

Run Levels in Linux

- Last Updated : 16 Apr, 2019

Prerequisite : [Linux Booting](#)

A run level is a state of init and the whole system that defines what system services are operating. Run levels are identified by numbers. Some system administrators use run levels to define which subsystems are working, e.g., whether X is running, whether the network is operational, and so on.

- Whenever a LINUX system boots, firstly the **init** process is started which is actually responsible for running other start scripts which mainly involves initialization of you hardware, bringing up the network, starting the graphical interface.
- Now, the **init** first finds the default **runlevel** of the system so that it could run the start scripts corresponding to the default run level.
- A **runlevel** can simply be thought of as the state your system enters like if a system is in a single-user mode it will have a **runlevel 1** while if the system is in a multi-user mode it will have a **runlevel 5**.
- A **runlevel** in other words can be defined as a **preset single digit integer** for defining the operating state of your LINUX or UNIX-based operating system. Each runlevel designates a different system configuration and allows access to different combination of **processes**.

The important thing to note here is that there are differences in the runlevels according to the operating system. The standard **LINUX kernel** supports these seven different runlevels :

- 0 – System halt *i.e* the system can be safely powered off with no activity.
- 1 – Single user mode.
- 2 – Multiple user mode with no NFS(network file system).
- 3 – Multiple user mode under the command line interface and not under the graphical user interface.
- 4 – User-definable.
- 5 – Multiple user mode under GUI (graphical user interface) and this is the standard runlevel for most of the LINUX based systems.
- 6 – Reboot which is used to restart the system.

By default most of the LINUX based system boots to runlevel 3 or runlevel 5.

In addition to the standard runlevels, users can modify the preset runlevels or even create new ones according to the requirement. Runlevels 2 and 4 are used for user defined runlevels and runlevel 0 and 6 are used for halting and rebooting the system.

Obviously the start scripts for each run level will be different performing different tasks. These start scripts corresponding to each run level can be found in special files present under **rc sub directories**.

At **/etc/rc.d** directory there will be either a set of files named **rc.0, rc.1, rc.2, rc.3, rc.4, rc.5** and **rc.6**, or a set of directories named **rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d** and **rc6.d**.

For example, run level 1 will have its start script either in file **/etc/rc.d/rc.1** or any files in the directory **/etc/rc.d/rc1.d**.

Changing runlevel

init is the program responsible for altering the run level which can be called using **telinit** command.

For example, to change a runlevel from 3 to runlevel 5 which will actually allow the GUI to be started in multi-user mode the **telinit** command can be used as :

*/*using telinit to change*

runlevel from 3 to 5/*

telinit 5

NOTE : The changing of runlevels is a task for the super user and not the normal user that's why it is necessary to be logged in as super user for the successful execution of the above telinit command or you can use **sudo** command as :

```
// using sudo to execute telinit
```

```
sudo telinit 5
```

The default runlevel for a system is specified in **/etc/inittab** file which will have an entry **id : 5 : initdefault** if the default runlevel is set to 5 or will have an entry **id : 3 : initdefault** if the default runlevel is set to 3.

Need for changing the runlevel

- There can be a situation when you may find trouble in **logging in** in case you don't remember the password or because of the corrupted **/etc/passwd** file (have all the user names and passwords), in this case the problem can be solved by booting into a single user mode *i.e* runlevel 1.
- You can easily halt the system by changing the runlevel to 0 by using **telinit 0**.