
init

Parent de tous les processus. Son rôle principal est de créer des processus depuis un script stocké dans `/etc/inittab`. L'ID du processus `init` est toujours `1`

runlevels

un **runlevel** est une configuration du système qui permet de sélectionner un groupe de processus. Il existe 8 runlevel. le runlevel peut être changé avec **telinit** qui envoie le signal approprié à `init`. Les runlevels **S**, **0**, et **6** sont réservés. Le runlevel **S** est utilisé pour initialiser le système au boot. Les runlevels **S** et **1** sont des modes simple utilisateur. Runlevel **0** est utilisé pour arrêter le système, runlevel **6** est utilisé pour rebooter le système. Après avoir booté au travers du runlevel **S**, le système entre dans un mode multi-utilisateurs (2 à 5). Les runlevels **7-9** sont aussi valides, bien qu'ils ne soient pas documentés.

Boot

Après qu'`init` ai été invoqué en tant que dernière étape de la séquence de boot du kernel, il cherche le fichier **`/etc/inittab`** pour voir s'il y'a une entrée de type **`initdefault`**, qui détermine le runlevel initial du système. S'il n'est pas spécifié, le runlevel doit être entré dans la console système.

En mode simple utilisateur, **`/sbin/sulogin`** est invoqué sur **`/dev/console`**

En entrant en mode simple utilisateur, `init` initialise les réglages des consoles `stty`.

En entrant en mode multi-utilisateur pour la première fois, **`init`** effectue les entrées **`boot`** et **`bootwait`** pour permettre aux systèmes de fichiers d'être montés avant que les utilisateurs puissent se logger. Ensuite toutes les entrées correspondant au runlevel sont traitées.

En démarrant un nouveau processus, **`init`** vérifie d'abord si le fichier **`/etc/initscript`** existe. Si c'est le cas, il utilise ce script pour démarrer le processus.

Chaque fois qu'un enfant se termine, **`init`** enregistre la raison dans **`/var/run/utmp`** et **`/var/log/wtmp`**

Changer les runlevels

Une fois tous les processus lancés, `init` attend qu'un de ses processus meurt, un signal **`powerfail`**, ou jusqu'à ce qu'il soit signalé par **`telinit`** de changer de runlevel. Quand une de ces 3 conditions se produit, il ré-examine **`/etc/inittab`**.

Si `init` n'est pas en mode simple utilisateur et reçoit un signal **`POWERFAIL (SIGPWR)`**, il lit le fichier **`/etc/powerstatus`**, et lance une commande basée sur le contenu de ce fichier. L'utilisation de **`SIGPWR`** et **`/etc/powerstatus`** n'est pas conseillé.

Quand `init` change de runlevel, il envoie un **`SIGTERM`** à tous les processus qui ne sont pas définis dans le nouveau runlevel. Puis il attend 5 secondes avant de forcer ces processus avec **`SIGKILL`**. Noter que `init` assume que tous ces processus (et leur descendant) restent dans le même groupe de processus qu'`init` a créé pour eux. Si un de ces processus change de groupe, il ne recevra pas ces signaux.

Variables d'environnement

`PATH` `/bin :/usr/bin :/sbin :/usr/sbin`

INIT_VERSION la version d'init

RUNLEVEL Le runlevel système courant

PREVLEVEL Le runlevel système précédant

CONSOLE La console système. S'il n'est pas défini par init, sera /dev/console par défaut.

Flags

Il est possible de passer des flags à init depuis le bootloader (ex : grub). Init accepte les flags suivant :

-s, S, single boot en mode simple utilisateur. Dans ce mode /etc/inittab est examiné et les scripts rc sont généralement lancés avant que le shell du mode simple utilisateur ne soit démarré.

1-5 Permet de spécifier le runlevel sur lequel booter.

-b, emergency Boot directement en mode simple utilisateur sans lancer de script de démarrage.

-a, auto LILO ajoute 'auto' à la ligne de commande si le kernel est booté avec la ligne de commande par défaut (sans l'intervention de l'utilisateur). init définit alors la variable AUTOBOOT à yes.

-z permet d'étendre un peu la ligne de commande, et donc prenne un peu plus d'espace dans la pile. Init peut ainsi manipuler la ligne de commande pour que ps(1) affiche le runlevel courant.

Interface

init écoute les messages sur un fifo dans **/dev/initctl**. Telinit l'utilise pour communiquer avec init.

Signaux

init réagit à certains signaux :

SIGHUP a le même effet que **telinit q**

SIGUSR1 init ferme et ré-ouvre son fifo **/dev/initctl**. utile pour les scripts de boot quand /dev est remonté.

SIGINT Généralement le kernel envoie ce signal à init quand **ctrl+alt+del** est pressé. Il active l'action **ctrlaltdel**.

SIGWINCH le kernel envoie ce signal quand la touche **KeyboardSignal** est pressée. Active l'action **kbrequest**