

OPEN-SOURCE EBOOK

++101 LINUX COMMANDS

BOBBY ILIEV

Table of Contents

101 Linux commands Open-source eBook

This is an open-source eBook with 101 Linux commands that everyone should know. No matter if you are a DevOps/SysOps engineer, developer, or just a Linux enthusiast, you will most likely have to use the terminal at some point in your career.

Additional Flags and their Functionalities:

Short Flag	Long Flag	Description
-a	-	Sort by memory usage.
-b	-	Batch mode operation. Starts top in 'Batch mode', which could be useful for sending output from top to other programs or to a file. In this mode, top will not accept input and runs until the iterations limit you've set with the '-n' command-line option or until killed.
-h	-	<code>top --user {user_name}</code> Only display processes owned by user.
-U	-user	Help.
-u	-	This is an alias equivalent to: -o cpu -O time.

The `sl` command

The `sl` command in Linux is a humorous program that runs a steam locomotive(`sl`) across your terminal.



Installation

Install the package before running.

```
sudo apt install sl
```

Syntax

```
| sl
```

The `echo` command

The `echo` command lets you display the line of text/string that is passed as an argument

Examples:

1. To Show the line of text or string passed as an argument:

```
echo Hello There
```

2. To show all files/folders similar to the `ls` command:

```
echo *
```

3. To save text to a file named `foo.bar`:

```
echo "Hello There" > foo.bar
```

4. To append text to a file named `foo.bar`:

```
echo "Hello There" >> foo.bar
```

Syntax:


```
echo [option] [string]
```

It is usually used in shell scripts and batch files to output status text to the screen or a file. The **-e** used with it enables the interpretation of backslash escapes

Additional Options and their Functionalities:

Option	Description
--------	-------------

\b	removes all the spaces in between the text
\c	suppress trailing new line with backspace interpreter '-e' to continue without emitting new line.
\n	creates new line from where it is used
\t	creates horizontal tab spaces
\r	carriage returns with backspace interpreter '-e' to have specified carriage return in output
\v	creates vertical tab spaces
\a	alert returns with a backspace interpreter '-e' to have sound alert
-n	omits echoing trailing newline .

The `w` command

The `w` command displays information about the users that are currently active on the machine and their [processes](#).

Examples:

1. Running the `w` command without [arguments](#) shows a list of logged on users and their processes.

```
w
```

2. Show information for the user named *hope*.

```
w hope
```

Syntax:

```
finger [-l] [-m] [-p] [-s] [username]
```

Additional Flags and their Functionalities:

Short Flag	Long Flag	Description
<code>-h</code>	<code>--no-header</code>	Don't print the header.

Short Flag	Long Flag	Description
<code>-u</code>	<code>--no-current</code>	Ignores the username while figuring out the current process and cpu times. <i>(To see an example of this, switch to the root user with <code>su</code> and then run both <code>w</code> and <code>w -u</code>.)</i>
<code>-s</code>	<code>--short</code>	Display abbreviated output <i>(don't print the login time, JCPU or PCPU times)</i> .
<code>-f</code>	<code>--from</code>	Toggle printing the from <i>(remote hostname)</i> field. The default as released is for the from field to not be printed, although your system administrator or distribution maintainer may have compiled a version where the from field is shown by default.
<code>--help</code>	<code>-</code>	Display a help message, and exit.
<code>-V</code>	<code>--version</code>	Display version information, and exit.
<code>-o</code>	<code>--old-style</code>	Old style output <i>(prints blank space for idle times less than one minute)</i> .
<code>user</code>	<code>-</code>	Show information about the specified the user only.

Additional Information

The header of the output shows (in this order): the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.

The following entries are displayed for each user:

- login name the tty
- name the remote
- host they are
- logged in from the amount of time they are logged in their
- idle time JCPU

- PCPU
- command line of their current process

The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running background jobs.

The PCPU time is the time used by the current process, named in the "what" field.

The `whoami` command

The `whoami` command displays the username of the current effective user. In other words it just prints the username of the currently logged-in user when executed.

To display your effective user id just type `whoami` in your terminal:

```
manish@godsmack:~$ whoami
# Output:
manish
```

Syntax:

```
whoami [-OPTION]
```

There are only two options which can be passed to it :

`--help`: Used to display the help and exit

Example:

```
whoami --help
```

Output:

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

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