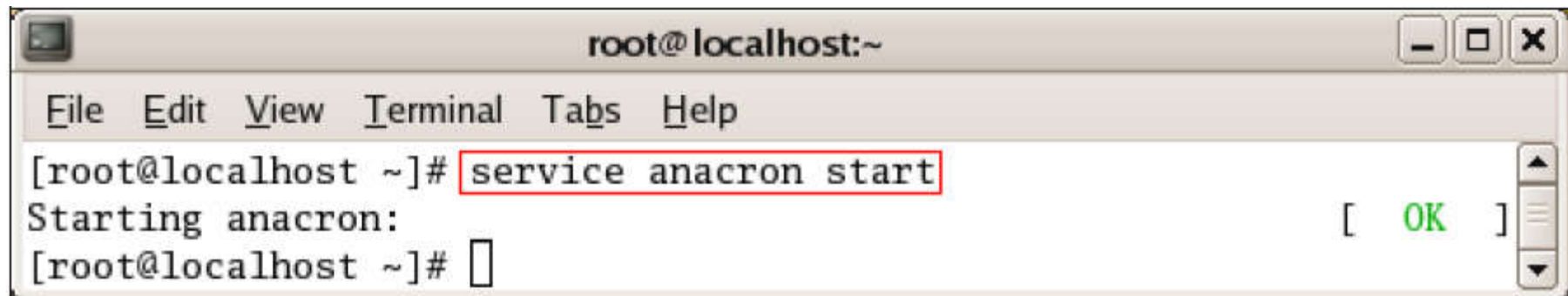
3.2- Journalisation

- ❑ Le fichier **/var/log/cron** est le fichier journal du service anacron.

```
Nov 25 16:43:05 ip-172-19-115-11 anacron[858]: Anacron 2.3 started on 2004-11-25  
Nov 25 16:43:06 ip-172-19-115-11 anacron[858]: Will run job `cron.weekly' in 10 min.
```

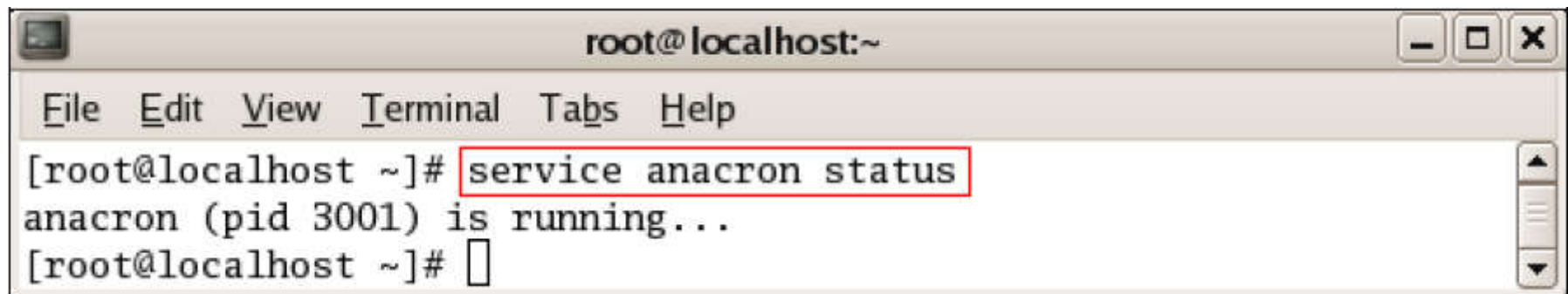
3.3- Lancement Et Interruption Du Service

- ❑ Pour **lancer** le service **anacron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service anacron start  
Starting anacron: [ OK ]  
[root@localhost ~]#
```

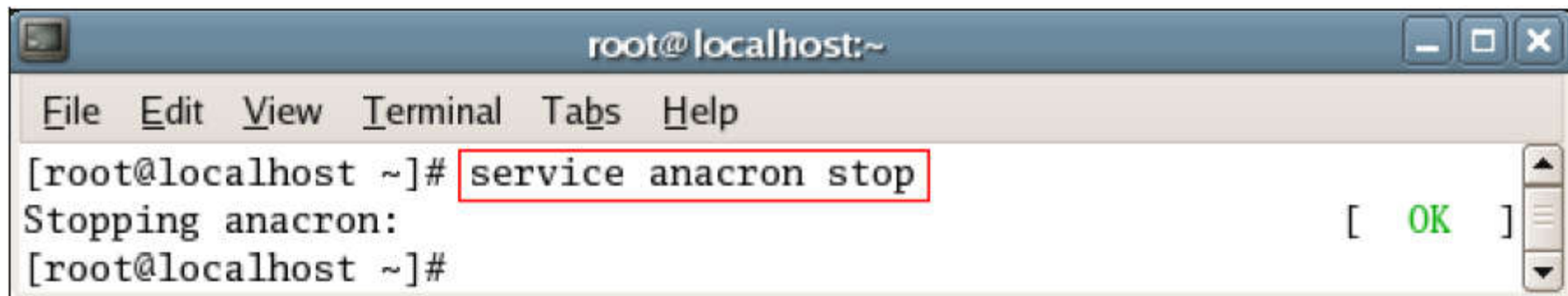
- ❑ Pour vérifier le **statut** du service **anacron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service anacron status  
anacron (pid 3001) is running...  
[root@localhost ~]#
```

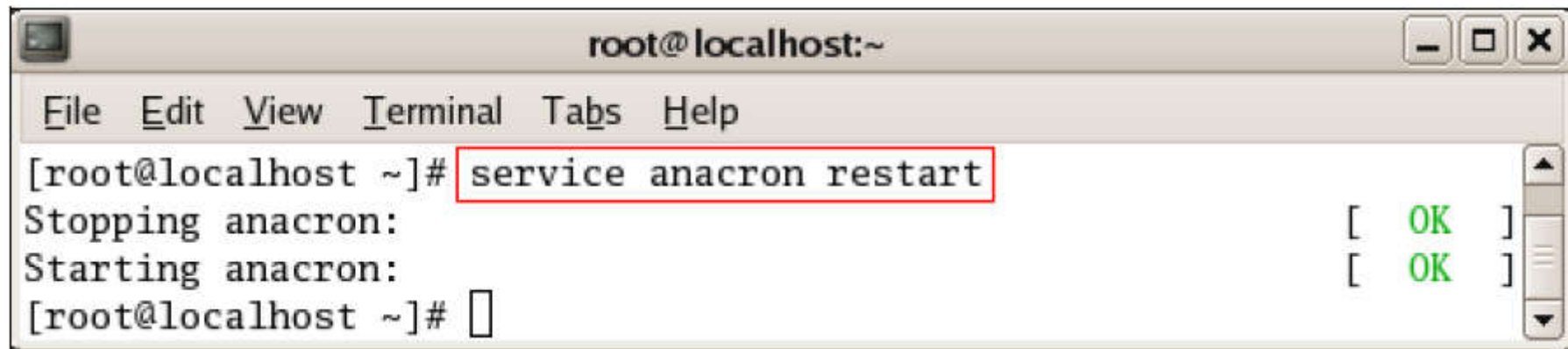
3.3- Lancement Et Interruption Du Service

- ❑ Pour **arrêter** le service **anacron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service anacron stop  
Stopping anacron: [ OK ]  
[root@localhost ~]#
```

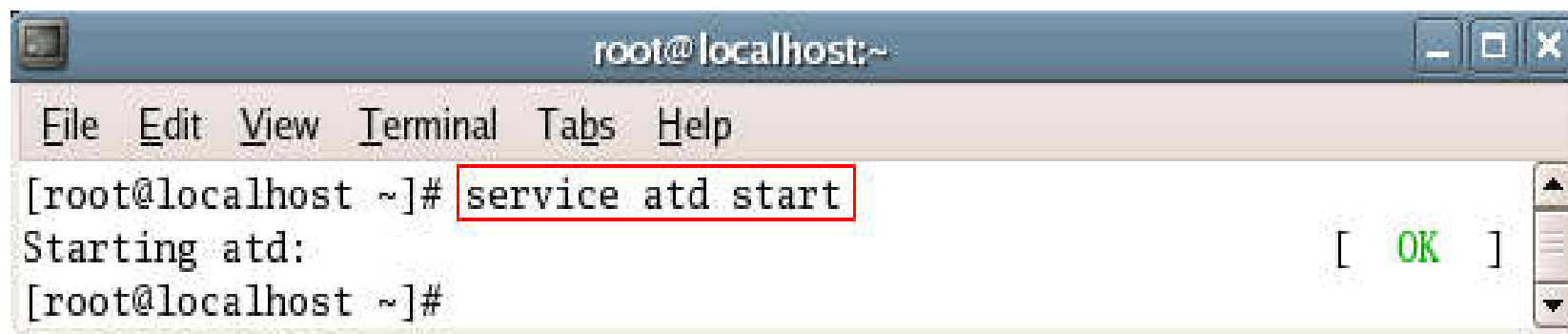
- ❑ Pour **redémarrer** le **statut** du service **anacron**, utilisez la commande :



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service anacron restart  
Stopping anacron: [ OK ]  
Starting anacron: [ OK ]  
[root@localhost ~]#
```

4.- AT et ATD

- ❑ La commande **at** sert uniquement à **planifier** des tâches ponctuelles ne devant être exécutées **qu'une seule fois à une date donnée**.
- ❑ Un démon **atd** est lancé **au démarrage** du système et gère **une file d'attente de travaux à exécuter** comme un spoule d'imprimante.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service atd start  
Starting atd: [ OK ]  
[root@localhost ~]#
```

4.- AT et ATD

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service atd status  
atd (pid 2489) is running...  
[root@localhost ~]#
```

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service atd restart  
Stopping atd: [ OK ]  
Starting atd: [ OK ]  
[root@localhost ~]#
```

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service atd stop  
Stopping atd: [ OK ]  
[root@localhost ~]#
```

4.- AT et ATD

- ❑ **at** *options1* heure date + incrément
- ❑ **at** *options2* tâches

Exécute des commandes entrées sur le canal d'entrée standard à une **heure précise** et, éventuellement, à une certaine **date**. La fin de l'entrée est marquée par **EOF**. **L'heure** peut être écrite sous forme numérique (avec accessoirement les minutes et les codes horaires) ou à l'aide d'un mot clé. La **date** peut être de la forme mois/jour ou un jour de la semaine ou un mot clé spécial.

4.- AT et ATD

❑ **at** *options1* heure date + incrément

❑ **at** *options2* tâches

Option1

- f **fichier** Les commandes contenues dans le fichier.
- m Envoie un email à l'utilisateur une fois la tâche effectuée.

Option2

- l Affiche la liste de toutes les tâches (ou seulement les tâches précisées) par l'utilisateur. Voir aussi **atq**.
- r Retire les tâches précédemment programmées. Vous devez avoir des droits particuliers ou être propriétaire de la tâche. Utilisez **-l** d'abord pour connaître la liste des tâches programmées. Voir aussi **atrm**.

4.- AT et ATD

Heure

hh:mm code

- L'heure peut être un nombre à un ou deux chiffres (par défaut, l'horloge est supposée de 24 heures) ;
- Les minutes optionnelles peuvent être écrites sur un ou deux chiffres ;
- Les deux points peuvent être omis : le format est ***h***, ***hh*** ou ***hhmm*** c'est-à-dire que **5**, **5:30**, **0530** et **1945** sont des heures valides. Si les **codes** horaires **AM** ou **PM** sont ajoutés, l'horloge est alors de 12 heures.
- **midnight | noon | now | teatime** : (**minuit | midi | mnt | 16h**)
Ces mots clés peuvent être utilisés en remplacement d'une heure numérique.
now doit être suivi **d'incrément**.
(**now + x** minutes / hours / days / weeks : **x** minutes / heures / jours / semaines à partir de maintenant)

4.- AT et ATD

Date

mois nombre, année

- Le mois doit être écrit en toute lettre ou simplement à l'aide de son abréviation en trois lettres ;
- L'année s'écrit quant à elle à l'aide d'un nombre à quatre chiffres;
- Le jour doit être écrit en toute lettre ou simplement à l'aide de son abréviation en trois lettres.

4.- AT et ATD

now | tomorrow

Référence **le jour en cours ou le suivant**. Si la **date** n'est pas précisée, **at** programme la tâche au jour en cours si **l'heure** n'est pas déjà passée, sinon elle est programmée pour le lendemain.

Incrément

Ajoute un incrément numérique si vous voulez programmer le jour ou l'heure d'exécution d'une tâche relativement à la date actuelle. Le nombre doit être suivi d'un des mots clé suivants :

**minute, hour,
day,
Week,
month,
Year.**

4.1- Ajout D'un Travail

Pour ajouter un travail :

at [-f <script>] <date d'exécution>

Si l'option **-f** n'est pas spécifié, **at** attend le script à exécuter directement sur son entrée standard et doit être terminé par les touches **[CTRL-D]**

```
# ls /tmp
```

```
scrollkeeper-tempfile.0 scrollkeeper-tempfile.1
```

```
# at 1305 August 3
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
```

```
at> /bin/rm /tmp/*
```

```
at> <EOT>
```

```
job 1 at 2008-08-03 13:05
```

4.1- Ajout D'un Travail

```
# ls /tmp#
```

```
# at 1:05pm Aug 3
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
```

```
at> /bin/rm/tmp/*
```

```
at> <EOT>
```

```
job 2 at 2008-08-03 13:05
```

4.1- Ajout D'un Travail

```
# at 11 am Friday
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
```

```
at> reboot
```

```
at> <EOT>
```

```
job 3 at 2008-12-03 11:00
```

```
# atq
```

```
3 2008-12-03 11:00 a root
```

```
# date
```

```
Fri Nov 26 15:32:56 EST 2008
```

4.1- Ajout D'un Travail

```
# at now + 7 hours
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
```

```
at> shutdown -h now
```

```
at> <EOT>
```

```
job 4 at 2008-11-26 22:41
```

```
# atq
```

```
3 2008-12-08 11:00 a root
```

```
4 2008-11-26 22:41 a root
```

4.1- Ajout D'un Travail

```
# at noon next month
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
```

```
at> reboot
```

```
at> <EOT>
```

```
job 5 at 2008-12-26 12:00
```

```
# atq
```

```
3  2008-12-03 11:00 a root
```

```
4  2008-11-26 22:41 a root
```

```
5  2008-12-26 12:00 a root
```


4.2- Contrôle Des Travaux

atq

- Affiche les tâches créées par la commande **at** qui sont encore **en attente**. Habituellement les tâches sont affichées dans l'ordre où elles vont être exécutées.
- Si aucun utilisateur n'est précisé où que vous n'avez pas de droit particulier, seules vos tâches sont listées.
- Pour chaque tâche créée par la commande **at**, **un fichier contenant le code de la tâche ainsi que le contexte** (variables d'environnement) est créé dans le répertoire **/var/spool/at**.

atrm

- Permet de supprimer les jobs créés par la commande **at**.
- **atrm** est **équivalente** à la commande '**at -d**'

4.2- Contrôle Des Travaux

batch

Il existe aussi la commande **batch** qui ne prend pas d'heure. Elle exécute la commande dès que la charge de la machine l'autorise. L'heure peut être précisée, dans ce cas elle sera considérée comme « à partir de cet heure, dès que possible ».

```
# batch
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c)  
/bin/sh
```

```
at> locate -u
```

```
at> <EOT>
```

```
job 6 at 2008-11-26 15:52
```

4.2- Sécurité

- ☐ Si le fichier **/etc/at.allow** existe, alors vous devez être mentionnés dans celui-ci pour pouvoir utiliser la commande **at**.
- ☐ Si le fichier **/etc/at.allow** n'existe pas, mais que le fichier **/etc/at.deny** existe, alors vous ne devez pas être mentionnés dans celui-ci, si vous désirez utiliser commande **at**.
- ☐ Si les deux fichiers existent, seul **/etc/at.allow** sera considéré.

4.2- Sorties Et Erreurs

Les sorties et erreurs de la commande exécutée par **at** sont renvoyées par mail à l'utilisateur, à moins que celui-ci les ait explicitement redirigées.