

OPEN-SOURCE EBOOK

# ++101 LINUX COMMANDS

BOBBY ILIEV

# Table of Contents

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

<b>The df command</b>	<b>51</b>
<b>The help command</b>	<b>55</b>
Syntax	56
Options	57
Examples of uses:	58
<b>The factor command</b>	<b>59</b>
Syntax	60
Options	61
Examples	62
<b>The uname command</b>	<b>63</b>
Syntax:	64
Examples	65
Options	66
<b>The mkdir command</b>	<b>67</b>
Syntax	68
Examples	69
Options	70
<b>The gzip command</b>	<b>71</b>
<b>Usage</b>	<b>72</b>
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
------------	-----------	-------------

-d	--debug	Show debugging messages
-r	--regex	Interpret each keyword as a regex
-w	--wildcard	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
-r	prints all the current runlevel
-d	print all the dead processes
-q	print all the login names and total number of logged on users
-h	print the heading of the columns displayed

Short Flag	Description
<b>-b</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](#).