



Tutorial Link <https://course.testpad.chitkara.edu.in/tutorials/Linux - Command Line Operations/60cc7afacaf5f6628f2d3f3>

## TUTORIAL

# Linux - Command Line Operations

## Topics

### 1.1 Command line operations

## Command line operations

### Command line operations

Linux system administrators spend a significant amount of their time at a command line prompt. They often automate and troubleshoot tasks in this text environment. There is a saying, "graphical user interfaces make easy tasks easier, while command line interfaces make difficult tasks possible". Linux relies heavily on the abundance of command line tools. The command line interface provides the following advantages:

- No GUI overhead is incurred.
- Virtually any and every task can be accomplished while sitting at the command line.
- You can implement scripts for often-used (or easy-to-forget) tasks and series of procedures.
- You can sign into remote machines anywhere on the Internet.
- You can initiate graphical applications directly from the command line instead of hunting through menus.
- While graphical tools may vary among Linux distributions, the command line interface does not.

A terminal emulator program emulates (simulates) a standalone terminal within a window on the desktop. By this, we mean it behaves essentially as if you were logging into the machine at a pure text terminal with no running graphical interface. Most terminal emulator programs support multiple terminal sessions by opening

additional tabs or windows. By default, on GNOME desktop environments, the `gnome-terminal` application is used to emulate a text-mode terminal in a window.

To open a terminal on any system using a recent GNOME desktop, including CentOS 7/Red Hat 7/OpenSUSE, click on Applications > System Tools > Terminal.

On any but some of the most recent GNOME-based distributions, you can always open a terminal by right-clicking anywhere on the desktop background and selecting Open in Terminal. If this does not work, try installing the `nautilus-open-terminal` package if your Linux distribution offers it. You can also hit "Alt + F2" and type in either `gnome-terminal` or `konsole`, whichever is appropriate or press "Ctrl + Alt + T".

There are some basic command line utilities that are used constantly, and it would be impossible to proceed further without using some of them in simple form before we discuss them in more detail. A short list has to include:

- `cat`: used to type out a file (or combine files)
- `head`: used to show the first few lines of a file
- `tail`: used to show the last few lines of a file
- `man`: used to view documentation.

Most input lines entered at the shell prompt have three basic elements:

- Command
- Options
- Arguments.

The command is the name of the program you are executing. It may be followed by one or more options (or switches) that modify what the command may do. Options usually start with one or two dashes, for example, `-p` or `--print`, in order to differentiate them from arguments, which represent what the command operates on. However, plenty of commands have no options, no arguments, or neither. In addition, other elements (such as setting environment

variables) can also appear on the command line when launching a task.

## Virtual Terminals

Virtual Terminals (VT) are console sessions that use the entire display and keyboard outside of a graphical environment. Such terminals are considered "virtual" because, although there can be multiple active terminals, only one terminal remains visible at a time. A VT is not quite the same as a command line terminal window; you can have many of those visible at once on a graphical desktop. One virtual terminal (usually number one or seven) is reserved for the graphical environment, and text logins are enabled on the unused VTs. Ubuntu uses VT 7, but CentOS/RHEL and openSUSE use VT 1 for the graphical display.

An example of a situation where using VTs is helpful is when you run into problems with the graphical desktop. In this situation, you can switch to one of the text VTs and troubleshoot. To switch between VTs, press CTRL-ALT-function key for the VT. For example, press CTRL-ALT-F6 for VT 6. Actually, you only have to press the ALT-F6 key combination if you are in a VT and want to switch to another VT.

## Turning Off the Graphical Desktop

Linux distributions can start and stop the graphical desktop in various ways. The exact method differs from distribution and among distribution versions. For the newer systemd-based distributions, the display manager is run as a service, you can stop the GUI desktop with the systemctl utility and most distributions will also work with the telinit command, as in:

```
$ sudo systemctl stop gdm (or sudo telinit 3)
```

and restart it (after logging into the console) with:

```
$ sudo systemctl start gdm (or sudo telinit 5)
```

On Ubuntu versions before 18.04 LTS, substitute lightdm for gdm.

## Rebooting and Shutting Down

The preferred method to shut down or reboot the system is to use the shutdown command. This sends a warning message, and then prevents further users from logging in. The init process will then control shutting down or rebooting the system. It is important to always shut down properly; failure to do so can result in damage to the system and/or loss of data.

The halt and poweroff commands issue shutdown -h to halt the system; reboot issues shutdown -r and causes the machine to reboot instead of just shutting down. Both rebooting and shutting down from the command line requires superuser (root) access. When administering a multiuser system, you have the option of notifying all users prior to shutdown, as in:

```
$ sudo shutdown -h 10:00 "Shutting down for scheduled maintenance."
```

## Locating Applications

Depending on the specifics of your particular distribution's policy, programs and software packages can be installed in various directories. In general, executable programs and scripts should live in the /bin, /usr/bin, /sbin, /usr/sbin directories, or somewhere under /opt. They can also appear in /usr/local/bin and /usr/local/sbin, or in a directory in a user's account space, such as /home/student/bin. One way to locate programs is to employ the which utility. For example, to find out exactly where the diff program resides on the filesystem:

```
$ which diff
```

```
/usr/bin/diff
```

If which does not find the program, whereis is a good alternative because it looks for packages in a broader range of system directories:

```
$ whereis diff
```

```
diff: /usr/bin/diff /usr/share/man/man1/diff.1.gz  
/usr/share/man/man1p/diff.1p.gz
```

as well as locating source and man files packaged with the program.

## Modifying the Command Line Prompt

The PS1 variable is the character string that is displayed as the prompt on the command line. Most distributions set PS1 to a known default value, which is suitable in most cases. However, users may want custom information to show on the command line. For example, some system administrators require the user and the host system name to show up on the command line as in:

```
student@codequotient $
```

This could prove useful if you are working in multiple roles and want to be always reminded of who you are and what machine you are on. The prompt above could be implemented by setting the PS1 variable to: `\u@\h \$.`

For example:

```
$ echo $PS1  
\$  
$ PS1="\u@\h \$ "  
student@codequotient $ echo $PS1  
\u@\h \$  
student@codequotient $
```

By convention, most systems are set up so that the root user has a pound sign (#) as their prompt.

## **Package Management Systems on Linux**

The core parts of a Linux distribution and most of its add-on software are installed via the Package Management System. Each package contains the files and other instructions needed to make one software component work well and cooperate with the other components that comprise the entire system. Packages can depend on each other. For example, a package for a web-based application written in PHP can depend on the PHP package.

There are two broad families of package managers: those based on Debian and those which use RPM as their low-level package manager. The two systems are incompatible, but broadly speaking, provide the same features and satisfy the same needs. There are some other systems used by more specialized Linux distributions.

## **Package Managers: Two Levels**

Both package management systems operate on two distinct levels: a low-level tool (such as dpkg or rpm) takes care of the details of unpacking individual packages, running scripts, getting the software installed correctly, while a high-level tool (such as apt-get, yum, dnf or zypper) works with groups of packages, downloads packages from the vendor, and figures out dependencies.

Most of the time users need to work only with the high-level tool, which will take care of calling the low-level tool as needed. Dependency resolution is a particularly important feature of the high-level tool, as it handles the details of finding and installing each dependency for you. Be careful, however, as installing a single package could result in many dozens or even hundreds of dependent packages being installed.

## Different Package Management Systems

The Advanced Packaging Tool (apt) is the underlying package management system that manages software on Debian-based systems. While it forms the backend for graphical package managers, such as the Ubuntu Software Center and synaptic, its native user interface is at the command line, with programs that include apt-get and apt-cache. For example,

```
$ apt-get install firefox
```

Yellowdog Updater Modified (yum) is an open source command-line package-management utility for the RPM-compatible Linux systems that belongs to the Red Hat/Fedora family. yum has both command line and graphical user interfaces. Recent Fedora versions have replaced yum with a new utility called dnf, which has less historical baggage, has nice new capabilities and is mostly backwards-compatible with yum for day-to-day commands.

zypper is the package management system for the SUSE/openSUSE family and is also based on RPM. zypper also allows you to manage repositories from the command line. zypper is fairly straightforward to use and resembles yum quite closely.



# CodeQuotient

Tutorial by codequotient.com | All rights

reserved, CodeQuotient 2025