# *Runlevels, System Shutdown and Reboot*

## Information

These notes were originally written in the year 2000 as part of a set of LPI Exam 101 training materials. The LPI training course at Bromley College was subsequently discontinued and some of the sections of the notes modified and incorporated into our one-day System Administration Courses. The remainder of the notes have now been made publicly available on the linuxtraining.org.uk website.

If you are a beginner please do not be put off of training courses by these notes, as they are rather technical. On the other hand if you are a more experienced Linux user we hope you find the coverage of this topic refreshingly clear.

For full details of our current Linux training please visit the site:
http://ce.bromley.ac.uk/linux

If you have reached this page from a search engine and wish to see the full contents list for the published notes please visit the site:
http://www.linuxtraining.org.uk

We hope you find these notes useful, but please remember that they apply to the 2.2 kernel. I will update them when I have the time.

Clive Gould - 21st December 2004

# *Runlevels, System Shutdown and Reboot*

## Objective 2

***Change runlevels and shutdown or reboot system: Securely change the runlevel of the system, specifically to single user mode, halt (shutdown) or reboot. Make sure to alert users beforehand, and properly terminate processes. Involves using the commands: shutdown, init***

## Runlevels

Runlevels are an important part of system initialisation. They allow control over the services started by the system during the initialisation process. There are variations between one distribution and another in terms of the exact number of runlevels and their purpose.

The numbers associated with the run levels for a particular distribution are listed in the **/etc/inittab** file as illustrated below:

```
 Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not
have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)

id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

# *Runlevels, System Shutdown and Reboot*

Depending on the runlevel chosen the appropriate rc script will be run, which will start any processes required for that particular runlevel. The default runlevel (initdefault) is normally set to the runlevel for either full mutiuser mode, which is the standard command line login, or X11 which is the X-Windows login.

You can change the default login by editing the contents of the initdefault parameter. This can be very useful if you get a distribution which inadvertently boots you into X when you would like a command line login instead. You should never change initdefault to either halt or reboot as the system will then either boot up and immediately shut down again, or reboot over and over again!

The single user mode is different from the other runlevels, and is used for system administration and repair when things go wrong. No other users can log in while the system is in this runlevel, however, you can still run multiple applications. In this mode **/bin/su** is run by default and you automatically become the root user without needing to supply a password.

This can be very useful if you have forgotten the root password as you can type linux single at the LILO command prompt and instantly become root. At the same time this also has grave security implications if unauthorised physical access to a server is available.

## Process Control Initialization -init, telinit

After it has spawned all of the processes specified,  init waits for one of its dependent processes to die, a power fail signal, or until it is signaled by the telinit command (tell init) to change the system's runlevel. When one of the above three conditions occurs, it re-examines the /etc/inittab file. New entries can be added to this file at any time. However, init still waits for one of the above three conditions to occur. To  provide for an instantaneous response, the telinit Q or q command can wake up init to re examine the /etc/inittab file.

The command /sbin/telinit is linked to /sbin/init. The command telinit can be used by the root to change the system from one runlevel to another. This involves init shutting down any processes in the current runlevel which do not belong in the new runlevel, and starting any processes in the new run level which were not present in the previous runlevel.

# *Runlevels, System Shutdown and Reboot*

The command telinit takes a one character argument and signals init to perform the appropriate action. The syntax for the telinit command is shown below:

```
telinit option argument
```

By default, telinit waits for 5 seconds between sending  processes the SIGTERM and SIGKILL. This can be changed using the **`-t seconds`** option to give processes more time to save data.

The table below shows the arguments that serve as directives to telinit:

| *Argument* | *Explanation* |
|---|---|
| `0,1,2,3,4,5 or 6` | Tell init to switch to the specified run level. |
| `Q` or `q` | Tell init to re-examine the /etc/inittab file. |
| `S` or `s` | Tell init to switch to single user mode. |
| `U` or `u` | Tell init to re execute itself (preserving the current state). The init process will not re-read the /etc/inittab file. |

# *Runlevels, System Shutdown and Reboot*

## Bring the System Down - shutdown

You should never shut a running system down just by switching the power off, or reboot a system by pressing the reset button. Instead the shutdown command, which brings the system down in a secure way, should always be used for halting or rebooting the system. All logged in users are notified that the system is going down, and login is blocked.

It is possible to shut the system down immediately, or after a specified delay. All processes are first notified that the system is going down by the signal SIGTERM. This gives programs like vi the time to save the file being edited, mail and news processing programs a chance to exit cleanly.

The shutdown command does its job by signalling the init process, asking it to change the runlevel.

The syntax for shutdown is shown below:

```
shutdown option(s) time warning_message
```

Common shutdown options are:

| *Option* | *Explanation* |
|---|---|
| -c | Cancel an already running shutdown. With this option it is of course not possible to give the time argument, but you can enter a explanatory message on the command line that will be sent to all users. |
| -F | Force fsck on reboot |
| -f | Skip fsck on reboot. |
| -h | Halt after shutdown. |
| -k | Don't really shutdown; only send the warning messages to everybody. |
| -r | Reboot after shutdown |
| -t *SEC* | Tell init to wait *SEC*conds between sending processes the warning and the kill signal, and changing to another runlevel. |

The shutdown command also requires a time argument. Common time arguments are:

| ***Argument*** | ***Explanation*** |
|---|---|
| hh:ss | Shutdown at the time specified |
| +*M* | Shutdown after *M* minutes have elapsed |
| now | Shutdown immediately (an alias for +0) |

The warning message is sent to all logged on users periodically until shutdown takes place. If no message is specified a default message is sent.

Only root can use the shutdown command unless the file **/etc/shutdown.allow** exists, in which case users listed in this file can also shutdown the system.

Assuming the key combination CTRL-ALT-DEL is trapped by an appropriate entry in /etc/inittab then the shutdown command will be called when these keys are pressed. This means that everyone who has physical access to the console keyboard could shut the system down. To prevent this, if shutdown is called from init, it checks to see if the file /etc/shutdown.allow is present. It then compares the login names in that file with the list of people that are logged in (from /var/run/utmp). Only if one of those authorised users or root is logged in, will it proceed. Otherwise it will write the following message to the system console:

```
shutdown: no authorised users logged in
```

An example of using the shutdown command is illustrated below:

```
[root@redhat /root]# shutdown -k +1

Broadcast message from root (0) Wed Aug 23 11:40:49 2000...

The system is going DOWN to maintenance mode in 1 minutes !!


Shutdown cancelled.
```