

TOWiOS

The Only Way is Open Source

LINUX COMMAND LINE INTERFACE

UBUNTU 23.04 LTS

November 2023



Note:

The following websites have been used for reference:

- <https://explainshell.com/explain?cmd=ls>
- <https://www.fosslinux.com/103546/the-beginners-guide-to-using-terminal-on-linux-mint.htm>
- https://linuxcommand.org/lc3_lts0010.php

The following information is not considered 'perfect' and is purely for testing purposes. If you choose to utilise the information provided you do so at your own risk.

TABLE OF CONTENTS

00: INTRODUCTION TO THE CLI.....	3
ACCESSING THE COMMAND LINE INTERFACE.....	3
THE CLI.....	3
BASIC INTRODUCTORY COMMANDS.....	4
01: SYSTEM UPDATE AND UPGRADE.....	5
UPDATE REPOSITORIES.....	5
UPGRADE THE INSTALLED PACKAGES ON THE SYSTEM.....	5
IDENTIFY DISTRO VERSION.....	6
UPGRADE THE DISTRO.....	6

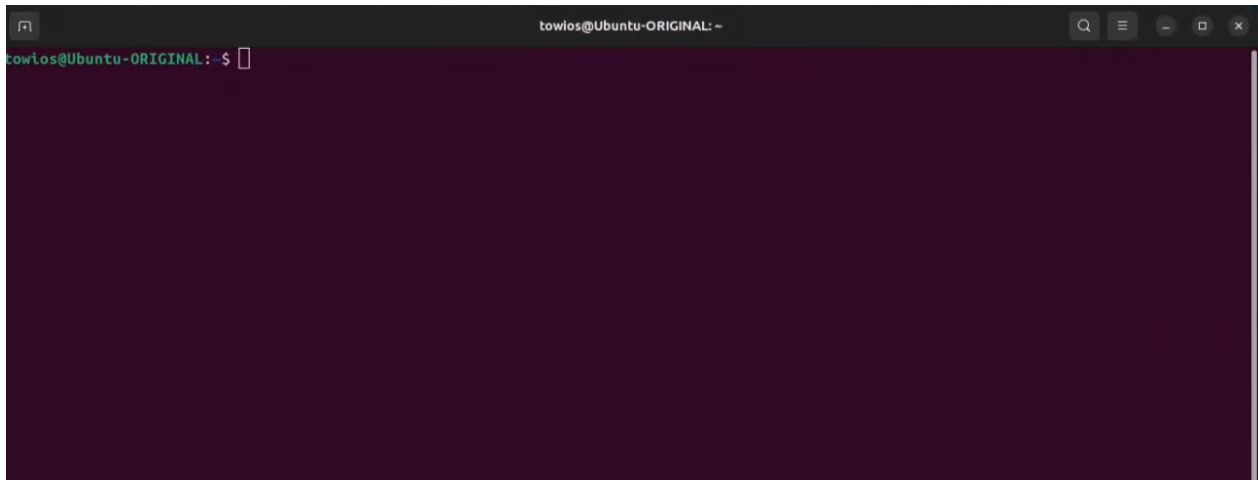
00: INTRODUCTION TO THE CLI

ACCESSING THE COMMAND LINE INTERFACE

To gain access to the Linux Command Line Interface (also known as the CLI), load the system's default terminal emulator program:

- Use the keyboard shortcut “**Ctrl + Alt + T**”,
OR
- Within the menu, search for the application named: ‘**terminal**’.

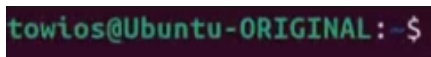
The ‘Terminal’:



The terminal application consists of a command line interface, and a shell. The command line interface (often referred to as the command prompt) is where you type your commands. While the shell (which you can't see) is the program that interprets and executes those commands.

THE CLI

The command line interface can be identified by the coloured text.

A close-up screenshot of the terminal prompt. The text 'towios@Ubuntu-ORIGINAL: ~\$' is displayed. 'towios' is green, '@Ubuntu-ORIGINAL' is blue, ': ~' is cyan, and '\$' is red.

This comprises of:

- The username of the current user logged in = “**towios**” .
- The computer’s hostname = “**Ubuntu-ORIGINAL**” .
- The current working directory. (Which is ‘home’ in this instance) “**~**”.
- “**\$**” = Which indicates that the terminal is ready to accept commands.

BASIC INTRODUCTORY COMMANDS

whoami

Used to display the username of the currently logged-in user. The command provides the name of the user associated with the current session or terminal.

pwd

Short for "print working directory". Displaying the current working directory, which is the directory you are currently located in within the file system. When you run the pwd command, it provides the full path to the directory where you are currently working. This can be helpful for orienting yourself in the file system and understanding your current location.

ls

Lists the files and directories within a specified directory. It provides a way to view the contents of a directory and obtain information about the files and directories it contains.

The ls command can be used with various options and arguments to customise the listing.

ls [*options*] [*directory*]

- *options* are optional and can modify the behaviour of the ls command, allowing you to control the output format and details displayed.
- *directory* is also optional and specifies the directory you want to list. If not provided, the command lists the contents of the current working directory.

Some common options for the ls command include:

- **-l**: List in long format, providing detailed information about each file and directory, including permissions, owner, group, size, and modification time.
- **-a**: List all files, including hidden files and directories (those whose names begin with a dot).
- **-h**: Print sizes in a human-readable format (e.g., "1K" instead of "1024").
- **-R**: Recursively list the contents of directories and their subdirectories.

cd

Stands for "change directory." When you run the cd command, you specify the directory to which you want to navigate, and your shell will move to that location. For example, `cd directory_name`

- To move up one level in the directory tree (to the parent directory), you can use two periods (..) as follows: `cd ..`
- To return to your home directory, you can use the tilde (~) symbol: `cd ~`
- To quickly switch to your previous directory (often referred to as the "last visited" directory), you can use the following command: `cd -`

touch

A simple utility used to create new, empty files with the specified names or update the access and modification timestamps of existing files. It is a versatile command that can serve several purposes, including creating files and changing file timestamps.

echo "\$SHELL"

Used to display the path to the current user's default shell. Let's break down the components of this command:

- **echo**: This is a command that is used to print or display text in the terminal.
- **"\$SHELL"**: This is a shell variable. The \$SHELL variable stores the path to the user's default shell. When you enclose it in double quotes ("SHELL"), it ensures that the value of the variable is treated as a single entity and prevents any word-splitting or other interpretation of the value.

01: SYSTEM UPDATE AND UPGRADE

UPDATE REPOSITORIES

The **sudo apt-get update** command in Ubuntu and other Debian-based Linux distributions is used to refresh the local package database on your system. It does not upgrade the installed packages; instead, it retrieves the latest information about the packages available in the repositories specified in your system's package sources.

Here's a breakdown of what happens when you run **sudo apt-get update**:

1. Refresh Package Lists:

- The command contacts the package repositories listed in the **/etc/apt/sources.list** file and in files within the **/etc/apt/sources.list.d/** directory.
- It retrieves information about the latest versions of software packages and their dependencies available in those repositories.

2. Update the Local Package Database:

- The retrieved information is used to update the local package database on your system.
- This database is crucial for the package manager (apt in this case) to know what packages are available, what versions are current, and what dependencies exist.

3. No Installation or Upgrade:

- Importantly, **sudo apt-get update** does not install or upgrade any packages on your system. It's purely an information-fetching operation.

By regularly running **sudo apt-get update**, you ensure that your system is aware of the latest software versions and dependencies available in the repositories. This is a crucial step before installing new software or upgrading existing packages, as it provides accurate information about what is currently available.

After running **sudo apt-get update**, you might want to follow it up with **sudo apt-get upgrade** to actually upgrade the installed packages to their latest versions. This sequence is commonly used to keep the system up-to-date with the latest software versions and security patches

UPGRADE THE INSTALLED PACKAGES ON THE SYSTEM

The **sudo apt-get upgrade** command in Ubuntu and other Debian-based Linux distributions is used to upgrade the installed packages on your system to their latest versions. It does not install new packages or remove existing ones; it only upgrades the currently installed packages.

Here's a breakdown of what happens when you run **sudo apt-get upgrade**:

1. Check for Updates:

- The command compares the currently installed versions of packages on your system with the latest versions available in the repositories.
- It identifies packages for which newer versions are available.

2. Download and Install Updates:

- For each package that has a newer version available, **sudo apt-get upgrade** downloads the updated package from the repository.
- It installs the new version of the package, replacing the older version on your system.

3. Dependency Handling:

- The package manager (apt in this case) takes care of handling dependencies. If the updated package requires newer versions of other packages, it ensures that those dependencies are also updated.

4. Configuration Changes:

- During the upgrade process, if there are configuration file changes for a package, the package manager may prompt you to decide how to handle those changes. You can choose to keep your current configuration, adopt the new one, or review the differences.

By running **sudo apt-get upgrade**, you ensure that your installed packages are kept up to date with the latest versions available in the repositories. Regularly performing upgrades is essential for security, bug fixes, and getting new features introduced in the software.

It's common to run **sudo apt-get update** before **sudo apt-get upgrade** to ensure that the local package database is up to date with the latest information from the repositories.

IDENTIFY DISTRO VERSION

To identify the distribution and version of Linux installed on your system, you can use various commands. Here are a few commonly used ones:

- **lsb_release -a**
- **cat /etc/os-release**
- **hostnamectl**
- **uname -a**
- **ls /etc/*-release**

Any of these commands should provide information about the Linux distribution and version installed on your system. Choose the one that you find most convenient or that best suits your needs.

UPGRADE THE DISTRO

The **sudo do-release-upgrade** command is used in Ubuntu and other Debian-based Linux distributions to upgrade the operating system to a new release. This command is specifically designed for upgrading from one Ubuntu release to another, and it provides a safe and controlled way to perform the upgrade.

Here's an explanation of what happens when you run **sudo do-release-upgrade**:

1. Check for a New Release:

- The command checks if a new release of Ubuntu is available. This could be a regular release or a Long Term Support (LTS) release, depending on your current version and preferences.

2. Prompt for Confirmation:

- If a new release is found, the command prompts you to confirm whether you want to proceed with the upgrade. It provides information about the new release and any important notes or considerations.

3. Backup Configuration Files:

- Before making changes, the **do-release-upgrade** command prompts you to backup important configuration files. It can automatically back up some files and may ask you about others.

4. Upgrade Process:

- If you confirm, the command begins the upgrade process. It downloads the necessary packages, install them, and upgrades your system to the new release.
- The process involves stopping and starting various services, updating system files, and handling configuration changes.

5. Remove Obsolete Packages:

- The command also removes packages that are no longer needed or are obsolete in the new release.

6. Cleanup:

- After the upgrade is complete, the system may prompt you to remove obsolete packages and perform other cleanup tasks.

7. Reboot:

- Finally, you are advised to reboot your system to apply all changes and start using the upgraded release.

It's important to note that major upgrades should be performed with caution, and it's recommended to have a backup of your important data before starting the process. The **do-release-upgrade** command provides a controlled way to upgrade, but unforeseen issues can still occur, especially if the system has been highly customised.

Always refer to the official Ubuntu documentation and release notes for the specific release you are upgrading to, as there may be additional considerations or instructions.