
inittab

Fichier de configuration de init

Une entrée de runlevel a le format suivant :

id :runlevels :action :process

id séquence unique de 1 à 4 caractères qui identifie l'entrée dans inittab.

runlevels liste les runlevels pour lesquels l'action spécifiée doit être exécutée

action décrit l'action à exécuter

process spécifie le processus à exécuter. S'il commence avec un '+', init ne créera pas de utmp et wtmp pour ce processus. nécessaire pour les getty qui insistent pour faire leur propre utmp/wtmp. c'est aussi un bug historique.

Le champ **runlevels** peut contenir plusieurs caractères pour différents runlevels. Les actions valides sont :

respawn Le processus sera redémarré s'il est terminé (ex getty)

wait Le processus sera démarré une fois en entrant dans le runlevel et init attendra qu'il se termine

once Le processus sera exécuté une fois en entrant dans le runlevel.

boot Le processus sera exécuté au boot Le champs runlevel est ignoré

bootwait Le processus sera exécuté durant le boot, init attendra qu'il se termine(ex : /etc/rc). Le runlevel est ignoré.

off Ne fait rien ondemand Sera exécuté quand le runlevel est appelé, cependant aucun changement de level ne sera exécuté.(les runlevels ondemand sont a, b, et c)

initdefault spécifie le runlevel par défaut après le boot. Le champs process est ignoré

sysinit Le processus sera exécuté durant le boot. Il sera exécuté avant une entrée boot ou bootwait. Le runlevel est ignoré.

powerwait Le processus est exécuté quand le courant s'affaiblit.

powerfail comme powerwait, excepté qu'init n'attend pas que le processus se termine

powerokwait Le processus sera exécuté une fois qu'init est informé que le courant est revenu

powerfailnow Le processus sera exécuté quand la batterie est vide et le courant s'arrête.

ctrlaltdel Le processus sera exécuté quand init recevra le signal SIGINT. généralement en pressant ctrl+alt+del

kbrequest Le processus sera exécuté quand init reçoit un signal depuis une combinaison de touches spécial.

Exemple simple

```
id:1:initdefault :
rc ::bootwait :/etc/rc
1:1:respawn :/etc/getty 9600 tty1
2:1:respawn :/etc/getty 9600 tty2
3:1:respawn :/etc/getty 9600 tty3
4:1:respawn :/etc/getty 9600 tty4
```

Il exécute /etc/rc durant le boot et lance getty sur tty1 à tty4

Exemple plus élaboré

```
id:2:initdefault :

# script d'initialisation/configuration au moment du boot
si ::sysinit :/etc/init.d/rcS

# quoi faire en mode simpleutilisateur
:S:wait :/sbin/sulogin

# /etc/init.d execute les scripts S et K pendant le changement de 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

# quoi faire si ctrl+alt+del sont pressés
ca ::ctrlaltdel :/sbin/shutdown -t1 -h now

# Runlevel 2,3 : getty sur les consoles virtuelles
# Runlevel 3 : getty sur le terminal (ttyS0) et le modem (ttyS1)
1:23:respawn :/sbin/getty tty1 VC linux
2:23:respawn :/sbin/getty tty2 VC linux
3:23:respawn :/sbin/getty tty3 VC linux
4:23:respawn :/sbin/getty tty4 VC linux
S0:3:respawn :/sbin/getty -L 9600 ttyS0 vt320
S1:3:respawn :/sbin/mgetty -x0 -D ttyS1
```