

OPEN-SOURCE EBOOK

++101 LINUX COMMANDS

BOBBY ILIEV

Table of Contents

101 Linux commands Open-source eBook	16
Hacktoberfest	17
About me	18
Ebook PDF Generation Tool	20
Book Cover	21
License	22
 The ls command	 23
 The cd command	 26
 The cat command	 29
 The tac command	 32
 The head command	 34
 The tail command	 36
 The pwd command	 39
 The touch Command	 41
 The cal Command	 44
 The bc command	 47

The df command	51
The help command	55
Syntax	56
Options	57
Examples of uses:	58
The factor command	59
Syntax	60
Options	61
Examples	62
The uname command	63
Syntax:	64
Examples	65
Options	66
The mkdir command	67
Syntax	68
Examples	69
Options	70
The gzip command	71
Usage	72
Compress a file	73
Decompress a file	74
Compress multiple files:	75
Decompress multiple files:	76

The `whatis` command

The `whatis` command is used to display one-line manual page descriptions for commands. It can be used to get a basic understanding of what a (unknown) command is used for.

Examples of uses:

1. To display what `ls` is used for:

```
whatis ls
```

2. To display the use of all commands which start with `make`, execute the following:

```
whatis -w make*
```

Syntax:

```
whatis [-OPTION] [KEYWORD]
```

Additional Flags and their Functionalities:

Short Flag	Long Flag	Description
<code>-d</code>	<code>--debug</code>	Show debugging messages
<code>-r</code>	<code>--regex</code>	Interpret each keyword as a regex
<code>-w</code>	<code>--wildcard</code>	The keyword(s) contain wildcards

The `who` command

The `who` command lets you print out a list of logged-in users, the current run level of the system and the time of last system boot.

Examples

1. Print out all details of currently logged-in users

```
who -a
```

2. Print out the list of all dead processes

```
who -d -H
```

Syntax:

```
who [options] [filename]
```

Additional Flags and their Functionalities

Short Flag	Description
<code>-r</code>	prints all the current runlevel
<code>-d</code>	print all the dead processes
<code>-q</code>	print all the login names and total number of logged on users
<code>-h</code>	print the heading of the columns displayed

Short Flag	Description
-b	print the time of last system boot

018-the-free-command.md

The `free` command

The `free` command in Linux/Unix is used to show memory (RAM/SWAP) information.

Usage

Show memory usage

Action: --- Output the memory usage - available and used, as well as swap

Details: --- Outputted values are not human-readable (are in bytes)

Command:

```
free
```

Idle time (Idle time is minutes if it is a single integer, hours and minutes if a : (colon) is present, or days and hours if a “d” is present.)

Login time

Site-specific information

Longer Format

A longer format is used by the finger command whenever a list of user's names is given. (Account names as well as first and last names of users are accepted.) This format is multiline, and includes all the information described above along with the following:

User's \$HOME directory

User's login shell

Contents of the .plan file in the user's \$HOME directory

Contents of the .project file in the user's \$HOME directory

The `groups` command

In Linux, there can be multiple users (those who use/operate the system), and groups (a collection of users). Groups make it easy to manage users with the same security and access privileges. A user can be part of different groups.

Important Points:

The `groups` command prints the names of the primary and any supplementary groups for each given username, or the current process if no names are given. If more than one name is given, the name of each user is printed before the list of that user's groups and the username is separated from the group list by a colon.

Syntax:

```
groups [username]
```

Example 1

Provided with a username

```
groups demon
```

In this example, username `demon` is passed with `groups` command and the output shows the groups in which the user `demon` is present, separated by a colon.

Example 2

When no username is passed then this will display the group membership for the current user:

```
groups
```

Here the current user is demon . So when we run the **groups** command without arguments we get the groups in which demon is a user.

Example 3

Passing root with groups command:

```
$demon# groups
```

Note: Primary and supplementary groups for a process are normally inherited from its parent and are usually unchanged since login. This means that if you change the group database after logging in, groups will not reflect your changes within your existing login session. The only options are **-help** and **-version**.

The `man` command

The `man` command is used to display the manual of any command that we can run on the terminal. It provides information like: DESCRIPTION, OPTIONS, AUTHORS and more.

Examples:

1. Man page for printf:

```
man printf
```

2. Man page section 2 for intro:

```
man 2 intro
```

3. Viewing the Manual for a Local File (using the `-l` flag):

```
man -l [LOCAL-FILE]
```

Syntax:

```
man [SECTION-NUM] [COMMAND NAME]
```

This is a sample from "101 Linux Commands eBook" by Bobby Iliev the Hacktoberfest
community.

For more information, [Click here](#).