

## 0.1 Processus d'initialisation de Linux

### 0.1.1 1. Introduction :

Dans ce document, je vais détailler la configuration des actions qui vont s'enchaîner juste après le chargement du noyau en mémoire. En résumé, vous démarrez votre OS à partir d'un chargeur d'OS comme Lilo ou Grub, en sélectionnant un noyau parmi la liste qu'il vous propose. Le noyau sélectionné est ensuite chargé en mémoire et il démarre à son tour le processus "Init". "Init" est le père de tous les processus. Il est donc le seul processus lancé par le noyau après son chargement. Son rôle principal est de créer des processus fils à partir du script `/etc/inittab`. Juste après que "init" ait été invoqué par le noyau, l'entrée de type "initdefault" (dans le fichier `/etc/inittab`) détermine le "runlevel" qui doit être exécuté. Si cette entrée ne figure pas dans `inittab`, un "runlevel" doit être entré dans la console.

### 0.1.2 2. Configuration de `/etc/inittab` :

Le fichier `/etc/inittab` définit quels processus sont lancés au démarrage. Comme dit précédemment, "init" distingue plusieurs "runlevels", chacun ayant une fonction particulière. Une entrée dans le fichier "inittab" à le format suivant :

```
id : runlevels : action : process
```

- » **id** :
- » **runlevels** : liste des "runlevels" pour lesquels l'action spécifiée doit être exécutée.
- » **action** : le type d'action qui doit être accompli.
- » **process** : spécifie le processus à exécuter.

### 0.1.3 2.1 id :

Un id est une séquence unique de 1 à 4 caractères qui identifie (de manière unique donc..) une entrée dans le fichier `/etc/inittab`.

### 0.1.4 2.2 runlevels :

La liste des processus que "init" doit charger dépend du runlevel. Un "runlevel" est une configuration système qui autorise seulement un certain groupe de processus à exister. Les processus engendrés par "init" pour chacun de ces runlevels sont définis dans le fichier `/etc/inittab`. Un runlevel peut prendre les huit valeurs suivantes : 0 à 6 et S ou s. Un utilisateur ayant des droits suffisants, peut changer de "runlevel" en cours d'utilisation, en faisant appel à la commande "telinit" qui envoie un signal au processus "init". Quand le "runlevel" est changé, tout les processus en cours qui ne sont pas spécifiés dans le nouveau "runlevel", sont tués. Par convention, les "runlevels" 0, 1 et 6 sont réservés de la manière suivante /

- 0 : pour arrêter le système
- 1 : pour démarrer en mode "single user"
- 6 : pour redémarrer le système

Mais en réalité cela dépend de votre distribution. Pour plus de détails reportez-vous aux didacticiels spécifiques à votre distribution, dans la section "Distributions".

### 0.1.5 2.3 actions :

Cette partie sert à indiquer les paramètres de lancement du processus :

- » **respawn** : Le processus sera relancé chaque fois qu'il se terminera (e.g. getty).
- » **wait** : Le processus sera lancé une seule fois ; quand le runlevel est entré. "init" attend qu'il se termine pour continuer.
- » **once** : Le processus sera lancé une seule fois ; quand le runlevel choisit aura été entré.
- » **boot** : Le processus sera exécuté pendant le démarrage du système. La valeur du "runlevel" est ignoré (= marche qq soit le "runlevel").
- » **bootwait** : Le processus sera exécuté pendant le démarrage du système ; "init" attend qu'il se termine pour continuer (e.g. /etc/rc). La valeur du "runlevel" est ignoré (= marche qq soit le "runlevel").
- » **off** : Ca ne fait rien.
- » **ondemand** : Le processus est exécuté chaque fois que le "runlevel" "ondemand" est appelé. Cependant, il n'en découle pas un changement de "runlevel". (les "runlevels" "ondemand" sont 'a', 'b', et 'c').
- » **initdefault** : L'entrée "initdefault" spécifie quel "runlevel" doit être lancé après le chargement du noyau. Si cette entrée n'est pas présente, "init" demandera, dans une console, quel "runlevel" doit être lancé. La valeur du "runlevel" est ignoré (= marche qq soit le "runlevel").
- » **sysinit** : Le processus sera exécuté pendant le démarrage du système. Il sera exécuté avant même l'entrée 'boot' ou 'bootwait'. La valeur du "runlevel" est ignoré (= marche qq soit le "runlevel").
- » **ctrlaltdel** : Le processus sera exécuté quand "init" aura reçu le signal SIGINT. C'est à dire que quelqu'un a pressé la combinaison CTRL-ALT-DEL dans la console. En règle générale, le processus spécifié par l'utilisateur a pour but de redémarrer la machine.
- » **kbrequest** : Le processus est exécuté quand "init" reçoit un signal du gestionnaire de clavier, lui indiquant ainsi qu'une certaine combinaison de touches ont été pressés dans la console. Pour plus d'informations à ce sujet, allez regarder la documentation de 'kbd'.

### 0.1.6 2.4 processus :

Dans cette partie vous indiquez simplement le script ou la commande à exécuter avec l'option que vous souhaitez, si nécessaire.

### 0.1.7 3. La pratique par l'exemple :

Nous allons essayer d'analyser un exemple de fichier /etc/inittab, que l'on peut retrouver sur une machine tournant sous Debian Woody :

```
# Definit le runlevel utilisé au démarrage. C'est à dire
  3 dans le cas présent.
id :3 :initdefault :
# Cette ligne lance, avant tout les autres, le script
  'rcS' dans le répertoire '/etc/init.d/'.
si : :sysinit :/etc/init.d/rcS
```

```

# Exécuter la commande '/sbin/sulogin' en mode single-user.
~~ :S :wait :/sbin/sulogin
# Selon le "runlevel" qui a été déterminé, cela lance
  le script '/etc/init.d/rc' avec cette valeur pour argument.
  # Ce script a pour but dans lancer tout les processus
  contenus dans /etc/rcX.d ( X= valeur du "runlevel" ).
10 :0 :wait :/etc/init.d/rc 0
11 :1 :wait :/etc/init.d/rc 1
12 :2 :wait :/etc/init.d/rc 2
13 :3 :wait :/etc/init.d/rc 3
14 :4 :wait :/etc/init.d/rc 4
15 :5 :wait :/etc/init.d/rc 5
16 :6 :wait :/etc/init.d/rc 6
# La commande est relancé automatiquement à chaque fois
  qu'elle se termine.
z6 :6 :respawn :/sbin/sulogin
# La combinaison CTRL-ALT-DEL entraine l'arrêt du système
  ('/sbin/shutdown -t1 -a -r now').
ca :12345 :ctrlaltdel :/sbin/shutdown -t1 -a -r now
# On ouvre un certain nombre de terminaux ('/sbin/getty')
  selon le "runlevel". # Remarque : sur la plupart des
  systèmes, tty7 est utilisé par 'X'.
1 :2345 :respawn :/sbin/getty 38400 tty1
2 :23 :respawn :/sbin/getty 38400 tty2
3 :23 :respawn :/sbin/getty 38400 tty3
4 :23 :respawn :/sbin/getty 38400 tty4
5 :23 :respawn :/sbin/getty 38400 tty5
6 :23 :respawn :/sbin/getty 38400 tty6

```

Pour s'assurer que vous avez tout compris, nous allons prendre un exemple en particu-

	id	runlevels
lier :	12	2
	le numéro <b>unique</b> identifiant cette entrée dans le fichier inittab.	cette entrée serat interprété si et seulement si

Et voilà, vous devez maintenant être capable de modifier à souhait la configuration de démarrage de votre machine pour y ajouter des processus ou, au contraire, en supprimer pour rendre le démarrage plus rapide.