Useful Linux Commands

Arnon Erba
Agricultural & Resource Economics

Updated October 25, 2019

Contents

Abstract

This document explains how to use several common command-line utilities found on Linux-based, Unix-based, and BSD-based systems. Each command is followed by a short description and some useful examples.

1 Command Structure & Manual Pages

1.1 Command Structure

The structure of a particular command depends on what the command is used for. Command-line utilities generally accept multiple **options** that can be used to control their output or change their behavior. Short options are specified with a single dash and can usually be combined:

- `ls -l`: List files in the current working directory in long format.
- `ps -eF`: List all running processes in extra full format.

Long options are specified with two dashes and cannot be combined:

- `git --version`: Show what version of Git is currently installed.
- `rsync --help`: Display a summary of ways to use Rsync.

Some commands also accept one or more arguments, such as the name of a file:

- `ls folder1`: List the contents of `folder1` instead of the contents of the current directory.
- `rm file.txt`: Permanently delete the file named `file.txt`.

Options can accept their own arguments. Additionally, options and arguments are often combined:

- `git commit -m "Fix broken feature"`: Commit changes in Git with a specific message. The `-m` option requires a text string as its argument.
- `grep -i pattern file2.txt`: Enable case-insensitive matching (`-i`) and search for the text string `pattern` in the file named `file2.txt`.
- `ssh -YC alice@server`: Enable trusted X11 forwarding (`-Y`) and compression (`-C`) and connect to the computer named `server` as the user `alice`.

Note: A list of valid options and arguments is generally provided in the **man page** for the utility.

1.2 Manual Pages

Unix-derived systems provide **man pages** (short for **manual pages**) as the primary form of documentation. The man page for a utility or command can be invoked by typing `man` followed by the name of the command (e.g. `man htop`). Man pages normally open in a pager such as `more` or `less` and can be exited by pressing the lowercase `q` key on the keyboard.

2 File & Directory Management

2.1 `ls` – List Files

The `ls` command is used to list files.

- `ls`: List files in the current working directory.
- `ls -l`: List files in long format (often aliased to just `ll`).
- `ls -la`: List files in long format and show hidden files (dotfiles).
- `ls -lt`: List files in long format and sort by modification time.

2.2 `cd` – Change Directory

The `cd` command is used to change your working directory.

- `cd folder1`: Change to a directory named `folder1`.
- `cd "Folder 2"`: Change to a directory named `Folder 2`. Quotes are required when working with paths that include spaces.
- `cd ..` : Move up one level in the directory structure.
- `cd -` : Change to the previous working directory.

2.3 `pwd` – Print Working Directory

The `pwd` command prints the full path to the current working directory.

2.4 `rm` – Remove Files

The `rm` command is used to remove files and non-empty directories. Use with caution.

- `rm file.txt`: Permanently delete the file named `file.txt`.
- `rm -i file.txt`: Remove the file named `file.txt`, but ask for confirmation first.
- `rm -r "Folder 2"`: Recursively remove `Folder 2` and **all** of its contents.
- `rm -rf "Folder 2"`: Recursively remove `Folder 2` and **all** of its contents **without asking for confirmation**.

2.5 `cp` – Copy Files & Directories

The `cp` command is used to copy files and directories.

- `cp original.txt copy.txt`: Copy the file named `original.txt` to `copy.txt`.
- `cp -r old_folder new_folder`: Recursively copy the folder named `old_folder` to `new_folder`.

2.6 `mv` – Move or Rename Files & Directories

The `mv` command is used to move files and directories. It can also be used to rename files and directories by "moving" them to a different name.

- `mv file2.txt folder1/file2.txt`: Move `file2.txt` into the directory `folder1`.
- `mv old.txt new.txt`: Rename `old.txt` to `new.txt`. **If `new.txt` already exists, it will be silently overwritten.**

2.7 `mkdir` – Make New Directories

The `mkdir` command creates new directories.

- `mkdir folder3`: Create an empty directory named `folder3`.

2.8 `rmdir` – Remove Empty Directories

The `rmdir` command removes empty directories. It is a safer alternative to `rm -f`.

- `rmdir folder3`: Remove the directory named `folder3`, if it is empty.

3 Resource & Process Management

3.1 `ps` – Process Status

The `ps` command is used to list processes. It can be used to quickly identify the PID (process ID) of a running process.

- `ps`: Show all processes owned by the current user and associated with the current terminal.
- `ps -u alice`: Show all processes owned by the user `alice`. Substitute your username here to list every process on the system that you own.
- `ps -u alice -f`: Show all processes owned by the user `alice` in full format.

3.2 `kill` – Kill Process

The `kill` command is used to kill processes by PID.

- `kill 1024`: Kill the process with PID 1024.
- `kill -9 1024`: Forcibly kill the process with PID 1024 if it is not responding.

3.3 `free` – Show Memory Usage

The `free` command is used to show the amount of free and used memory in the system.

- `free -g`: Show the amount of free and used memory in gibibytes.

3.4 `df` – Show Free Disk Space

The `df` command is used to show the amount of free and used disk space across all mounted file systems.

- `df -h`: Show free and used disk space in human-readable format.
- `df -h /home`: Show disk usage on the `/home` file system.

3.5 `du` – Estimate Disk Usage of Files

The `du` command is used to estimate the disk usage of files and directories.

- `du -h file2.txt`: Show the size of `file2.txt` in human-readable format.
- `du -sh folder1`: Show the total size of the directory `folder1` in human-readable format.
- `du -sh * | sort -h`: Show the summarized size of every file and directory inside the current working directory and sort the output by size.

3.6 `htop` – Interactive Resource Monitor

Htop is an interactive process viewer and resource monitor based on the original `top` utility. Htop is controlled by pressing various keys while the program is open:

- `q`: Quit htop and return to the shell.
- `H`: Show/hide threads (shown in green).
- `M`: Sort output by memory usage.
- `P`: Sort output by CPU usage.
- `F9`: Choose a signal to send to the highlighted process using the `kill` utility.

Htop also accepts several options, such as:

- `htop -u alice`: Only show processes owned by the user `alice`. Substitute your username here to monitor your own resource usage in real-time.
- `htop -C`: Start htop in monochrome mode.

4 File Manipulation

4.1 `cat` – Concatenate Files

The `cat` utility can combine text files or print them to standard output.

- `cat file1.txt file2.txt > file3.txt`: Combine the contents of `file1.txt` and `file2.txt` into `file3.txt`.
- `cat file3.txt`: Print the contents of `file3.txt` to standard output.

4.2 `grep` – Search Inside Files

Grep is a powerful tool for searching the contents of text files.

- `grep pattern file2.txt`: Search for the string `pattern` inside `file2.txt`.

A full explanation of how to use grep is beyond the scope of this document, but much more information can be found in the man page.

4.3 `vim` – Text Editor

Vim is a popular command-line text editor. Though complicated to use at first, Vim is full of useful features and is a valuable tool when manipulating files on the command line. An explanation of how to use Vim is beyond the scope of this document.

4.4 `vi` – Text Editor

Vim is based on vi, one of the original Unix text editors. Though less featureful than Vim, vi is sometimes available on systems where Vim is not.

4.5 `nano` – Text Editor

Nano is a friendlier text editor available on many Linux systems. Unlike Vim, Nano displays some of its basic commands on-screen for reference. Commands such as `^X` must be entered as a shortcut using the Control key (e.g. by pressing Control+X).