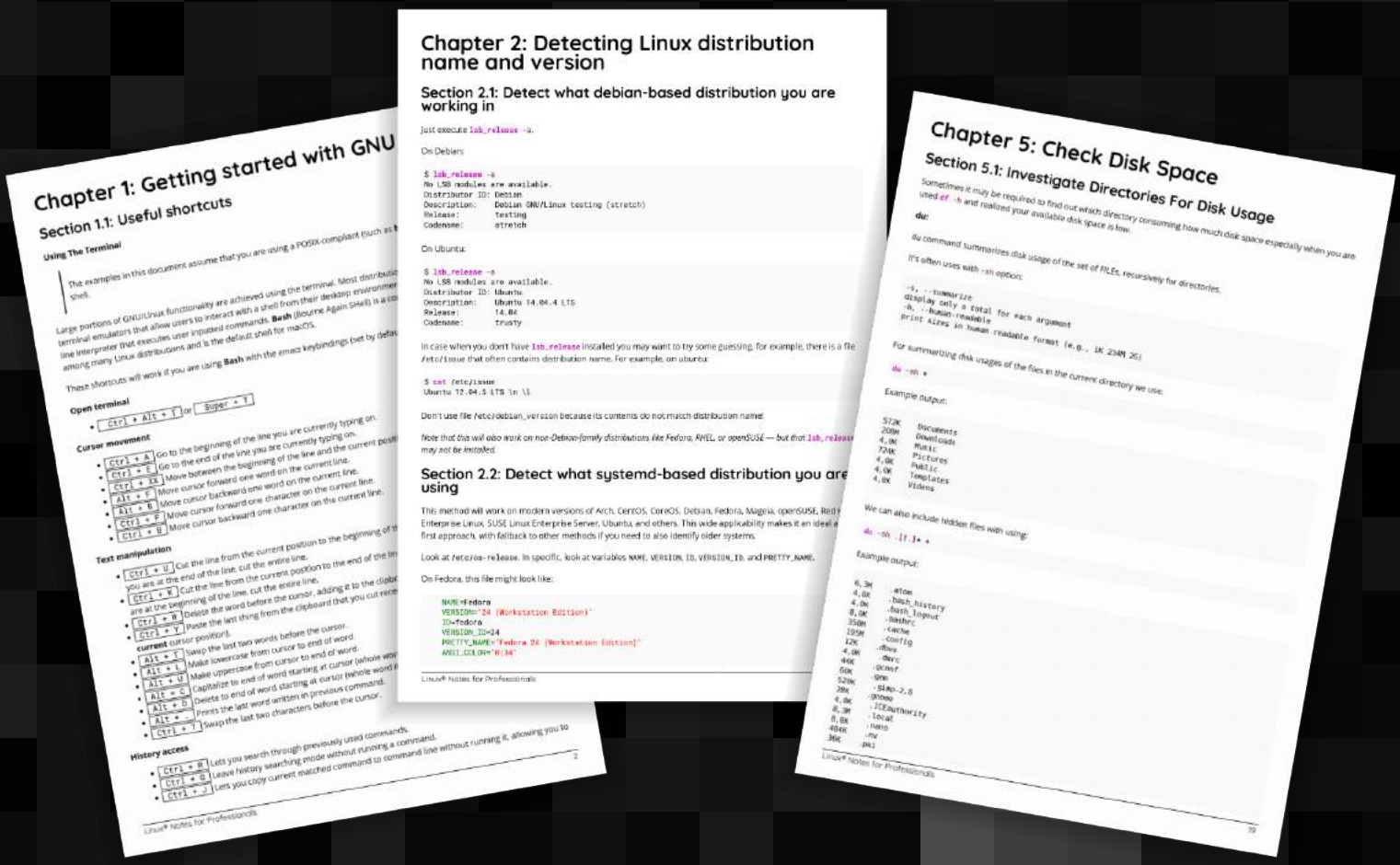
Linux®

Notes for Professionals



50+ pages

of professional hints and tricks

Contents

About

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

Listing files inside a directory

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

File/directory create, copy and remove

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

File/directory permissions and groups

Command	Utility

`chmod <specification> filename`	Change the file permissions. Specifications = `u` user, `g` group, `o` other, `+` add permission, `-` remove, `r` read, `w` write,`x` execute.
`chmod -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.
`chmod go=+r myfile`	Add read permission for the owner and the group.
`chmod a +rwx myfile`	Allow all users to read, write or execute `myfile`.
`chmod go -r myfile`	Remove read permission from the group and others.
`chown owner1 filename`	Change ownership of a file to user `owner1`.
`chgrp grp_owner filename`	Change primary group ownership of file `filename` to group `grp_owner`.
`chgrp -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
`man <name>`	Read the manual page of <name>.
`man <section> <name>`	Read the manual page of <name>, related to the given section.
`man -k <editor>`	Output all the software whose man pages contain <editor> keyword.
`man -K <keyword>`	Outputs all man pages containing <keyword> within them.
`apropos <editor>`	Output all the applications whose one line description matches the word *editor*. When **not able to recall** the name of the application, use this command.
`help`	In Bash shell, this will display the list of all available bash commands.
`help <name>`	In Bash shell, this will display the info about the <name> bash command.
`info <name>`	View all the information about <name>.
`dpkg -l`	Output a list of all installed packages on a Debian-based system.
`dpkg -L packageName`	Will list out the files installed and path details for a given package on Debian.
`dpkg -l | grep -i <edit>`	Return all .deb installed packages with <edit> irrespective of cases.
`less /var/lib/dpkg/available`	Return descriptions of all available packages.
`whatis vim`	List a one-line description of vim.
`<command-name> --help`	Display usage information about the <tool-name>. Sometimes `command -h` also works, but not for all commands.

User identification and who is who in Linux world

Command	Usability
`hostname`	Display hostname of the system.

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

Process related information

Command	Usability
`top`	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.
`ps`	List processes currently running on current shell session
`ps -u root`	List all of the processes and commands root is running
`ps 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 and 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/
```