

INIT(8) Gestionnaire de systeme d'OpenBSD INIT(8)

## NOM

**init** - systeme de gestion d'initialisation

## RESUME

**init [-fs]**

## DESCRIPTION

Le programme **init** est la dernière étape du processus de boot. Ça exécute normalement la séquence des événements décrite dans [rc\(8\)](#) et si cela réussit, les opérations multi-utilisateur commencent. Si le script de boot échoue, **init** démarre les opérations en simple-utilisateur pour donner au superutilisateur un shell sur la console. Le programme **init** peut être passé en paramètre à partir du programme de boot pour prévenir le fait que le système passe en multiutilisateur et pour, à la place, exécuter un invité de commande simple utilisateur sans initialiser les démons normaux.

Les paramètres suivants peuvent être passés depuis le programme de démarrage :

**-f** Active le mode démarrage-éclair

**-s** Démarre directement dans le mode utilisateur simple.

The system is then quiescent for maintenance work and may later be made to go to multi-user by exiting the single-user shell (with `^D`). This causes **init** to run the `/etc/rc` startup command file in fastboot mode (skipping disk checks).

If the console entry in the [ttys\(5\)](#) file does not contain the ```secure` flag, then **init** will require that the superuser password be entered before the system will start a single-user shell. The password check is skipped if the console is marked as ```secure`. The kernel [securelevel\(7\)](#) is normally set to 0 while in single-user mode, and raised to 1 when the system begins multi-user operations. This action will not take place if the `securelevel` is -1, and can be modified via the `/etc/rc.securelevel` script.

In multi-user operation, **init** maintains processes for the terminal ports found in the file [ttys\(5\)](#). **init** reads this file, and executes the command found in the second field. This command is usually [getty\(8\)](#); **getty** opens and initializes the tty line and executes the login program. The login program, when a valid user logs in, executes a shell for that user. When this shell dies, either because the user logged out or an abnormal termination occurred (a signal), the **init** program wakes up, deletes the user from the [utmp\(5\)](#) file of current users and records the logout in the `wtmp` file. The cycle is then restarted by **init** executing a new **getty** for the line.

Line status (on, off, secure, getty, or window information) may be changed in the `ttys` file without a reboot by sending the signal `SIGHUP` to **init** with the command ```kill -s HUP 1`. On receipt of this signal, **init** re-reads the `ttys` file. When a line is turned off in `ttys`,

`init` will send a `SIGHUP` signal to the controlling process for the session associated with the line. For any lines that were previously turned off in the `ttys` file and are now on, `init` executes a new `getty` to enable a new login. If the `getty` or `window` field for a line is changed, the change takes effect at the end of the current login session (e.g., the next time `init` starts a process on the line). If a line is commented out or deleted from `ttys`, `init` will not do anything at all to that line. However, it will complain that the relationship between lines in the `ttys` file and records in the `utmp` file is out of sync, so this practice is not recommended. `init` will terminate multi-user operations and resume single-user mode if sent a terminate (`TERM`) signal, for example, `kill -s TERM 1`. If there are processes outstanding that are deadlocked (because of hardware or software failure), `init` will not wait for them all to die (which might take forever), but will time out after 30 seconds and print a warning message.

`init` will cease creating new `getty(8)` and allow the system to slowly die away, if it is sent a terminal stop (`TSTP`) signal, i.e., `kill -s TSTP 1`. A later hangup will resume full multi-user operations, or a terminate will start a single-user shell. This hook is used by `reboot(8)` and `halt(8)`. `init` will terminate multi-user operations, kill all `getty(8)`, run `/etc/rc.shutdown`, and halt the machine if user-defined signal 1 (`USR1`) or user-defined signal 2 is received. `/etc/rc.shutdown` can specify that a powerdown is requested. Alternatively, `USR2` specifically requests a powerdown. The role of `init` is so critical that if it dies, the system will reboot itself automatically. If, at bootstrap time, the `init` process cannot be located, the system will panic with the message `panic: init died (signal %d, exit %d)`.

## RESSOURCES

When `init` spawns a process it sets the process priority, `umask`, and resource limits based on `/etc/login.conf`. When starting the `rc(8)` files, the login class `daemon` is used. When starting a window system or `getty(8)`, the login class `default` is used. No resource changes are made when entering single user mode.

## FICHIERS

`/dev/console` system console device  
`/dev/tty*` terminal ports found in `ttys`  
`/etc/rc` system startup commands  
`/etc/rc.securelevel` commands that run before the security level changes  
`/etc/rc.shutdown` script run at shutdown time  
`/etc/ttys` terminal initialization information file  
`/fastboot` tells `rc(8)` not to run `fscck(8)` during the next boot  
`/var/run/utmp` record of users currently logged in  
`/var/log/wtmp` record of all logins and logouts

## DIAGNOSTICS

`getty` repeating too quickly on port %s, sleeping  
A process being started to service a line is exiting

quickly each time it is started. This is often caused by a ringing or noisy terminal line. Init will sleep for 10 seconds, then continue trying to start the process.

some processes would not die; ps axl advised. A process is hung and could not be killed when the system was shutting down. This condition is usually caused by a process that is stuck in a device driver because of a persistent device error condition.

## VOIR AUSSI

[kill\(1\)](#), [login\(1\)](#), [sh\(1\)](#), [fbtab\(5\)](#), [login.conf\(5\)](#), [ttys\(5\)](#), [securelevel\(7\)](#), [crash\(8\)](#), [getty\(8\)](#), [halt\(8\)](#), [rc\(8\)](#), [rc.shutdown\(8\)](#), [reboot\(8\)](#), [shutdown\(8\)](#)

## HISTORIQUE

Une commande init est apparue dans la version 6 de l'UNIX d'AT&T.

OpenBSD 4.4 24 août 2008 2

From:

<http://www.mouet-mouet.net/> - mouet-mouet !

Permanent link:

<http://www.mouet-mouet.net/doku.php?id=maxime:openbsd:manpages-fr:8:init&rev=1220214307>

Last update: **2021/10/08 00:16**

