

# Linux<sup>®</sup>

## Notes for Professionals

### Chapter 1: Getting started with GNU

#### Section 1.1: Useful shortcuts

##### Using The Terminal

The examples in this document assume that you are using a POSIX compliant (such as the Bourne shell).

Large portions of GNU/Linux functionality are achieved using the terminal. Most distributions provide terminal emulators that allow users to interact with a shell from their desktop environment. The line interpreter that executes user inputted commands, **Bash** (Bourne Again Shell) is a common among many Linux distributions and is the default shell for many.

These shortcuts will work if you are using **Bash** with the emacs keybindings (set by default).

##### Open terminal

- **Ctrl + Alt + T** or **Super + T**

##### Cursor movement

- **Ctrl + A** Go to the beginning of the line you are currently typing on.
- **Ctrl + E** Go to the end of the line you are currently typing on.
- **Ctrl + B** Move cursor backward one character on the current line.
- **Alt + B** Move cursor backward one word on the current line.
- **Ctrl + F** Move cursor forward one character on the current line.
- **Alt + F** Move cursor forward one word on the current line.

##### Text manipulation

- **Ctrl + U** Cut the line from the current position to the beginning of the line.
- **Ctrl + E** Cut the line from the current position to the end of the line.
- **Ctrl + X** Cut the line from the beginning of the line, cut the entire line.
- **Ctrl + R** Delete the word before the cursor, adding it to the clipboard.
- **Ctrl + Y** Paste the last thing from the clipboard that you cut recently.
- **Ctrl + W** Delete the last two words before the cursor.
- **Alt + U** Swap the last two words before the cursor.
- **Alt + L** Make lowercase from cursor to end of word.
- **Alt + U** Make uppercase from cursor to end of word.
- **Alt + C** Capitalize to end of word starting at cursor (whole word).
- **Alt + D** Delete to end of word starting at cursor (whole word).
- **Alt + B** Prints the last word written in previous command.
- **Alt + F** Prints the last two characters before the cursor.

##### History access

- **Ctrl + R** Lets you search through previously used commands.
- **Ctrl + G** Leaves history searching mode without running a command.
- **Ctrl + J** Lets you copy current matched command to command line without running it, allowing you to

Linux® Notes for Professionals

### Chapter 2: Detecting Linux distribution name and version

#### Section 2.1: Detect what debian-based distribution you are working in

Just execute **lsb\_release -a**.

On Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux testing (stretch)
Release:       10.0
Codename:      stretch
```

On Ubuntu:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.4 LTS
Release:       14.04
Codename:      trusty
```

In case when you don't have **lsb\_release** installed you may want to try some guessing, for example, there is a file **/etc/issue** that often contains distribution name. For example, on ubuntu:

```
$ cat /etc/issue
Ubuntu 12.04.5 LTS (12.10)
```

Don't use file **/etc/issue** because its contents do not match distribution name.

Note that this will also work on non-Debian-family distributions like Fedora, Arch, or openSUSE — but that **lsb\_release** may not be installed.

#### Section 2.2: Detect what systemd-based distribution you are using

This method will work on modern versions of Arch, CentOS, CoreOS, Debian, Fedora, Mageia, openSUSE, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu, and others. This wide applicability makes it an ideal first approach, with fallback to other methods if you need to also identify older systems.

Look at **/etc/os-release**. In specific, look at variables **NAME**, **VERSION**, **ID**, **VERSION\_ID**, and **PRETTY\_NAME**.

On Fedora, this file might look like:

```
NAME="Fedora"
VERSION="24 (Workstation Edition)"
ID="fedora"
VERSION_ID="24"
PRETTY_NAME="Fedora 24 (Workstation Edition)"
ANSI_COLOR="0;34"
```

Linux® Notes for Professionals

### Chapter 5: Check Disk Space

#### Section 5.1: Investigate Directories For Disk Usage

Sometimes it may be required to find out which directory consuming how much disk space especially when you are used **df -h** and realized your available disk space is low.

**du**:

**du** command summarizes disk usage of the set of files, recursively for directories. It's often uses with **-sh** option:

```
-s, --summarize
    Display only a total for each argument
-h, --human-readable
    Print sizes in human readable format (e.g., 1K 234M 2G)
```

For summarizing disk usages of the files in the current directory we use:

```
du -sh *
```

Example output:

```
57K  documents
20M  downloads
4.0K  music
72K  pictures
4.0K  public
4.0K  templates
4.0K  videos
```

We can also include hidden files with using:

```
du -sh . | grep *
```

Example output:

```
6.3M  .atom
4.0K  .bash_history
4.0K  .bash_logout
8.0K  .bashrc
350M  .cache
100M  .config
12K   .desktop
4.0K  .dove
44K   .emacs
60K   .emacs.d
520K  .gnome
28K   .gnome-2.8
4.0K  .gnome
8.3K  .ICEauthority
8.3K  .local
48K   .nautilus
36K   .nvc
6K    .pki
```

Linux® Notes for Professionals

**50+ pages**  
of professional hints and tricks

# Contents

<b>About</b>	1
<b>Chapter 1: Getting started with GNU/Linux</b>	2
Section 1.1: Useful shortcuts	2
Section 1.2: File Management Commands	3
Section 1.3: Hello World	5
Section 1.4: Basic Linux Utilities	5
Section 1.5: Searching for files by patterns in name/contents	6
Section 1.6: File Manipulation	7
Section 1.7: File/Directory details	8
<b>Chapter 2: Detecting Linux distribution name and version</b>	11
Section 2.1: Detect what debian-based distribution you are working in	11
Section 2.2: Detect what systemd-based distribution you are using	11
Section 2.3: Detect what RHEL / CentOS / Fedora distribution you are working in	12
Section 2.4: Uname - Print information about the current system	13
Section 2.5: Detect basic information about your distro	13
Section 2.6: Using GNU coreutils	13
Section 2.7: Find your linux os (both debian & rpm) name and release number	14
<b>Chapter 3: Getting information on a running Linux kernel</b>	15
Section 3.1: Getting details of Linux kernel	15
<b>Chapter 4: Shell</b>	16
Section 4.1: Changing default shell	16
Section 4.2: Basic Shell Utilities	17
Section 4.3: Create Your Own Command Alias	18
Section 4.4: Locate a file on your system	18
<b>Chapter 5: Check Disk Space</b>	19
Section 5.1: Investigate Directories For Disk Usage	19
Section 5.2: Checking Disk Space	21
<b>Chapter 6: Getting System Information</b>	23
Section 6.1: Statistics about CPU, Memory, Network and Disk (I/O operations)	23
Section 6.2: Using tools like Iscpu and Lshw	23
Section 6.3: List Hardware	24
Section 6.4: Find CPU model/speed information	25
Section 6.5: Process monitoring and information gathering	26
<b>Chapter 7: ls command</b>	28
Section 7.1: Options for ls command	28
Section 7.2: ls command with most used options	28
<b>Chapter 8: File Compression with 'tar' command</b>	30
Section 8.1: Compress a folder	30
Section 8.2: Extract a folder from an archive	30
Section 8.3: List contents of an archive	30
Section 8.4: List archive content	31
Section 8.5: Compress and exclude one or multiple folder	31
Section 8.6: Strip leading components	31
<b>Chapter 9: Services</b>	32
Section 9.1: List running service on Ubuntu	32
Section 9.2: Systemd service management	32

<b>Chapter 10: Managing Services</b>	33
Section 10.1: Diagnosing a problem with a service	33
Section 10.2: Starting and Stopping Services	33
Section 10.3: Getting the status of a service	34
<b>Chapter 11: Modifying Users</b>	35
Section 11.1: Setting your own password	35
Section 11.2: Setting another user's password	35
Section 11.3: Adding a user	35
Section 11.4: Removing a user	35
Section 11.5: Removing a user and its home folder	35
Section 11.6: Listing groups the current user is in	35
Section 11.7: Listing groups a user is in	35
<b>Chapter 12: LAMP Stack</b>	36
Section 12.1: Installing LAMP on Arch Linux	36
Section 12.2: Installing LAMP on Ubuntu	37
Section 12.3: Installing LAMP stack on CentoOS	38
<b>Chapter 13: tee command</b>	40
Section 13.1: Write output to stdout, and also to a file	40
Section 13.2: Write output from the middle of a pipe chain to a file and pass it back to the pipe	40
Section 13.3: write the output to multiple files	40
Section 13.4: Instruct tee command to append to the file	40
<b>Chapter 14: Secure Shell (SSH)</b>	42
Section 14.1: Connecting to a remote server	42
Section 14.2: Installing OpenSSH suite	42
Section 14.3: Configuring an SSH server to accept connections	43
Section 14.4: Passwordless connection (using a key pair)	43
Section 14.5: Generate public and private key	43
Section 14.6: Disable ssh service	43
<b>Chapter 15: SCP</b>	45
Section 15.1: Secure Copy	45
Section 15.2: Basic Usage	45
<b>Chapter 16: GnuPG (GPG)</b>	46
Section 16.1: Exporting your public key	46
Section 16.2: Create and use a GnuPG key quickly	46
<b>Chapter 17: Network Configuration</b>	47
Section 17.1: Local DNS resolution	47
Section 17.2: Configure DNS servers for domain name resolution	47
Section 17.3: See and manipulate routes	47
Section 17.4: Configure a hostname for some other system on your network	48
Section 17.5: Interface details	49
Section 17.6: Adding IP to an interface	50
<b>Chapter 18: Midnight Commander</b>	52
Section 18.1: Midnight Commander function keys in browsing mode	52
Section 18.2: Midnight Commander function keys in file editing mode	52
<b>Chapter 19: Change root (chroot)</b>	54
Section 19.1: Requirements	54
Section 19.2: Manually changing root in a directory	54
Section 19.3: Reasons to use chroot	55
<b>Chapter 20: Package Managers</b>	56

<u>Section 20.1: How to update packages with the apt package manager</u>	56
<u>Section 20.2: How to install a package with the pacman package manager</u>	56
<u>Section 20.3: How to update packages with the pacman package manager</u>	56
<u>Section 20.4: How to update packages with yum</u>	57
<b><u>Chapter 21: Compiling the Linux kernel</u></b>	58
<u>Section 21.1: Compilation of Linux Kernel on Ubuntu</u>	58
<b><u>Credits</u></b>	59
<b><u>You may also like</u></b>	61

# About

Please feel free to share this PDF with anyone for free,  
latest version of this book can be downloaded from:  
<https://goalkicker.com/LinuxBook>

This *Linux® Notes for Professionals* book is compiled from [Stack Overflow Documentation](#), the content is written by the beautiful people at Stack Overflow. Text content is released under Creative Commons BY-SA, see credits at the end of this book whom contributed to the various chapters. Images may be copyright of their respective owners unless otherwise specified

This is an unofficial free book created for educational purposes and is not affiliated with official Linux® group(s) or company(s) nor Stack Overflow. All trademarks and registered trademarks are the property of their respective company owners

The information presented in this book is not guaranteed to be correct nor accurate, use at your own risk

Please send feedback and corrections to [web@petercv.com](mailto:web@petercv.com)

# Chapter 1: Getting started with GNU/Linux

## Section 1.1: Useful shortcuts

### Using The Terminal

The examples in this document assume that you are using a POSIX-compliant (such as **bash**, **sh**, **zsh**, **ksh**) shell.

Large portions of GNU/Linux functionality are achieved using the terminal. Most distributions of Linux include terminal emulators that allow users to interact with a shell from their desktop environment. A shell is a command-line interpreter that executes user inputted commands. **Bash** (Bourne Again SHell) is a common default shell among many Linux distributions and is the default shell for macOS.

These shortcuts will work if you are using **Bash** with the *emacs* keybindings (set by default):

### Open terminal

- `Ctrl + Alt + T` or `Super + T`

### Cursor movement

- `Ctrl + A` Go to the beginning of the line you are currently typing on.
- `Ctrl + E` Go to the end of the line you are currently typing on.
- `Ctrl + XX` Move between the beginning of the line and the current position of the cursor.
- `Alt + F` Move cursor forward one word on the current line.
- `Alt + B` Move cursor backward one word on the current line.
- `Ctrl + F` Move cursor forward one character on the current line.
- `Ctrl + B` Move cursor backward one character on the current line.

### Text manipulation

- `Ctrl + U` Cut the line from the current position to the beginning of the line, adding it to the clipboard. If you are at the end of the line, cut the entire line.
- `Ctrl + K` Cut the line from the current position to the end of the line, adding it to the clipboard. If you are at the beginning of the line, cut the entire line.
- `Ctrl + W` Delete the word before the cursor, adding it to the clipboard.
- `Ctrl + Y` Paste the last thing from the clipboard that you cut recently (undo the last delete at the **current** cursor position).
- `Alt + T` Swap the last two words before the cursor.
- `Alt + L` Make lowercase from cursor to end of word.
- `Alt + U` Make uppercase from cursor to end of word.
- `Alt + C` Capitalize to end of word starting at cursor (whole word if cursor is at the beginning of word).
- `Alt + D` Delete to end of word starting at cursor (whole word if cursor is at the beginning of word).
- `Alt + .` Prints the last word written in previous command.
- `Ctrl + T` Swap the last two characters before the cursor.

### History access

- `Ctrl + R` Lets you search through previously used commands.
- `Ctrl + G` Leave history searching mode without running a command.
- `Ctrl + J` Lets you copy current matched command to command line without running it, allowing you to

make modifications before running the command.

- **Alt + R** Revert any changes to a command you've pulled from your history, if you've edited it.
- **Ctrl + P** Shows last executed command, i.e. walk back through the command history (Similar to up arrow).
- **Ctrl + N** Shows next executed command, i.e. walk forward through the command history (Similar to down arrow).

### Terminal control

- **Ctrl + L** Clears the screen, similar to the clear command.
- **Ctrl + S** Stop all output to the screen. This is useful when running commands with lots of long output. But this doesn't stop the running command.
- **Ctrl + Q** Resume output to the screen after stopping it with Ctrl+S.
- **Ctrl + C** End currently running process and return the prompt.
- **Ctrl + D** Log out of the current shell session, similar to the exit or logout command. In some commands, acts as End of File signal to indicate that a file end has been reached.
- **Ctrl + Z** Suspends (pause) currently running foreground process, which returns shell prompt. You can then use `bg` command allowing that process to run in the background. To again bring that process to foreground, use `fg` command. To view all background processes, use `jobs` command.
- **Tab** Auto-complete files and directory names.
- **Tab Tab** Shows all possibilities, when typed characters doesn't uniquely match to a file or directory name.

### Special characters

- **Ctrl + H** Same as Backspace.
- **Ctrl + J** Same as Return (historically Line Feed).
- **Ctrl + M** Same as Return (historically Carriage Return).
- **Ctrl + I** Same as Tab.
- **Ctrl + G** Bell Character.
- **Ctrl + @** Null Character.
- **Esc** Deadkey equivalent to the **Alt** modifier.

### Close Terminal

- **Ctrl + Shift + W** To close terminal tab.
- **Ctrl + Shift + Q** To close entire terminal.

Alternatively, you can switch to the *vi* keybindings in **bash** using `set -o vi`. Use `set -o emacs` to switch back to the *emacs* keybindings.

## Section 1.2: File Management Commands

Linux uses some conventions for present and parent directories. This can be a little confusing for beginners.

Whenever you are in a terminal in Linux, you will be in what is called the *current working directory*. Often your command prompt will display either the full working directory, or just the last part of that directory. Your prompt could look like one of the following:

```
user@host ~/somedir $
user@host somedir $
user@host /home/user/somedir $
```

which says that your current working directory is `/home/user/somedir`.

In Linux `..` represents the parent directory and `.` represents the current directory.

Therefore, if the current directory is `/home/user/somedir`, then `cd ../somedir` will not change the working directory.

The table below lists some of the most used file management commands

### Directory navigation

Command	Utility
<code>pwd</code>	Get the full path of the current working directory.
<code>cd -</code>	Navigate to the last directory you were working in.
<code>cd ~</code> or just <code>cd</code>	Navigate to the current user's home directory.
<code>cd ..</code>	Go to the parent directory of current directory (mind the space between <code>cd</code> and <code>..</code> )

### Listing files inside a directory

Command	Utility
<code>ls -l</code>	List the files and directories in the current directory in long (table) format (It is recommended to use <code>-l</code> with <code>ls</code> for better readability).
<code>ls -ld dir-name</code>	List information about the directory <code>dir-name</code> instead of its contents.
<code>ls -a</code>	List all the files including the hidden ones (File names starting with a <code>.</code> are hidden files in Linux).
<code>ls -F</code>	Appends a symbol at the end of a file name to indicate its type ( <code>*</code> means executable, <code>/</code> means directory, <code>@</code> means symbolic link, <code>=</code> means socket, <code> </code> means named pipe, <code>&gt;</code> means door).
<code>ls -lt</code>	List the files sorted by last modified time with most recently modified files showing at the top (remember <code>-l</code> option provides the long format which has better readability).
<code>ls -lh</code>	List the file sizes in human readable format.
<code>ls -lR</code>	Shows all subdirectories recursively.
<code>tree</code>	Will generate a tree representation of the file system starting from the current directory.

### File/directory create, copy and remove

Command	Utility
<code>cp -p source destination</code>	Will copy the file from <b>source</b> to <i>destination</i> . <code>-p</code> stands for preservation. It preserves the original attributes of file while copying like file owner, timestamp, group, permissions etc.
<code>cp -R source_dir destination_dir</code>	Will copy source directory to specified destination recursively.
<code>mv file1 file2</code>	In Linux there is no <b>rename</b> command as such. Hence <code>mv</code> moves/renames the file1 to file2.
<code>rm -i filename</code>	Asks you before every file removal for confirmation. <b>IF YOU ARE A NEW USER TO LINUX COMMAND LINE, YOU SHOULD ALWAYS USE <code>rm -i</code></b> . You can specify multiple files.
<code>rm -R dir-name</code>	Will remove the directory <code>dir-name</code> recursively.
<code>rm -rf dir-name</code>	Will remove the directory <b>dir</b> recursively, ignoring non-existent files and will never prompt for anything. <b>BE CAREFUL USING THIS COMMAND!</b> You can specify multiple directories.
<code>rmdir dir-name</code>	Will remove the directory <code>dir-name</code> , if it's empty. This command can only remove empty directories.
<code>mkdir dir-name</code>	Create a directory <code>dir-name</code> .
<code>mkdir -p dir-name/dir-name</code>	Create a directory hierarchy. Create parent directories as needed, if they don't exist. You can specify multiple directories.
<code>touch filename</code>	Create a file <code>filename</code> , if it doesn't exist, otherwise change the timestamp of the file to current time.

### File/directory permissions and groups

Command	Utility
---------	---------



<b>chmod</b> <specification> filename	Change the file permissions. Specifications = u user, g group, o other, + add permission, - remove, r read, w write, x execute.
<b>chmod</b> -R <specification> dir-name	Change the permissions of a directory recursively. To change permission of a directory and everything within that directory, use this command.
<b>chmod</b> go+=r myfile	Add read permission for the owner and the group.
<b>chmod</b> a +rwx myfile	Allow all users to read, write or execute myfile.
<b>chmod</b> go -r myfile	Remove read permission from the group and others.
<b>chown</b> owner1 filename	Change ownership of a file to user owner1.
<b>chgrp</b> grp_owner filename	Change primary group ownership of file filename to group grp_owner.
<b>chgrp</b> -R grp_owner dir-name	Change primary group ownership of directory dir-name to group grp_owner recursively. To change group ownership of a directory and everything within that directory, use this command.

## Section 1.3: Hello World

Type the following code into your terminal, then press `Enter`:

```
echo "Hello World"
```

This will produce the following output:

```
Hello World
```

## Section 1.4: Basic Linux Utilities

Linux has a command for almost any tasks and most of them are intuitive and easily interpreted.

### Getting Help in Linux

Command	Usability
<b>man</b> <name>	Read the manual page of <name>.
<b>man</b> <section> <name>	Read the manual page of <name>, related to the given section.
<b>man</b> -k <editor>	Output all the software whose man pages contain <editor> keyword.
<b>man</b> -K <keyword>	Outputs all man pages containing <keyword> within them.
<b>apropos</b> <editor>	Output all the applications whose one line description matches the word <i>editor</i> . When <b>not able to recall</b> the name of the application, use this command.
<b>help</b>	In Bash shell, this will display the list of all available bash commands.
<b>help</b> <name>	In Bash shell, this will display the info about the <name> bash command.
<b>info</b> <name>	View all the information about <name>.
<b>dpkg</b> -l	Output a list of all installed packages on a Debian-based system.
<b>dpkg</b> -L packageName	Will list out the files installed and path details for a given package on Debian.
<b>dpkg</b> -l   <b>grep</b> -i <edit>	Return all .deb installed packages with <edit> irrespective of cases.
<b>less</b> /var/lib/dpkg/available	Return descriptions of all available packages.
<b>whatis</b> vim	List a one-line description of vim.
<command-name> --help	Display usage information about the <tool-name>. Sometimes <b>command</b> -h also works, but not for all commands.

### User identification and who is who in Linux world

Command	Usability
<b>hostname</b>	Display hostname of the system.

<b>hostname</b>	-f Displays Fully Qualified Domain Name (FQDN) of the system.
<b>passwd</b>	Change password of current user.
<b>whoami</b>	Username of the users logged in at the terminal.
<b>who</b>	List of all the users currently logged in as a user.
<b>w</b>	Display current system status, time, duration, list of users currently logged in on system and other user information.
<b>last</b>	Who recently used the system.
<b>last</b> root	When was the last time <b>root</b> logged in as user.
<b>lastb</b>	Shows all bad login attempts into the system.
<b>chmod</b>	Changing permissions - read,write,execute of a file or directory.

## Process related information

Command	Usability
<b>top</b>	List all processes sorted by their current system resource usage. Displays a continually updated display of processes (By default 3 seconds). Use q key to exit top.
<b>ps</b>	List processes currently running on current shell session
<b>ps -u</b> root	List all of the processes and commands root is running
<b>ps</b> aux	List all the processes by all users on the current system

## Section 1.5: Searching for files by patterns in name/contents

A common task of someone using the Linux Command Line (shell) is to search for files/directories with a certain name or containing certain text. There are 2 commands you should familiarise yourself with in order to accomplish this:

### Find files by name

```
find /var/www -name '*.css'
```

This will print out the full path/filename to all files under `/var/www` that end in `.css`. Example output:

```
/var/www/html/text-cursor.css
/var/www/html/style.css
```

For more info:

```
man find
```

### Find files containing text

```
grep font /var/www/html/style.css
```

This will print all lines containing the pattern `font` in the specified file. Example output:

```
font-weight: bold;
font-family: monospace;
```

Another example:

```
grep font /var/www/html/
```