**How to check and change a default runlevel on Ubuntu Linux**

admin

Ubuntu

24 August 2018

To check your current and previous runlevel use runlevel command:

# runlevel

N 2

From the above command output we can see that there was no previous runlevel detected and current run level is 2. Runlevel list defined by the Ubuntu Linux:

* 1 - Single
* 2,3,4,5 - User defined runlevels. Default 2
* 6 - Reboot

To temporarily change the runlevel on your Ubuntu Linux system use telinit or directly invoke init command. The following [linux command](https://linuxconfig.org/linux-commands) will change runlevel to 1:

# init 1

OR

# telinit 1

***SUBSCRIBE TO NEWSLETTER*** *Subscribe to Linux Career*[*NEWSLETTER*](https://bit.ly/2X5D30q)*and receive latest Linux news, jobs, career advice and tutorials.*

To permanently change your runlevel on Ubuntu Linux edit the following line within /etc/init/rc-sysinit.conf:

env DEFAULT\_RUNLEVEL=2

Another alternative on how to change a default runlevel on Ubuntu Linux is to edit Grub's default startup sequence. Locate /etc/default/grub file and edit line:

GRUB\_CMDLINE\_LINUX=""

To include your desired runlevel. For example to change default runlevel to 5 edit the above line to desired runlevel. For example to change to runlevel 5 insert:

GRUB\_CMDLINE\_LINUX="5"

and run Grub update command with administrative privileges:

# update-grub

After reboot the system will now boot to runlevel 5.

On Linux systems, run levels are operational levels that describe the state of the system with respect to what services are available.

One run level is restrictive and used only for maintenance; network connections will not be operational, but admins can log in through a console connection.

Others allow anyone to log in and work, but maybe with some differences in the available services. This post examines how run levels are configured and how you can change the run level interactively or modify what services are available.

[[Get regularly scheduled insights by signing up for Network World newsletters.]](https://www.networkworld.com/newsletters/signup.html)

The default run state on Linux systems — the one that will be used when the system starts up (unless instructed otherwise) — is usually configured in the **/etc/inittab** file, which generally looks something like this:

id:3:initdefault:

Some, including Debian systems, default to run state 2, rather than 3, and don’t generally have an /etc/inittab file at all.

How run levels are set up by default and how they are configured depends in part on the particular distribution you are running. On some systems, for example, run level 2 is multi-user, and run level 3 is multi-user with NFS (file system sharing) support. On others, run levels 2-5 are basically identical. Run level 1 is single-user mode. Run levels on Debian systems, for example, will default to this kind of setup:

Run levels on Debian systems default to this kind of setup:

0 = halted

1 = single user (maintenance mode)

2 = multi-user mode

3-5 = same as 2

6 = reboot

On Linux systems that use run level 3 to share file systems with other systems, it easy to start or stop file system sharing without changing anything about the system but the run level. Changing run level from 2 to 3 would allow the file systems to be shared. Changing the run level from 3 to 2 would disable sharing.

What processes are run in any run level depends on the contents of the /etc/rc?.d directory where ? might be 2, 3, 4, or 5 (corresponding to the run level).

On the system used in the example below (an Ubuntu system), we can see that the content of the directories for these four run states are all set up the same — because the configuration of the directories are the same.

/etc/rc2.d$ ls

README S20smartmontools S50saned S99grub-common

S20kerneloops S20speech-dispatcher S70dns-clean S99ondemand

S20rsync S20sysstat S70pppd-dns S99rc.local

/etc/rc2.d$ cd ../rc3.d

/etc/rc3.d$ ls

README S20smartmontools S50saned S99grub-common

S20kerneloops S20speech-dispatcher S70dns-clean S99ondemand

S20rsync S20sysstat S70pppd-dns S99rc.local

/etc/rc3.d$ cd ../rc4.d

/etc/rc4.d$ ls

README S20smartmontools S50saned S99grub-common

S20kerneloops S20speech-dispatcher S70dns-clean S99ondemand

S20rsync S20sysstat S70pppd-dns S99rc.local

/etc/rc4.d$ cd ../rc5.d

/etc/rc5.d$ ls

README S20smartmontools S50saned S99grub-common

S20kerneloops S20speech-dispatcher S70dns-clean S99ondemand

S20rsync S20sysstat S70pppd-dns S99rc.local

And what are these files? They’re all symbolic links that point to scripts in the /etc/init.d directory that start services. And the names of the files are important because they determine the order in which the scripts are run. For example, S20 scripts are run before S50 scripts.

$ ls -l

total 4

-rw-r--r-- 1 root root 677 Feb 16 2016 README

lrwxrwxrwx 1 root root 20 Aug 30 14:40 S20kerneloops -> ../init.d/kerneloops

lrwxrwxrwx 1 root root 15 Aug 30 14:40 S20rsync -> ../init.d/rsync

lrwxrwxrwx 1 root root 23 Aug 30 16:10 S20smartmontools -> ../init.d/smartmontools

lrwxrwxrwx 1 root root 27 Aug 30 14:40 S20speech-dispatcher -> ../init.d/speech-dispatcher

lrwxrwxrwx 1 root root 17 Aug 31 14:12 S20sysstat -> ../init.d/sysstat

lrwxrwxrwx 1 root root 15 Aug 30 14:40 S50saned -> ../init.d/saned

lrwxrwxrwx 1 root root 19 Aug 30 14:40 S70dns-clean -> ../init.d/dns-clean

lrwxrwxrwx 1 root root 18 Aug 30 14:40 S70pppd-dns -> ../init.d/pppd-dns

lrwxrwxrwx 1 root root 21 Aug 30 14:40 S99grub-common -> ../init.d/grub-common

lrwxrwxrwx 1 root root 18 Aug 30 14:40 S99ondemand -> ../init.d/ondemand

lrwxrwxrwx 1 root root 18 Aug 30 14:40 S99rc.local -> ../init.d/rc.local

The /etc/rc1.d directory, as you’d probably suspect, is different because run level 1 is so different. It contains symbolic links that point to a very different set of scripts. Notice, too, that some of the symbolic links start with the letter K, while others start with the more normal S. This is because some services need to ***stop*** when a system enters single user mode. While some of these links point to the same scripts that are used in other run levels, the K (kill) indicates that these scripts will be run with an argument that instructs the services to stop rather than one that instructs them to start.

/etc/rc1.d$ ls -l

total 4

lrwxrwxrwx 1 root root 20 Aug 30 14:40 K20kerneloops -> ../init.d/kerneloops

lrwxrwxrwx 1 root root 15 Aug 30 14:40 K20rsync -> ../init.d/rsync

lrwxrwxrwx 1 root root 15 Aug 30 14:40 K20saned -> ../init.d/saned

lrwxrwxrwx 1 root root 23 Aug 30 16:10 K20smartmontools -> ../init.d/smartmontools

lrwxrwxrwx 1 root root 27 Aug 30 14:40 K20speech-dispatcher -> ../init.d/speech-dispatcher

-rw-r--r-- 1 root root 369 Mar 12 2014 README

lrwxrwxrwx 1 root root 19 Aug 30 14:40 S30killprocs -> ../init.d/killprocs

lrwxrwxrwx 1 root root 19 Aug 30 14:40 S70dns-clean -> ../init.d/dns-clean

lrwxrwxrwx 1 root root 18 Aug 30 14:40 S70pppd-dns -> ../init.d/pppd-dns

lrwxrwxrwx 1 root root 16 Aug 30 14:40 S90single -> ../init.d/single

You can change the default run level on a system, though there is rarely a need to do so. For example, you could configure a Debian system to default to a run level of 3 (rather than 2) by setting up an /etc/inittab file that looks, for example, like this one:

id:3:initdefault:

Once you make the change and reboot, the runlevel command would show you this:

$ runlevel

N 3

Alternately, if you used the **init 3** command, you would also change run levels (rebooting is not required to change run states) and your runlevel output would look like this:

$ runlevel

2 3

Of course, there’s little reason to change your default state by creating or modifying **/etc/inittab** unless you modify the symbolic links in the corresponding /etc/rc?.d directory to differentiate what will be running in the modified run state.

**How to use run levels on Linux**

To recap, here's a quick Q&A on run levels:

**How do you tell what run level you are in?**

Use the **runlevel** command.

**How do you see what processes are associated with a particular run level?**

Look at the associated run level start directory (e.g., /etc/rc2.d for run level 2).

**How do you know what the default run level is?**

Check **/etc/inittab** if it exists. If not, just ask runlevel. You’re likely already in that run level.

**How do you change run levels?**

Use the **init** command (e.g., init 3) to change it temporarily. Modify or set up /etc/inittab to make a permanent change.

**Can you change what services run in some particular run level?**

Of course — by modifying the symbolic links in the associated /etc/rc?.d directory.

**What else should you consider?**

You should always exercise some caution when changing run levels on a Linux server to ensure that you’re not going to be affecting services that are currently in use or users who are logged in.