

An A-Z index of the **Bash** command line for Linux.

`adduser` Add a user to the system
`addgroup` Add a group to the system
`alias` Create an alias •
`apropos` Search Help manual pages (man -k)
`apt-get` Search for and install software packages
`aptitude` Search for and install software packages
`aspell` Spell Checker
`awk` Find and Replace text, database sort/val

`basename` Strip directory and suffix from filename
`bash` GNU Bourne-Again SHell
`bc` Arbitrary precision calculator language
`bg` Send to background
`break` Exit from a loop •
`builtin` Run a shell builtin
`gzip?` Compress or decompress named file(s)

`cal` Display a calendar
`case` Conditionally perform a command
`cat` Display the contents of a file
`cd` Change Directory
`cfdisk` Partition table manipulator for Linux
`chgrp` Change group ownership
`chmod` Change access permissions
`chown` Change file owner and group
`chroot` Run a command with a different root directory
`chkconfig` System services (runlevel)
`cksum` Print CRC checksum and byte counts
`clear` Clear terminal screen
`cmp` Compare two files
`comm` Compare two sorted files line by line
`command` Run a command - ignoring shell functions •
`continue` Resume the next iteration of a loop •
`cp` Copy one or more files to another location
`cron` Daemon to execute scheduled commands
`crontab` Schedule a command to run at a later time
`csplit` Split a file into context-determined pieces
`cut` Divide a file into several parts

Linux Commands

*This material is based on work supported by the
National Science Foundation under Grant No. 0802551*



Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author (s) and do not necessarily reflect the views of the National Science Foundation

Lesson Overview

Linux can be used with a graphical user interface (GUI) that uses menu, buttons, icons, and labels to represent tasks, actions, or choices. Selecting a GUI element performs the selected action. The GUI is a very powerful tool that makes computing easier—especially for new users. However, a GUI interface requires many steps and mouse clicks to perform tasks and not all GUI interfaces are intuitive.

A faster method for interacting with a Linux system is the use of a command line interface in which a user instructs the computer to perform a variety of tasks by simply typing commands on a command line or in a terminal window.

In this lesson, you will be using the command line interface to execute a series of commands. You will explore useful Linux commands, scripts, and programming concepts that every budding administrator must know. Learning Linux commands is important because many complex task require the command line, or are easier to perform in that environment. Linux systems require command line use for some tasks, and all good Linux administrators must be comfortable using it.



Student Expectations

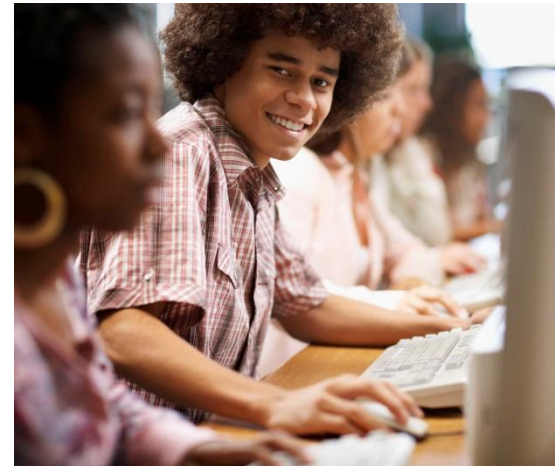
You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives.

Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.



Objective

Given a Linux command line interface, a student will be able to execute the shell command and use the appropriate syntax to perform basic command line shell functions as per industry standards.



Lesson Outline

During this lesson, you will explore:

- Programming terms
- Bash shell
- Command line tasks
- Syntax & Characters
- Text Editors

Select **PLAY** below to view an introductory video.



View Video
Videolesson2LessonOutline(C2L2S1
3).mp4

Choosing a Text Editor

Text editors are simple computer programs that allow you to create or edit text files. Text editors are often used for programming purposes or to write and edit computer code. Text editors differ from word processors in several ways, but one prominent difference is that text editors process only the actual characters in a file while word processors add special formatting and other hidden code.

To edit scripts and computer code, you need a text editor. Linux offers more than 15 different editors, so you are free to use the one that offers the most value to you. Some of the more prominent editors include:

- VI
- VIM
- EMACS
- GEDIT
- CREAM
- JEDIT
- NANO

- SCITE
- LEAFPAD
- GEANY
- BLUEFISH
- XEMACS
- KWRITE
- SCRIBES
- LYX

Select **PLAY** below to view videos on text editors.



View Video
VideoLesson2TextEditors(C2L2S14V1).
mp4

Text editor overview



View Video
VideoLesson2TextEditors(C2L2S14V2).
mp4

Description of text editors

Selecting the Default Editor

To set the default editor on your Ubuntu system, use the **update-alternatives** command. Open a terminal window and type the following command:

```
sudo update-alternatives - -config editor
```

Enter your sudo password to continue . . .

The resulting screen will provide you with a list of choices for your editor. You may keep the default editor—indicated by an asterisk (*) or you may select your preferred editor from the list of options that generally include the following:

Selection	Path	Priority	Status
*0	/bin/nano	40	Auto
1	/bin/ed	-100	Manual
2	/bin/nano	40	Manual
3	/usr/bin/vim.tiny	10	Manual

```
gregory@LincsVB: ~  
File Edit View Search Terminal Help  
To run a command as administrator (user "root"), use "sudo <  
See "man sudo_root" for details.  
gregory@LincsVB:~$ sudo update-alternatives --config editor  
[sudo] password for gregory:  
There are 3 choices for the alternative editor (providing /u  
-----  
Selection    Path          Priority  Status  
-----  
*0           /bin/nano     40       auto mode  
1           /bin/ed       -100     manual mode  
2           /bin/nano     40       manual mode  
3           /usr/bin/vim.tiny 10       manual mode  
Press enter to keep the current choice[*], or type selection
```

Note:
You can test your editor choice by typing **crontab -e** at the command line.

Recommended Editors

Of the many available text editors, four are recommended:

VI is a display oriented, interactive text editor which allows a user to create, modify, and store files on the computer via a terminal.

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the Vi editor distributed with most UNIX systems. Vim is short for 'Vi Improved.'

GNU **Emacs** is an extensible, customizable text editor—and more. It has several features including content-sensitive editing modes and complete built-in documentation.

Cream is a set of scripts and add-on tools that can be added to Vim. Cream does not change the visual look of Vim, but Cream expands the functionality of Vim and makes it easier to use.

Required Reading: Vi

- [Calling VI](#)
- [Vi and Vim Editors](#)
- [Evil text editor?](#)
- [The Vi Editor](#)
- [Using Vi](#)

Required Reading: Vim

- [Vim home](#)
- [Vim Installation](#)
- [Vim color editor](#)

Required Reading: Emacs

- [Emacs home](#)
- [Emacs popularity](#)
- [Emacs guide](#)
- [Emacs commands](#)
- [More Emacs commands](#)

Required Reading: Cream

- [Cream for Vim](#)
- [Adding Cream to Vim](#)
- [Cream for Vim](#)
- [Cream](#)

Command Line Commands

After selecting your editor, it is time to explore commands.



Select **PLAY** below to view two videos on command line tasks:

View Video
VideoLesson2UsefulCommands(C2L2
S17V1).mp4

Useful commands



View Video
VideoLesson2NetworkCommands(C2
L2S17V2).mp4

Network commands

Frequently Used Commands

Once you access a terminal window in Linux, there are hundreds of commands at your disposal. The list below introduces a few of these commands:

- **ls** - list contents of a directory
- **sudo** - "super-user do" (grants administrator rights)
- **cd** - change directory
- **aptitude** - APT package management system (update, install, remove, search)
- **clear** - clear screen
- **chmod** - change file access permissions
- **chown** - change file and group ownership
- **cp** - copy
- **mv** - move
- **rm** - remove
- **cat** - concatenate files (dump to screen)
- **nano** - basic text editor
- **vi** - advanced text editor
- **fdisk** - partition table manipulator
- **df** - disk free (remaining / used disk space)
- **users** - users currently logged in
- **useradd** - add a user
- **usermod** - modify existing user
- **uname** - show system data (try `uname -a`)
- **mount** - mount a file system, cd or removable drive
- **umount** - un-mount a file system, cd or removable drive
- **top** - show current running processes
- **touch** - create new, empty, file
- **reboot** - reboot your system
- **shutdown** - shutdown your system
- **passwd** - change user password
- **ping** - ping a network device or location (ping google.com)
- **more** - show output one screen at a time
- **exit** - logout of the terminal
- **eject** - eject a cdrom or removable device

Additional Links

- [Commands in PDF format](#)

Grep Command

The **grep** command searches the given file for lines containing a match to the given strings or words. By default, **grep** prints the matching lines.

Use **grep** to search for lines of text that match one or many regular expressions, and outputs only the matching lines.

A regular grep search would have this format:

```
% grep "\<[A-Za-z].*" file
```

Grep Regular Expression	
^	Denotes the beginning of a line
\$	Denotes the end of a line
.	Matches any one characters
*	Matches 0 or more of the previous characters
.*	Matches any number or type of characters
[]	Matches on character for the one listed in the the Square brackets
[^]	Does not match any characters listed
\<, \>	Denotes the beginning and end (respectively) of a word

Required Reading

- [Using the terminal](#)
- [What is grep?](#)
- [Searching with grep](#)
- [Grep command](#)
- [Data manipulation](#)
- [Using grep](#)
- [Getting a grip](#)

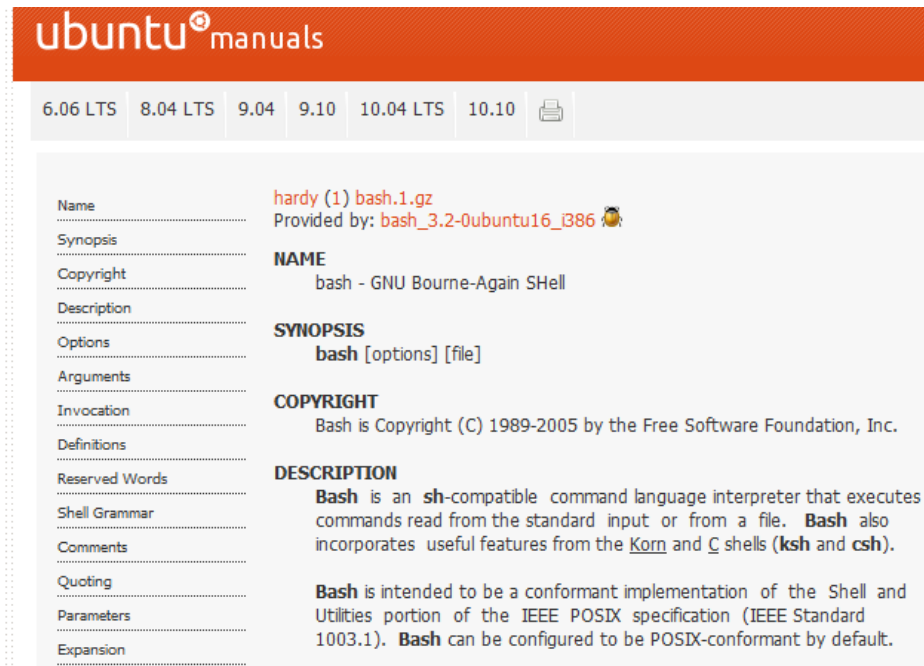
Select **PLAY** below to view a video on grep.



View Video
VideoLesson2GrepCommand(
C2L2S19).mp4

Bash Shell

BASH Shell is a software tool that interprets commands for the GNU operating system. It can run on most Unix-based system and is the default shell for the Linux kernel and Mac OS X. Bash shell allows a computer user to interface with the Linux kernel and other areas of the operating system.



The screenshot shows the Ubuntu manual page for Bash. The header is orange with the text 'ubuntu manuals'. Below the header is a navigation bar with links for different Ubuntu versions: 6.06 LTS, 8.04 LTS, 9.04, 9.10, 10.04 LTS, and 10.10. The main content area is white and contains the following information:

Name	hardy (1) bash.1.gz Provided by: bash_3.2-0ubuntu16_b386
Synopsis	
Copyright	NAME bash - GNU Bourne-Again SHell
Description	
Options	SYNOPSIS bash [options] [file]
Arguments	
Invocation	COPYRIGHT Bash is Copyright (C) 1989-2005 by the Free Software Foundation, Inc.
Definitions	
Reserved Words	DESCRIPTION Bash is an sh -compatible command language interpreter that executes commands read from the standard input or from a file. Bash also incorporates useful features from the Korn and C shells (ksh and csh).
Shell Grammar	
Comments	
Quoting	Bash is intended to be a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification (IEEE Standard 1003.1). Bash can be configured to be POSIX-conformant by default.
Parameters	
Expansion	



View Video
VideoLesson2Bash1(C2L2S2oV1).m
p4

Bash 1



View Video
VideoLesson2Bash2(C2L2S2oV2).m
p4

Bash 2

Required Reading

- [Bash shell programming](#)
- [Bash command list](#)
- [Advanced Bash](#)
- [Bash Unix command](#)

Programming Terms and Concepts

Script

Normally a script is a text file that is a listing of single file commands. A [script](#) can be one line or thousands of lines of commands.

Variable

A [variable](#) is a predefined structure that represents a number, phrase, term or other relevant data.

Expression

In programming, expression refers to a line of source code that returns a value when executed. An expression can comprise any combination of variables, values, operators, and functions.

Process

A process is a program that is being executed and can be identified by its unique PID (process identification number). In other words, it is a program in execution!

- [Linux process](#)
- [Controlling system processes](#)
- [Show running processes](#)
- [Learning Linux process](#)



View Video
VideoLesson2Scripting(C2L2S21V1).
mp4

Scripting



View Video
VideoLesson2Programming(C2L2S
21V2).mp4

Programming Terms

Handbook on Shell Scripting
[Linux Shell Scripting](#)
[Teach Yourself Scripting](#)
[Linux shell for beginners](#)

Shell, Pipe, & Alias

Shell Pipe

The term *pipe* in shell programming refers to the process of directing the output of one script or program to the input of another script or program.

Alias

Simply stated, an alias is just another name for something. For example, a young dog could have an alias of *puppy*.

Responsibilities of the Shell

- Reading inputs and parsing the command line
- Evaluating special characters
- Setting up pipes and background processing
- Handling various signals
- Setting programs up to be executed

Required Reading: Shell Pipe

- [Pipes & Redirectors](#)
- [Intro to Bash Programming](#)
- [Bash Tutorial](#)
- [Named Pipes](#)
- [Bash Shell](#)

Required Reading: Alias

- [Linux alias](#)
- [Using aliases](#)

Required Reading: Shell

- [Shell Script](#)
- [Shell scripting tutorial](#)

Special Shell Characters

Shell scripting uses special characters that have specific meanings:

CHARACTERS	MEANING
\ “ ‘	(backslash, double quotes, single quote) are used for quoting
< >	Greater than and less than signs are used for input/output direction
	The vertical line pipes the output of the command to the left of the pipe symbol " " to the input of the command on the right of the pipe symbol
;	The semicolon separates multiple commands written on a single line
[Space] and [Tab]	A single space and a tabbed space are used to separate the command words
[Newline]	A new line completes a command or set of commands
()	Parenthesis enclose command(s) to be launched in a separate shell (subshell).
{ }	Curly brackets enclose a group of commands to be launched by the current shell. E.g. { dir }. Be sure to include the space before and after the command within the brackets.
&	Ampersand causes the preceding command to execute in the background (i.e., asynchronously, as its own separate process) so that the next command does not wait for its completion

Special Shell Characters Contd.

Shell scripting uses special characters that have specific meanings:

CHARACTERS	MEANING
*	When a filename is expected, the asterisk matches any filename except those starting with a period (or any part of a filename, except the initial period)
?	When a filename is expected, the question sign matches any single character
[]	When a filename is expected, the square brackets match any single character enclosed inside the pair of [].
&&	Ampersands serve as AND characters connecting two commands. For example, command1 && command2 will execute <i>command2</i> only if <i>command1</i> exits with the exit status 0 (no error). Additionally, cat file1 && cat file2 will display <i>file2</i> only if displaying <i>file1</i> succeeded.
	The double vertical functions as an OR statement connecting two commands. For example command1 command2 will execute <i>command2</i> only if <i>command1</i> exits with the exit status of non-zero (with an error). Additionally, cat file1 cat file2 will display <i>file2</i> only if displaying <i>file1</i> did not succeed.
=	The equal sign assigns a value to a variable. Critical: Use no spaces before or after the “=” when assigning a value!

Current List of Special Characters

Review the following links for special characters used in shell scripting:

- [Special characters](#)
- [Escape characters](#)
- [Advanced guide](#)
- [Metacharacters](#)
- [Linux cookbook](#)



Lesson Summary

A Linux Administrator can create powerful scripts to accomplish almost any administrative task. The administrator that is “fluent” in using shell scripting will have the competitive advantage over those who do not!

Shell scripting is the use of shell commands stored in a plain text file that can be executed. Shell scripting is a very powerful tool that may be used to automate many complex computing tasks.

To write commands, programmers may use text editors, which are powerful programs that are compatible with plain text characters and will not add invisible coding or formatting to text. Linux users may choose from a wide variety of text editors including Vi, Vim, and Emacs.

Shell scripting assigns special meaning to certain characters including the asterisk (*), question sign (?), backslash (\) and ampersand (&).

