**RUNLEVEL in LINUX/UNIX system**

**RUNLEVELS: It is nothing but a mode of operation in Linux OS.**

* A runlevel is a software configuration of the system which allows only a selected group of processes to exist.
* The processes spawned by INIT for each of these runlevels are defined in the /etc/inittab file.
* INIT can be in one of eight runlevels: 0-6 and S or s. The runlevel is changed by a privileged user running telinit command, which sends appropriate signals to INIT, telling it which runlevel to change to.



* Runlevels 0, 1, and 6 are reserved.
* Runlevel 0 is used to halt the system, runlevel 6 is used to reboot the system, and runlevel 1 is used to get the system down into single user mode i.e. into a state where administrative tasks can be performed; this is the default if neither the -h or -r flag is given to shutdown.
* Runlevel 5 is full desktop mode with X sessions enabled providing graphical user interface.
* Text based systems are defaulted to boot up with runlevel 3(Multiuser mode with network access).
* Runlevel S is not really meant to be used directly, but more for the scripts that are executed when entering runlevel 1.
* Runlevels 7-9 are also valid, though not really documented. This is because "traditional" Unix variants don't use them.
* Runlevels S and s are in fact the same. Internally they are aliases for the same runlevel.

**Runlevels and their meaning:**

| **Runlevels** | **System Configuration** |
| --- | --- |
| 0 | System halt (Do not use this for initdefault!) |
| 1 | Single user mode |
| 2 | Local multiuser without remote network (e.g. NFS) |
| 3 | Full multiuser with network |
| 4 | Not used |
| 5 | Full multiuser with network and xdm(Graphical mode with X session enabled.) |
| 6 | System reboot (Do not use this for initdefault!) |

* SHUDOWN does its job by signaling the INIT process, asking it to change the runlevel.
* To see which actions are taken on halt or reboot see the appropriate entries for these runlevels in the file /etc/inittab.

**#The default runlevel of a system in which it will boot up is defined here in /etc/inittab file:**

20 id:3:initdefault:

**Changing Runelevels**

To change the runlevel temporarily we can use init command. For eg. to change to runlevel 2 from 3, use:

|  |
| --- |
| localhost:>init 2 |

But again after system reboot the system will go the default runlevel i.e 3. To change runlevel permanently, edit the **/etc/initab** file and change the runlevel value in the above line i.e. 20 id:3:initdefault.

* **To Check the current runlevel of the system**

runlevel -- find the current and previous system runlevel.

localhost:>runlevel
N 3

Here, we get two outputs. First one represents previous runlevel and second one is the current runlevel. If there is no previous runlevel, it will be displayed as ‘N’ as shown in above output.

**Also you can use who -r to check runlevel of the system.**

localhost:/sbin:>who -r
run-level 3 2014-10-27 18:40 last=S

* **To find what services and processes are running in a particular runlevel, we can use chkconfig command.**

localhost:/etc:>chkconfig|more

SuSEfirewall2\_init off
SuSEfirewall2\_setup off
aaeventd off
acpid on
apache2 on
arpd off

* **To get previous runlevel of the system**

PREVLEVEL
To get previous runlevel (It is useful after a runlevel switch is done).

SYNOPSIS
runlevel [utmp]

* Runlevel reads the system utmp file (typically /var/run/utmp) to locate the runlevel record, and then prints the previous and current system runlevel on its standard output, separated by a single space.
* If no utmp file exists, or if no runlevel record can be found, runlevel prints the word unknown and exits with an error.
* Runlevel can be used in rc scripts as a substitute for the System-V who -r command. However, in newer versions of init(8) this information is also available in the environment variables RUNLEVEL and PREVLEVEL.
* INIT process is the last step in booting and it has the process id as “1”. It is responsible for starting all system processes as defined in /etc/inittab file.
* INIT process first checks the runlvel which is defined in /etc/inittab file by default and then starts the system in that runlevel which means that all the processes/services which are defined fro that runlevel get executed.
* Most desktop computers boot in runlevel 5 by default which a full graphical user interface. In this mode the users get X session feature enabled.
* Most text based servers boot in runlevel 3 which provide text based login prompt. Usually it is advisable not to use graphical interface in a server as it takes a lot space and resource to run.
* Each runlevel has its own directory structure. These directories are present inside /etc/rc.d/ as rc0.d, rc1.d... rc6.d, rcS.d. Here each number represent the runlevel.

**Directory structure for RUNLEVEL 0:**

server:/etc/rc.d/rc0.d:>ll

total 8

drwxr-xr-x 2 root root 4096 2010-06-17 09:49 .

drwxr-xr-x 11 root root 4096 2015-01-28 19:54 ..

lrwxrwxrwx 1 root root 7 2010-10-30 19:59 S01halt -> ../halt

**Directory structure for RUNLEVEL 1:**

server:/etc/rc.d/rc1.d:>ll

total 8

drwxr-xr-x 2 root root 4096 2014-02-04 09:31 .

drwxr-xr-x 11 root root 4096 2015-01-28 19:54 ..

lrwxrwxrwx 1 root root 6 2010-10-30 19:59 S06kbd -> ../kbd

lrwxrwxrwx 1 root root 15 2010-10-30 19:59 S09irq\_balancer -> ../irq\_balancer

lrwxrwxrwx 1 root root 9 2010-10-30 19:59 S09splash -> ../splash

lrwxrwxrwx 1 root root 9 2010-10-30 19:59 S10single -> ../single

**Directory structure for RUNLEVEL S/s:**

server:/etc/rc.d/rcS.d:>ll

total 8

drwxr-xr-x 2 root root 4096 2014-02-04 09:31 .

drwxr-xr-x 11 root root 4096 2015-01-28 19:54 ..

lrwxrwxrwx 1 root root 13 2010-10-30 19:59 S03boot.clock -> ../boot.clock

lrwxrwxrwx 1 root root 6 2010-10-30 19:59 S06kbd -> ../kbd

lrwxrwxrwx 1 root root 9 2010-10-30 19:59 S09splash -> ../splash

lrwxrwxrwx 1 root root 9 2010-10-30 19:59 S10single -> ../single

Runlevel S or s bring the system to single user mode and do not require a /etc/inittab file.

In single user mode, /sbin/sulogin is invoked on /dev/console. While entering single user mode, INIT initializes the consoles stty settings to sane values.

Clocal mode is set. Hardware speed and handshaking are not changed.

**More on runlevel: Booting**

After init is invoked as the last step of the kernel boot sequence, it looks for the file /etc/inittab to see if there is an entry of the type **initdefault** (see inittab(5)).

 **The initdefault entry determines the initial runlevel of the system. Please note that if there is no such entry (or no /etc/inittab at all), a runlevel must be entered at the system console.**

When entering a multi-user mode for the first time, init performs the boot and bootwait entries to allow file systems to be mounted before users can log in. Then all entries matching the runlevel are processed.

**ENVIRONMENT**

 Init sets the following environment variables for all its children:

 **PATH** /bin:/usr/bin:/sbin:/usr/sbin

 INIT\_VERSION

It is useful to determine if a script runs directly from INIT.

 **RUNLEVEL**

 The current system runlevel

 **PREVLEVEL**

 The previous runlevel (useful after a runlevel switch)

 **CONSOLE**

 The system console; this is taken from the kernel setting; however if it is not set INIT will set it to /dev/console by default.

**BOOTFLAGS**

 It is possible to pass a number of flags to init from the boot monitor (eg. LILO). INIT accepts the following flags:

 **-s, S, single**

 Single user mode boot. In this mode /etc/inittab is examined and the bootup rc scripts are usually run before the single user mode shell is started.

 **1-5:** Runlevel to boot into.

 **-b, emergency**

 Boot directly into a single user shell without running any other startup scripts.

 **-a, auto**

 The LILO boot loader adds the word "auto" to the command line if it booted the kernel with the default command line (without user intervention).

 If this is found INIT sets the "AUTOBOOT" environment variable to "yes". Note that you cannot use this for any security measures - of course the user could specify "auto" or -a on the command line manually.

 **-z xxx**

 The argument to -z is ignored. You can use this to expand the command line a bit, so that it takes some more space on the stack. INIIT can then manipulate the command line so that ps shows the current runlevel.

**INTERFACE**

 Init listens on a fifo in /dev, /dev/initctl, for messages. Telinit uses this to communicate with init. The interface is not very well documented or finished. Those interested should study the initreq.h file in the src/ subdirectory of the init source code tar archive.

**SIGNALS**

 Init reacts to several signals:

 SIGHUP

 Has the same effect as telinit q.

 SIGUSR1

* On receipt of this signals, init closes and re-opens its control fifo.
* /dev/initctl. Useful for bootscripts when /dev is remounted.

 SIGINT

 Normally the kernel sends this signal to init when CTRL-ALT-DEL is pressed. It activates the ctrlaltdel action.

 SIGWINCH

 The kernel sends this signal when the KeyboardSignal key is hit. It activates the kbrequest action.

CONFORMING TO

 INIT is compatible with the System V init. It works closely together with the scripts in the directories /etc/init.d and /etc/init.d/rc{runlevel}.d.

As per system’s convention, there is a **README** file in the directory /etc/init.d explaining how these scripts work.

**README File explaining all about runlevel and INIT in detail**

The scripts for controlling the system are placed in /etc/init.d/ (they have been moved according to the Linux Standard Base (LSB) specification). These scripts are executed directly or indirectly by /sbin/init, the father of all processes. The configuration of /sbin/init is given by the file /etc/inittab (see inittab(5)).

 At boot time, the boot level master script /etc/init.d/boot is called to initialize the system (e.g. file system check, ...). It also executes some hardware init scripts linked into /etc/init.d/boot.d/. Then it calls /etc/init.d/boot.local, which executes the local commands.

 After system startup, /sbin/init will normally switch on the default run level given in /etc/inittab. It calls the run level master script /etc/init.d/rc to start or stop services provided by the other scripts under /etc/init.d/.

 Both scripts, then boot level master script /etc/init.d/boot and the the run level master script /etc/init.d/rc starts all other boot or runlevel scripts either sequential or partial parallel within their dependencies order.

 To control the services of a run level, the corresponding scripts are linked into run level directories /etc/init.d/rc<X>.d/, where <X>=0,1,2,3,4,5,6,S is the run level number.

 **There are two kinds of symbolic link:**

**1. Start links**: Called when entering a runlevel.

**2. Stop links**: Called when leaving a runlevel.

Note that each service in the run levels 2, 3, 4, and 5 consists of a start and a stop link. Within SuSE boot concept a differential link scheme is used to be able to change a runlevel in comparison with the former level.

 If parallel executing of the boot scripts is enabled (see /etc/sysconfig/boot variable RUN\_PARALLEL) then both master scripts uses the program startpar(8) which starts or stops multiple services in parallel.

 Startpar(8) will look for the files /etc/init.d/.depend.boot, /etc/init.d/.depend.start, and /etc/init.d/.depend.stop to get the dependencies for each service. The files will be written, beside the symbolic links in the boot and runlevel directories, by the program insserv(8).

 To avoid redundant starts when changing run levels, only those services are started which have no start link in the previous run level. And to avoid redundant stops when changing run levels, only those services are stopped which have no start link in the current level. To control this behavior, the names of the scripts are added on the names of the start and stop links.

 To control the order of service starts and stops, the start and stop links include a number in their link name.

 The system configuration files in /etc/sysconfig contain most of the variables used to configure the installed services. These variables can easily be changed by YaST or by using an editor. After using an editor, the script /sbin/SuSEconfig must be called to distribute the settings into the system.

**For example:**

 The script /etc/init.d/lpd starts or stops the line printer daemon for the printing service, according to the flag used:

 /etc/init.d/lpd start

 And

 /etc/init.d/lpd stop

 To do this automatically in run level 3, this script is linked into /etc/init.d/rc3.d/ with these two symbolic links

 /etc/init.d/rc3.d/S20lpd -> ../lpd

 and

 /etc/init.d/rc3.d/K20lpd -> ../lpd

 The corresponding link with the letter S is used to start a service. For the printing service the number between the letter S and the name should be greater than the number of the start link of the network service.

The corresponding link with the letter K is used to stop a service. The number of the stop link for the printing service should be less than that of the stop link for the network service so that the printer daemon is stopped before shutting down the network service.

**Run levels and their services**

**0** This level is used for halting the system. The only valid service for this level is the script halt, which is linked into /etc/init.d/rc0.d/. The script halt executes /etc/init.d/halt.local. Special system issues for halt or reboot should be added there.

**6** This level is used for rebooting the system. The only valid service for this level is the script reboot, which is linked into /etc/init.d/rc6.d/. The script reboot executes /etc/init.d/halt.local. Special system issues for halt or reboot should be added there.

 **S** This mode is used to switch from boot phase into single user mode. The last valid service for this mode is the script single, which is linked into /etc/init.d/rcS.d/. In this mode you have only one console.

**1** According to the Linux Standard Base (LSB) specification this runlevel is used to switch from normal runlevel into single user mode. This is different from former SuSE Linux versions!

**2** The runlevel 2 is without remote networking. Note that on some other systems this is identical with the single user mode. This run level can have more than one virtual console.

**3** The runlevel 3 is with network. This run level is for server stations not automatically running X sessions.

**5** The level 5 is with network and xdm(1). You should have a configured and perfectly running X Window System for this work station run level.

**4** The run level 4 is not (yet) used.

**/etc/init.d/skeleton**

 This script is a model for writing your own. You can use insserv(8) to include your own script into a run level.

**Related Files to system runlevel:**

 /etc/init.d/\*

 /etc/init.d/boot

 /etc/init.d/boot.local

 /etc/init.d/halt

 /etc/init.d/halt.local

 /etc/init.d/rc

 /etc/init.d/reboot

 /etc/init.d/skeleton

 /etc/init.d/single

 /etc/init.d/boot.d/S[0-9][0-9]\*

 /etc/init.d/rc0.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc1.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc2.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc3.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc4.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc5.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rc6.d/{K,S}[0-9][0-9]\*

 /etc/init.d/rcS.d/{K,S}[0-9][0-9]\*

 /etc/init.d/.depend.boot

 /etc/init.d/.depend.start

 /etc/init.d/.depend.stop

 /etc/inittab

 /etc/sysconfig/boot

 /etc/sysconfig

**Warnings**

 INIT assumes that processes and descendants of processes remain in the same process group which was originally created for them.

If the processes change their group, INIT can't kill them and you may end up with two processes reading from one terminal line.

Authors:

Shankar Kumar Bhagat

[www.shankysportal.com](http://www.shankysportal.com)

Copyright reserved.