

# Préparation LPI

## Exam 101

### 101.3. Gestion des niveaux d'exécution

## Sommaire

- Niveaux d'exécution
- /etc/inittab
- Service, chkconfig
- Runlevel, reboot, poweroff, shutdown

- Poids : 3
- D finir le niveau d'ex cution par d faut
- Changer dynamiquement les niveaux d'ex cution
- Arr t et red marrage   la ligne de commande
- Avertissement des utilisateurs avant changements
- Arr t « propre » des processus

- **Le noyau :**
  - détecte le matériel
  - monte les systèmes de fichier
  - lance le programme `/sbin/init` qui lance les processus d'initialisation du système
- **init :**
  - seul processus à être lancé par le noyau
  - Lancé avec le niveau d'exécution en paramètre
  - Processus numéro 1.

```
# ps -aux | grep init
root      1  0.0  0.2  2664  436 ?        S      08:29   0:01 init
[5]
.....
```

- **init** : père de tous les autres processus
- Il gère le démarrage des autres processus, sessions de login et l'arrêt du système
- Le démarrage (ou l'arrêt) des processus est fonction du **niveau d'exécution** (*runlevels*) :
  - 0 : arrêt du système
  - 1 : mode mono-utilisateur
  - 2 : mode multi-utilisateur sans support réseau
  - 3 : mode multi-utilisateur (pas de mode graphique X11)
  - 4 : idem niveau 3 modifiable par l'utilisateur
  - 5 : mode multi-utilisateur avec démarrage de la couche graphique X11
  - 6 : mode reboot (idem niveau 0 mais redémarrage à chaud)

- **/etc/inittab** : fichier d'initialisation d'init

```
# more /etc/inittab
...
id:5:initdefault: (1)
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit (2)

10:0:wait:/etc/rc.d/rc 0 (3)
11:1:wait:/etc/rc.d/rc 1 (3)
12:2:wait:/etc/rc.d/rc 2 (3)
13:3:wait:/etc/rc.d/rc 3 (3)
14:4:wait:/etc/rc.d/rc 4 (3)
15:5:wait:/etc/rc.d/rc 5 (3)
16:6:wait:/etc/rc.d/rc 6 (3)

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1 (4)
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

- (1) précise quel niveau est utilisé pour démarrer le système
- (2) programme à exécuter dès le début du démarrage, avant qu'un niveau soit défini
- (3) En fonction du niveau, exécute le programme `rc` avec le numéro de niveau en paramètre
  - en niveau 5 : `init` lance la commande `rc 5`
- (4) Mise en place des consoles virtuelles (consoles de sessions textuelles de `[Ctrl]+[Alt]+[F1]` à `[Ctrl]+[Alt]+[F6]`)

- Permet également de gérer la séquence [Ctrl]+[Alt]+[Del]
- Permet de lancer la version de xdm choisie en runlevel 5

```
# Trap CTRL-ALT-DELETE  
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

```
# Run xdm in runlevel 5  
x:5:respawn:/etc/X11/prefdm -nodaemon
```

- En fonction de chaque niveau, le programme `rc` va lancer des scripts de configuration du système
- Tous les scripts sont regroupés à partir du répertoire `/etc/rc.d`
  - `/etc/rc.d/rc` : programme général d'exécution des scripts en fonction des niveaux
  - `/etc/rc.d/init.d` : répertoire contenant tous les scripts de configuration des services
  - `/etc/rc.d/rc0.d` à `/etc/rc.d/rc6.d` : répertoires qui contiennent les scripts (en fait des liens les scripts réels situés dans `/etc/rc.d/init.d`) à lancer en fonction du niveau d'exécution
  - `/etc/rc.d/rc.sysinit` : script de configuration exécuté une fois au démarrage quelque soit le niveau d'exécution
  - `/etc/rc.d/rc.local` : script utilisable par l'utilisateur pour ajouter l'exécution de commandes ou script personnels (exécuté en dernier)

- Principe des répertoires rcx.d :
  - Chaque sous-répertoire contient des liens symboliques qui pointent vers les scripts réels situés dans **/etc/rc.d/init.d**
  - Chaque lien respecte le formalisme suivant
  - La première lettre du script commence soit par **S** (pour Start – scripts lancé à l'entrée du niveau exécution) soit par **K** (pour Kill – scripts lancés à la sortie du niveau d'exécution)
  - suivent 2 chiffres (de 00 à 99) destiné à donner un ordre d'exécution des scripts puis le nom du script réel (dans init.d)
  - La présence d'un lien détermine si le script réel (dans init.d) sera exécuté ou non pour le niveau d'exécution
  - Exemple : le script `/etc/rc.d/rc5.d/S10network` sera lancé en niveau d'exécution 5, au démarrage du système, avec l'option « start »

- Pour changer de niveau d'exécution
  - Utiliser la commande `telinit runlevel` ou `init runlevel`
  - `#telinit 3` : fait passer au niveau d'exécution 3
- Mode single user (niveau 1)
  - Dans ce mode : vous n'avez qu'une seule console textuelle et le réseau est inactif
  - ATTENTION : vous êtes directement sous l'identité de root...sans avoir à saisir de mot de passe !
  - mode à utiliser pour des opérations de maintenance (pb disque)...ou si vous avez oublié le mot de passe de root !
  - `#telinit 3` : si la machine est démarrée
  - Si la machine n'est pas encore démarrée, il faut intervenir au niveau du chargeur de démarrage (GRUB)

- Options de telinit
  - q ou Q : force telinit de relire le contenu du fichier /etc/inittab  
Utilise le niveau d'exécution par défaut
  - s ou S : passe en single mode

- Passage en niveau “single user” avec GRUB
  - Sélectionner la version du noyau sous lequel démarrer et taper « e » (edit)
  - Dans la liste qui apparaît, sélectionner la ligne qui commence par « kernel » et taper « e »
  - Se positionner à la fin de cette ligne et insérer un espace puis le mot « single »
  - Taper « Entrée » pour quitter le mode d'édition
  - Taper « b » pour démarrer en mode single user

- Passage en niveau “single user” avec LILO
  - Appuyer sur la touche Entr e pour  viter le boot automatique
  - Apr s le prompt « boot : » Saisir linux single
  - Remplacer “linux” par la directive d finie dans lilo.conf si besoin

- Pour contrôler l'exécution des services  
**# service nom\_service option**
  - le nom du service correspond au nom du script qui se trouve dans /etc/init.d
  - options usuelles :
    - **start** : démarre le service
    - **stop** : arrête le service
    - **restart** ou **reload** : équivalent à un stop puis à un start
    - **status** : indique si le service est actif

```
# service cups restart
Arrêt de cups :
Démarrage de cups :
```

[ OK ]

- Initialisation des services
  - **chkconfig** contrôle l'activation des services

```
# chkconfig --level 35 yum on
# chkconfig --list yum
yum    0:arrêt 1:arrêt 2:arrêt 3:marche 4:arrêt 5:marche
6:arrêt
```

- **chkconfig** permet d'intégrer de nouveaux services
- exemple :
  - #**chkconfig -add yum**
  - cette commande crée automatiquement le lien `/etc/rc.d/rc5.d/S50yum`

- `chkconfig` permet de
  - Ajouter un nouveau service   g rer
  - Supprimer l'ex cution d'un service
  - Lister l' tat des services   d marrer
  - Changer les informations de d marrage d'un service
  - V rifier les informations de d marrage
- Chaque service devant  tre g r  par `chkconfig` doit comporter au minimum deux lignes dans le script d'initialisation
  - Une contient les niveaux d'ex cution et les priorit s de d marrage et d'arr t
  - La seconde contient la description du service

```
#!/bin/bash
#
# yum      Update notification daemon
#
# Author:   Jeremy Katz <katzj@redhat.com>
#
# chkconfig: 345 97 03
#
# description: This is a daemon which periodically checks for updates \
# and can send notifications via mail, dbus or syslog.
# processname: yum-updatesd
# config: /etc/yum/yum-updatesd.conf
# pidfile: /var/run/yum-updatesd.pid
#

### BEGIN INIT INFO
# Provides: yum-updatesd
# Required-Start: $syslog $local_fs messagebus
# Required-Stop: $syslog $local_fs messagebus
# Default-Start: 2 3 4 5
```

- Options de `chkconfig`
  - `--level niveau` : spécifie le niveau d'exécution impacté
  - `--add service` : prise en charge d'un nouveau service
  - `--del service` : le service ne sera plus pris en charge par `chkconfig`
  - `--list service` : liste les services gérés par `chkconfig`
  - `on` : utilisé pour démarrer un service (par défaut niveau 3,4 et 5)
  - `off` : utilisé pour arrêter un service
  - `reset` : réinitialisation de la prise en charge du service en fonction des informations contenues dans le script `d'init`

```
[root@localhost ~]# chkconfig --list
```

```
ConsoleKit    0:arrêt 1:arrêt 2:marche    3:marche    4:marche
```

```
5:marche 6:arrêt
```

```
NetworkManager 0:arrêt 1:arrêt 2:arrêt 3:marche    4:marche
```

```
5:marche6:arrêt
```

```
NetworkManagerDispatcher 0:arrêt 1:arrêt 2:arrêt 3:marche
```

```
4:marche5:marche    6:arrêt
```

```
acpid        0:arrêt 1:arrêt 2:marche    3:marche    4:marche
```

```
....
```

```
services basés sur xinetd :
```

```
  chargen-dgram: arrêt
```

```
  chargen-stream: arrêt
```

```
  cvs:          arrêt
```

```
....
```

```
  echo-stream:  arrêt
```

```
  rsync:       marche
```

```
[root@localhost ~]# chkconfig --list yum-updatesd  
yum-updatesd 0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
```

```
[root@localhost ~]# chkconfig yum-updatesd off
```

```
[root@localhost ~]# chkconfig --list yum-updatesd  
yum-updatesd 0:arrêt 1:arrêt 2:arrêt 3:arrêt 4:arrêt 5:arrêt 6:arrêt
```

```
[root@localhost ~]# chkconfig --level 35 yum-updatesd on
```

```
[root@localhost ~]# chkconfig --list yum-updatesd  
yum-updatesd 0:arrêt 1:arrêt 2:arrêt 3:marche 4:arrêt 5:marche 6:arrêt
```

```
[root@localhost ~]# chkconfig yum-updatesd reset
```

```
[root@localhost ~]# chkconfig --list yum-updatesd  
yum-updatesd 0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
```

```
[root@localhost ~]# chkconfig --list yum-updatesd
yum-updatesd 0:arrêt 1:arrêt 2:arrêt 3:marche 4:arrêt 5:marche 6:arrêt
```

```
[root@localhost ~]# chkconfig --del yum-updatesd
```

```
[root@localhost ~]# chkconfig --list yum-updatesd
le service yum-updatesd prend en charge chkconfig, mais il n'est enregistré à aucun niveau
(exécutez « chkconfig --add yum-updatesd »)
```

```
[root@localhost ~]# chkconfig --add yum-updatesd
```

```
[root@localhost ~]# chkconfig --list yum-updatesd
yum-updatesd 0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
```

- `runlevel` permet de vérifier le niveau d'exécution actuel ainsi que le niveau d'exécution précédant
- `runlevel` récupère cette information depuis `/var/run/utmp`
- `runlevel` affiche le niveau précédant le niveau actuel ou affiche N sinon (pas de modification depuis le démarrage)

```
# runlevel  
N 3
```

- Indication d'un passage du runlevel 3 au runlevel 5

```
# runlevel  
3 5
```

- shutdown : permet de redémarrer ou arrêter le système  
# shutdown options délai message
- Intérêt de cette commande : délai possible avant exécution et avertissement des utilisateurs si nécessaire; login bloqué (en créant le fichier /etc/nologin)
- runlevel affiche le niveau précédant le niveau actuel ou affiche N sinon (pas de modification depuis le démarrage)
- Options
  - -k : avertissement sans réellement exécuter la commande
  - -r : reboot
  - -h : halt ou poweroff

- Options (suite)
  - -P : shutdown puis arrêt alimentation
  - -H : shutdown sans arrêt alimentation
  - now : shutdown immédiat
  - -f : pas de fsck au reboot
  - -F : force le fsck au reboot
  - -c : annule la procédure de shutdown si délai et si la procédure n'a pas démarré
  - -a : vérifie que l'utilisateur demandant le shutdown est présent dans /etc/shutdown.allow
- Heure du shutdown au format hh:mm ou +m correspondant à un retard de m minutes du shutdown
- Un message libre d'avertissement aux utilisateurs

- `halt` : demande d'arrêt, `poweroff` ou `reboot` du système  
# `shutdown options délai message`
- Même binaire pour les commandes `poweroff` et `reboot`
- Écrit cette information dans le fichier `/var/log/wtmp`
- Attention : tout utilisateur connecté à la console peut lancer les commandes `halt`, `reboot` ou `poweroff`
- Dans tous les autres runlevel autres que 0 ou 6, `halt` invoque `shutdown` avec les options `-h` ou `-r`

```
# ll /sbin/halt /sbin/poweroff /sbin/reboot
-rwxr-xr-x 1 root root 12308 Jan 21  2009 /sbin/halt
lrwxrwxrwx 1 root root    4 Jan 11 11:24 /sbin/poweroff -> halt
lrwxrwxrwx 1 root root    4 Jan 11 11:24 /sbin/reboot -> halt
```

