

NAME

inittab — script for the init process

DESCRIPTION

The *inittab* file supplies the script to *init*'s role as a general process dispatcher. The process that constitutes the majority of *init*'s process dispatching activities is the line process */etc/getty* which initiates individual terminal lines. Other processes typically dispatched by *init* are daemons and the *shell*.

The lines file is composed of entries that are position dependent and have the following format:

id:rstate:action:process

Each entry is delimited by a newline, however a backslash (\) preceding a newline indicates a continuation of the entry. Up to 512 characters per entry are permitted. Comments may be inserted in the *process* field using the *sh* convention for comments. (See *sh(1)*) Comments for lines which spawn *gettys* are displayed by the *who* command. It is expected that they will contain some information about the line such as the location. There are no limits (other than maximum entry size) imposed on the number of entries within the *inittab* file. The entry fields are:

- id* This is one or two characters used to uniquely identify an entry.
- rstate* This defines the *run state* in which this entry is to be processed. Run states effectively correspond to a configuration of processes in the system. That is, each process spawned by *init* is assigned a run state or run states in which it is allowed to exist. The *run states* are represented by a number ranging from 0 through 6. As an example, if the system is in *run state* 1, only those entries having a 1 in the *run state* field will be processed. When *init* is requested to change *run states*, all processes which do not have an entry in the *rstate* field for the target *run state* will be sent the warning signal and allowed a 20 second grace period before being forcibly terminated by a kill signal. The *rstate* field can define multiple *run states* for a process by selecting more than one run state in any combination from 0 through 6. If no *run state* is specified, then the process is assumed to be valid at *all run states* 0-6. There are three other values *a*, *b*, and *c* which can appear in the *rstate* field even though they are not true *run states*. Entries which have these characters in the *rstate* field are processed only when the *telinit* process requests them to be run (regardless of the current *run state* of the system). They differ from *run states* in that *init* can never enter *run state* *a*, *b* or *c*. Also, a request for the execution of any of these processes does not change the current *run state*. Furthermore, a process started by an *a*, *b* or *c* command is not killed when *init* changes levels. They are only killed if their line in */etc/inittab* is marked off in the *action* field, their line is deleted entirely from */etc/inittab*, or *init* goes into the *SINGLE USER* state.
- action* Key words in this field tell *init* how to treat the process specified in the *process* field. The actions recognized by *init* are as follows:
- respawn** If the process does not exist then start the process, do not wait for its termination (continue scanning the *inittab* file), and when it dies restart the process. If the process currently exists then do nothing and continue scanning the *inittab* file.
 - wait** Upon *init*'s entering the run state that matches the entry's *rstate*, start the process and wait for its termination. All subsequent reads of the *inittab* file while *init* is in the same run state will cause *init* to ignore this entry.
 - once** Upon *init*'s entering a run state that matches the entry's *rstate*, start the process, do not wait for its termination and when it dies do not restart the process. If upon entering a new run state the process is still running

from a previous run state change the program will not be restarted.

boot The entry is to be processed only at *init*'s boot time read of the *inittab* file. *Init* is to start the process, not wait for its termination, and when it dies not restart the process. In order for this instruction to be meaningful, the *rstate* should be the default or it must match *init*'s run state at boot time. This action is useful for an initialization function following a hardware reboot of the system.

bootwait The entry is to be processed only at *init*'s boot time read of the *inittab* file. *Init* is to start the process, wait for its termination and when it dies not restart the process.

powerfail Execute the process associated with this entry only when *init* receives a power fail signal (*SIGPWR* (*signal*(2))).

powerwait Execute the process associated with this entry only when *init* receives a power fail signal (*SIGPWR* (*signal*(2))) and wait until it terminates before continuing any processing of *inittab*.

off If the process associated with this entry is currently running, send the warning signal (*SIGTRM* (*signal*(2))) and wait 20 seconds before forcibly terminating the process via the kill signal (*SIGKIL* (*signal*(2))). If the process is nonexistent ignore the entry.

ondemand This instruction is really a synonym for the *respawn* action. It is functionally identical to *respawn* but is given a different keyword in order to divorce its association with run states. This is used only with the *a*, *b*, or *c* values described in the '*rstate*' field.

initdefault

An entry with this **action** verb is only scanned when *init* initially comes up. *Init* uses this entry, if it exists, to determine which *run state* to enter initially. It does this by taking the highest run level specified in the *rstate* field and using that as its initial state. Two points to note. If the *rstate* field is empty, this is interpreted as '0123456' and so *init* will enter *run state* 6. The second point is that the **initdefault** entry cannot specify that *init* start in the *SINGLE USER* state. Also to be noted is that if *init* doesn't find an **initdefault** entry in */etc/inittab*, then it will request an initial *run state* from the user at reboot time.

process This is a *sh* command to be executed. The entire **process** field is prepended with *exec* and passed to a forked *sh* as **sh -c 'exec command'**. For this reason any legal *sh* syntax can appear in the the **process** field. Comments can be inserted with the

FILES

/etc/inittab

SEE ALSO

getty(1M), *init*(1M), *sh*(1), *who*(1), *exec*(2), *open*(2), *signal*(2)