Getting Started with Shell Commands

The graphical tools in Ubuntu are very easy to use, but if you want to perform advanced tasks in Linux, you may need to learn the Ubuntu command-line interface (CLI). The command-line tools are faster and used often in debugging the system. The command-line interface in Linux can be compared to the disk operating system (DOS) in Windows.

We mainly use the command line when we work with ROS. Knowledge of the Linux terminal commands is a prerequisite for working with ROS.

The Ubuntu command-line interface is in a tool called Terminal. Use the Ubuntu Dash search to find the Terminal application. Figure 1-31 shows an example.

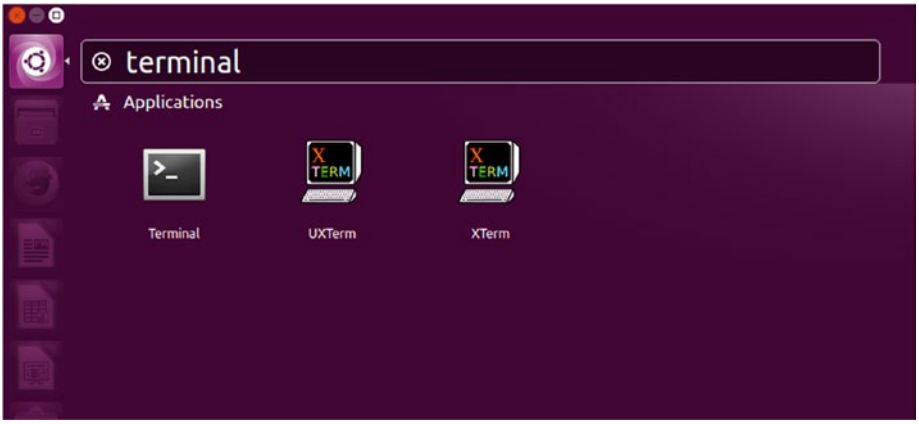

Figure 1-31. *Searching for the Terminal application*

Click Terminal to open the application, which is shown in Figure 1-32.

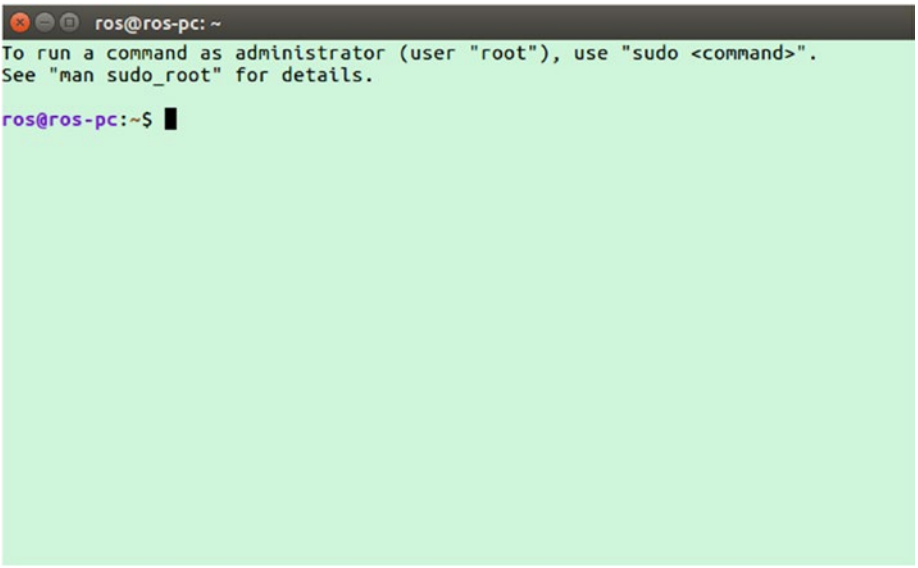

Figure 1-32. *The Ubuntu terminal*

Terminal Commands Cheat Sheet

This section covers useful shell commands for working with robots and ROS. The following are the popular commands that you want to explore.

man: Manual Pages for Shell Commands

The `man` command stands for *manual*. This command provides the manual page of a given command.

Usage: `man <shell command>`

Example: `man ls`

The preceding asks for the manual page of `ls`. Figure 1-33 shows the output of `man ls`.

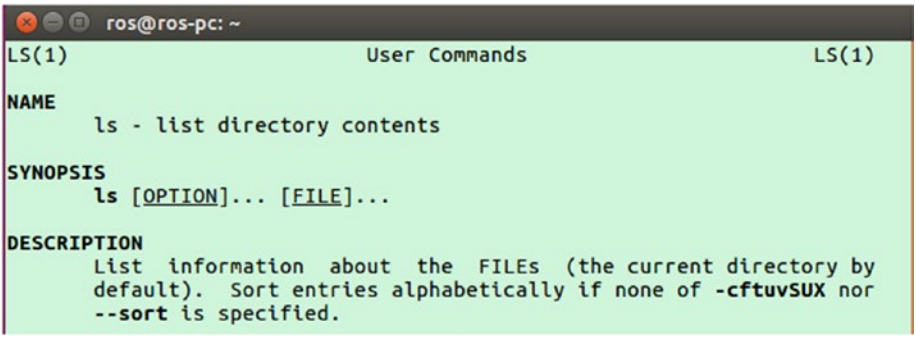

Figure 1-33. *The manual page of ls*

ls: List Directory Content

The `ls` command lists the content of files and folders in the current directory.

Usage: `ls`

The output of `ls` is shown in Figure 1-34.

Figure 1-34. *List of files in the current path*

cd: Change Directory

The cd command switches from one folder to another (see Figure 1-35).

Usage: `cd <Directory_path>`

Example: `cd Desktop`

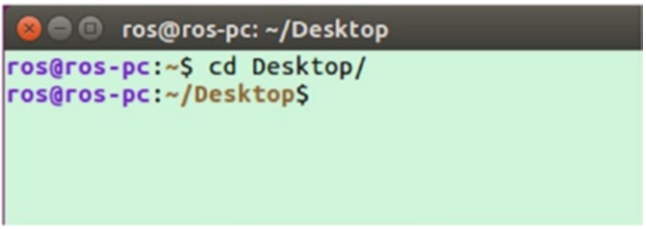

Figure 1-35. *Changing folders*

pwd: Current Terminal Path

The pwd command returns the current path of the terminal. This is useful for getting the absolute path.

Usage: `pwd`

Figure 1-36 shows the output of the pwd command.

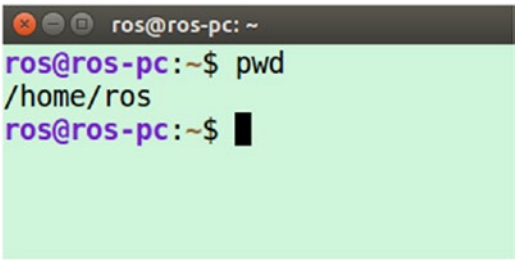

Figure 1-36. *Command to get current path*

mkdir: Create a Folder

The `mkdir` command creates an empty folder or directory.

Usage: `mkdir <folder_name>`

Example: `mkdir robot`

Figure **1-37** shows how to create and list folders.

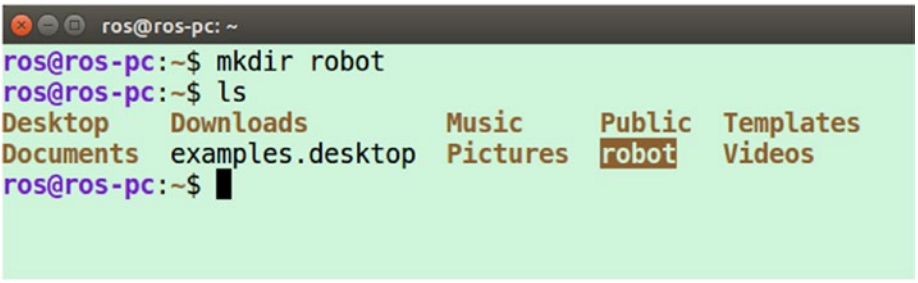

Figure 1-37. *Creating a new folder*

rm: Delete a File

The `rm` command deletes a file.

Usage: `rm <file_path>`

Example: `rm test.txt`

An example is shown Figure 1-38. The files are listed before deletion and after deletion to confirm that the files were actually deleted.

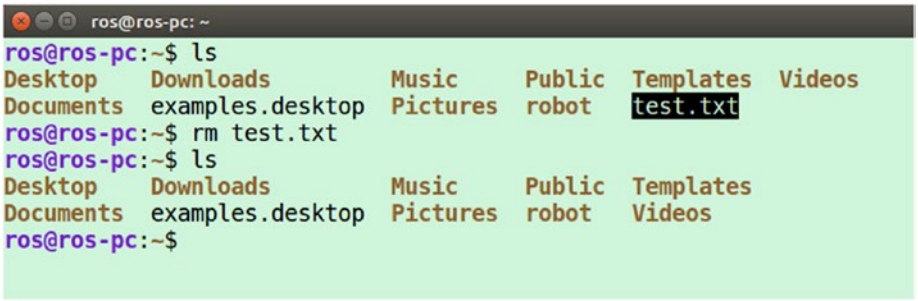

Figure 1-38. *Deleting a file*

To delete a folder by recursively deleting its files, use the following command.

```
$ rm -r <folder_name>
```

To delete a file inside the root (`/`) file system, use `sudo` before the `rm` command.

```
$ sudo rm <file_name>
```

rmdir: Delete a Folder

The `rmdir` command deletes an empty folder. You may need to delete files before using this command.

Usage: `rmdir <folder_name>`

Example: `rmdir robot`

Figure 1-39 shows an example of this command.

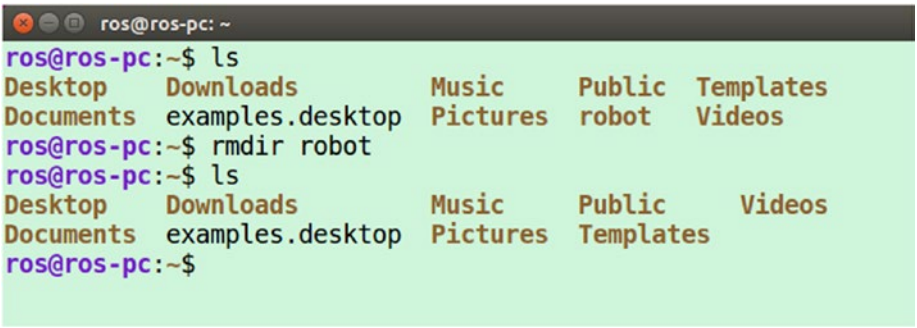

Figure 1-39. *Deleting an empty folder*

mv: Move a File from One Place to Another

The `mv` command moves a file from one location to another and then renames the file.

Usage: `mv source_file destination/destination_file`

Example: `mv test.txt test_2.txt`

In Figure **1-40**, `test.txt` is moved into the same folder under a different name (i.e., `test_2.txt`).

It is moving the file by renaming the file.

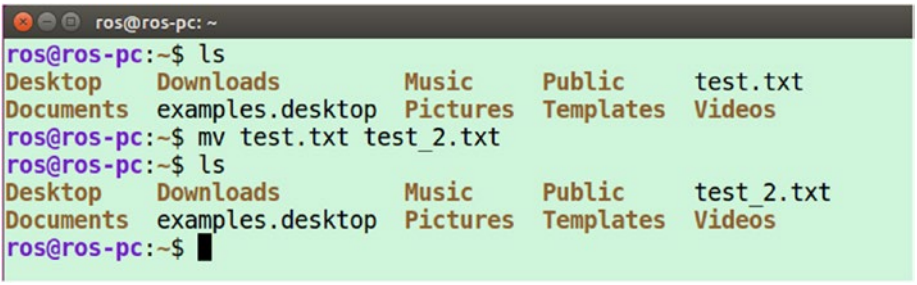

Figure 1-40. *Moving a file*

cp: Copy a File from One Path to Another

The cp command copies files from one location to another.

Usage: `cp source_file destination_folder/destination_file`

Example: `cp test.txt test_2.txt`

Figure 1-41 demonstrates this example.

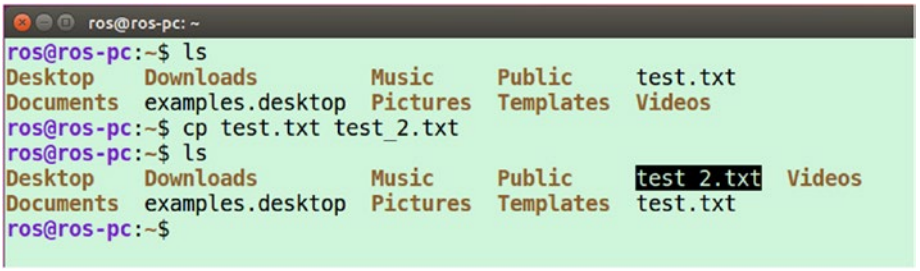

Figure 1-41. *Copying a file*

dmesg: Display a Kernel Message

The dmesg command is very useful for debugging the system. It displays the kernel logs (see Figure 1-42). From these logs, you can debug the problem.

Usage: `dmesg`

```
ros@ros-pc: ~
ros@ros-pc:~$ dmesg
[    0.000000] Linux version 4.10.0-28-generic (buildd@lgw01-12) (gcc version 5.
4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.4) ) #32~16.04.2-Ubuntu SMP Thu Jul 20
 10:19:48 UTC 2017 (Ubuntu 4.10.0-28.32~16.04.2-generic 4.10.17)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.10.0-28-generic root=UUI
D=fa38abcb-2976-4252-ac26-db79f83f21be ro quiet splash
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Centaur CentaurHauls
[    0.000000] ------------[ cut here ]------------
[    0.000000] WARNING: CPU: 0 PID: 0 at /build/linux-hwe-vH8Hlo/linux-hwe-4.10.
0/arch/x86/kernel/fpu/xstate.c:595 fpu__init_system_xstate+0x36d/0x99e
[    0.000000] XSAVE consistency problem, dumping leaves
[    0.000000] Modules linked in:
[    0.000000] CPU: 0 PID: 0 Comm: swapper Not tainted 4.10.0-28-generic #32~16.
04.2-Ubuntu
```

Figure 1-42. *Checking the kernel logs*

lspci: List of PCI Devices in the System

The `lspci` command also debugs the PC. This command lists the PCI devices in the PC (see Figure 1-43).

Usage: `lspci`

```
ros@ros-pc: ~
ros@ros-pc:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphi
cs Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controll
er (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Contr
oller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Cont
roller [AHCI mode] (rev 02)
ros@ros-pc:~$
```

Figure 1-43. *Listing the PCI devices*

lsusb: List of USB Devices in the System

The `lsusb` command lists all USB devices (see Figure 1-44).

Usage: `lsusb`

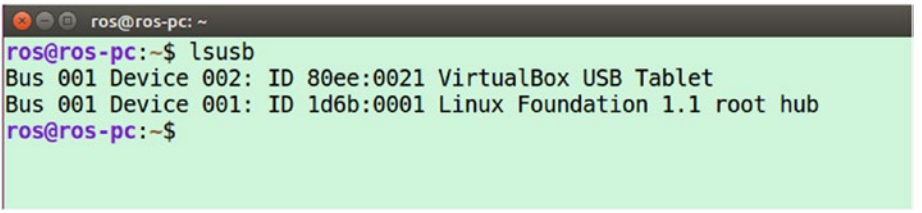

Figure 1-44. *Listing the USB devices*

sudo: Run a Command in Administrative Mode

The `sudo` command is one of the most important. We use it regularly. It runs a command with administrative privileges (see Figure 1-45). We can also completely switch to root (administrator) mode using this command.

Usage: `sudo <parameter> <command>`

Example: `sudo -i`

This example command switches to root mode.

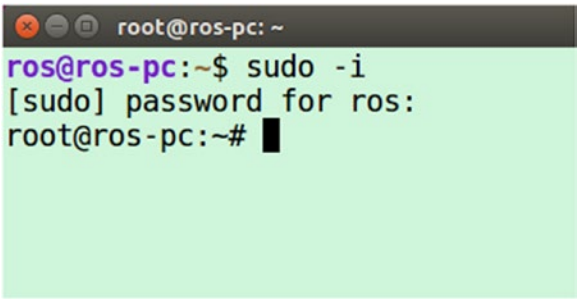

Figure 1-45. *Switching to adminstrator mode*

Figure 1-46 shows the results of executing a command in root mode.

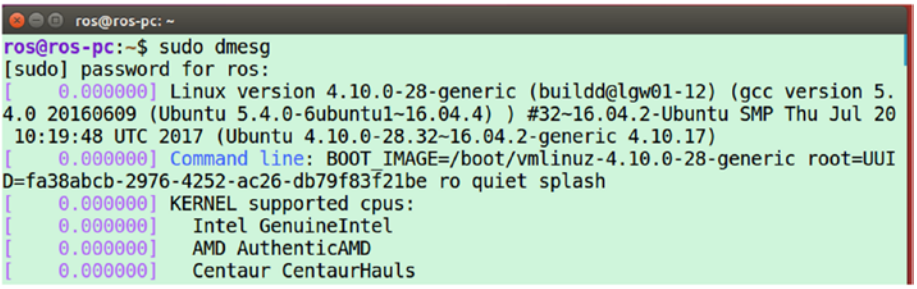

Figure 1-46. *Running a command with administrative privilege*

ps: List the Running Process

The `ps` command lists the running process in your system.

Usage: `ps <command arguments>`

Example: `ps -A`

When we execute the `ps` command, it lists the process in the current terminal. If we run `ps -A`, it lists all the processes running in the system. Both results are shown in Figure 1-47. PID is the process ID, which identifies the running process. TTY is the terminal type.

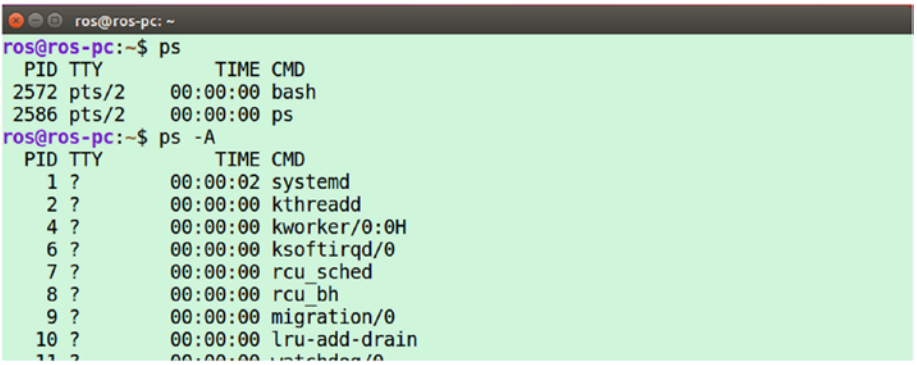

Figure 1-47. *Listing the processes running on the system*

kill: Kill a Process

To end a process running in the system, use the `kill` command.

Usage: `kill <PID>`

Usage: `kill 2573`

To kill a process, we have to identify the PID of process and provide it with the command. The results of the command are shown in Figure 1-48.

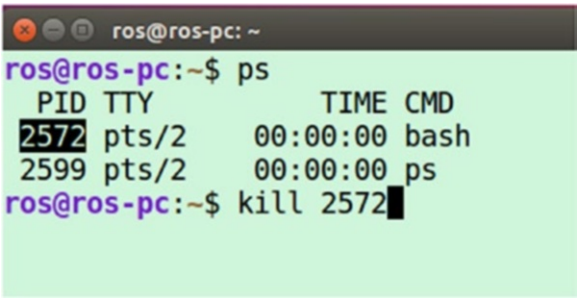

Figure 1-48. *Killing a process*

apt-get: Install a Package in Ubuntu

The `apt-get` command is important and very useful when working with Ubuntu and ROS. It installs an Ubuntu package that is either in the Ubuntu repositories or on the local system. The packages are called Debian packages, which have .deb extensions. Installing a package requires root permission, so we have to use `sudo` before the command. We can also update the list of packages in the repositories using this command.

Usage: `$ sudo apt-get <command_argument> <package_name>`

Example: `$ sudo apt-get update`

Example: `$ sudo apt-get install htop`

Example: `$ sudo apt-get remove htop`

Figure 1-49 shows the Ubuntu package update using `sudo apt-get update`. This command updates the package download location in the local system.

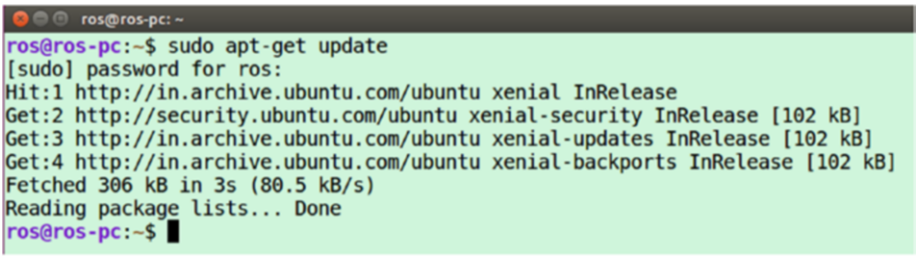

Figure 1-49. *Updating the Ubuntu software repository*

Figure 1-50 shows how to install a package. We are installing a tool called `htop`. It is a terminal process viewer.

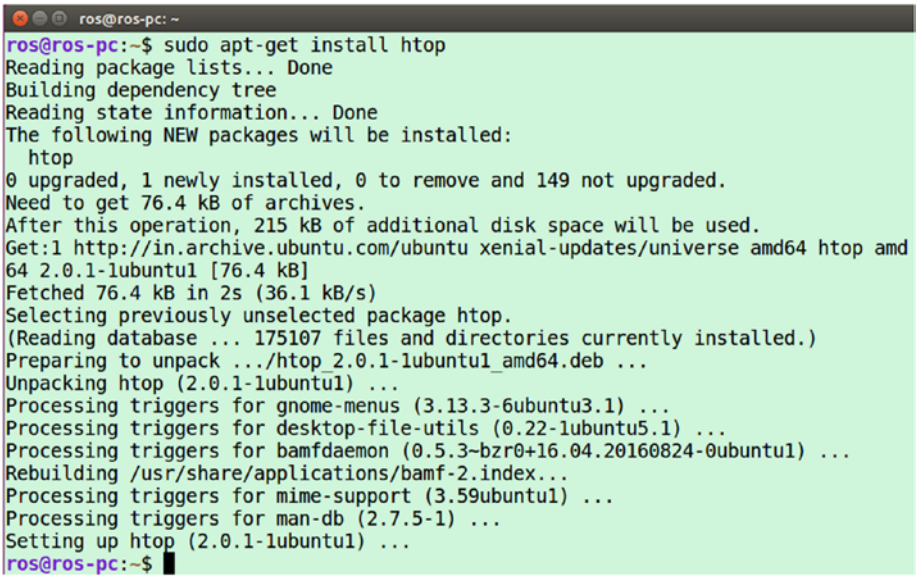

Figure 1-50. *Installing a package on Ubuntu*

The `sudo apt-get remove htop` command in Figure 1-51 shows how to remove a package. We have to use the `remove` argument to delete it.

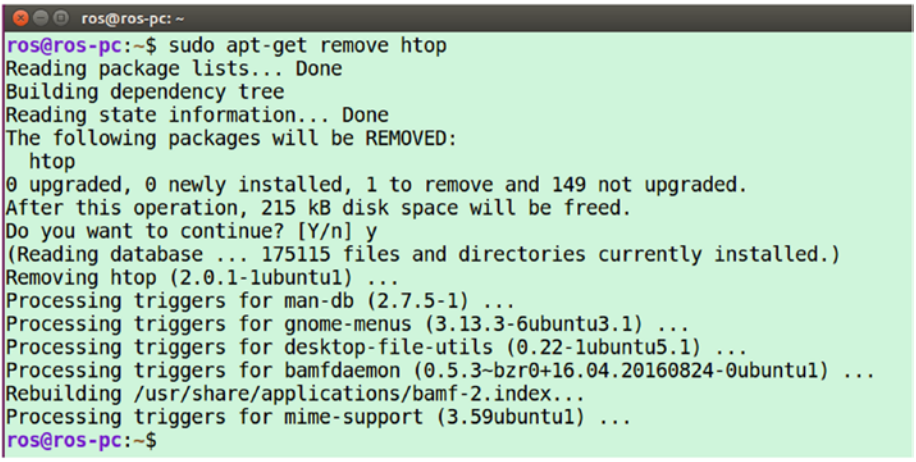

Figure 1-51. *Removing a package from Ubuntu*

Figure 1-52 shows how to install a local Debian package using the `apt-get` command. The local file is on the same path of the terminal, and the name of the Debian file is `htop.deb`, so we can use the following:

```
$ sudo apt-get install ./htop.deb
```

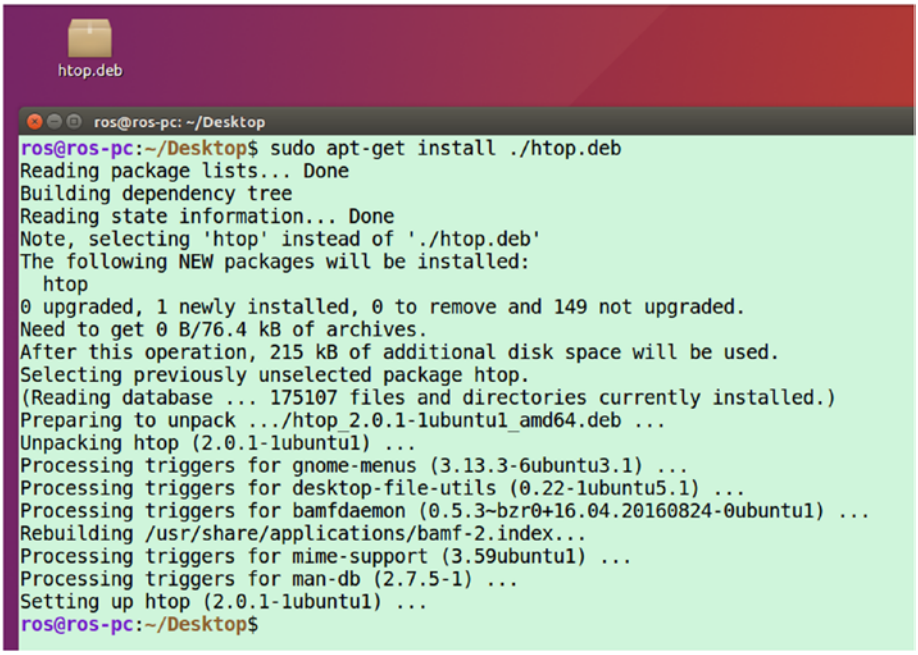

Figure 1-52. *Installing a Debian package in Ubuntu*

dpkg -i: Install a Package in Ubuntu

The dpkg command is another way to install a Debian package.

Usage: `dpkg <command_arguments> debian file name`

Example: `dpkg -i htop.deb`

Figure 1-53 shows the results of the dpkg command.

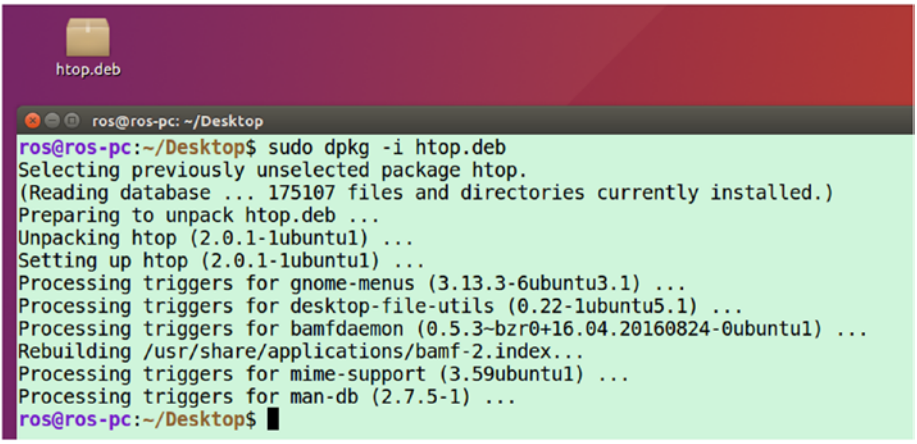

Figure 1-53. *Installing a Debian package in Ubuntu*

reboot: Reboot the System

We can restart the system using the terminal command (see Figure 1-54).

Usage: `sudo reboot`

This instantly reboots the system.

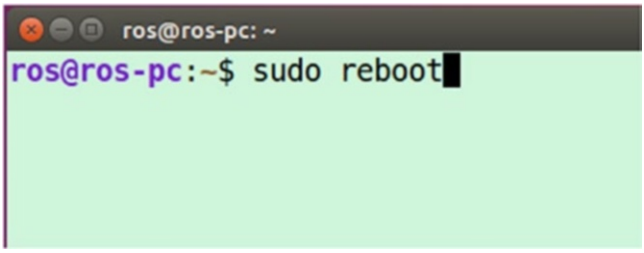

Figure 1-54. *Rebooting PC*

poweroff: Switch off the System

If you want to instantly shut down the system, use the `poweroff` command (see Figure 1-55).

Usage: `$ sudo poweroff`

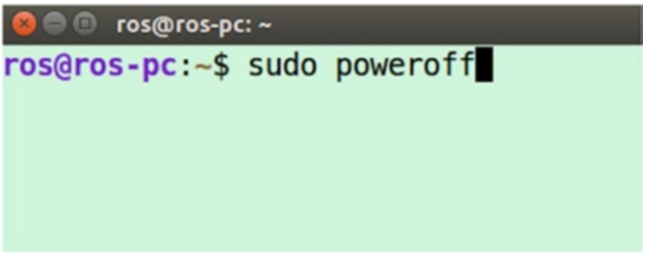

Figure 1-55. *Shutting down the PC*

htop: Terminal Process View

The `htop` is a process viewer in Linux (see Figure 1-56). It is not installed in the system by default. You have to install it using `apt-get`. This command is very useful for managing process.

Usage: `htop`

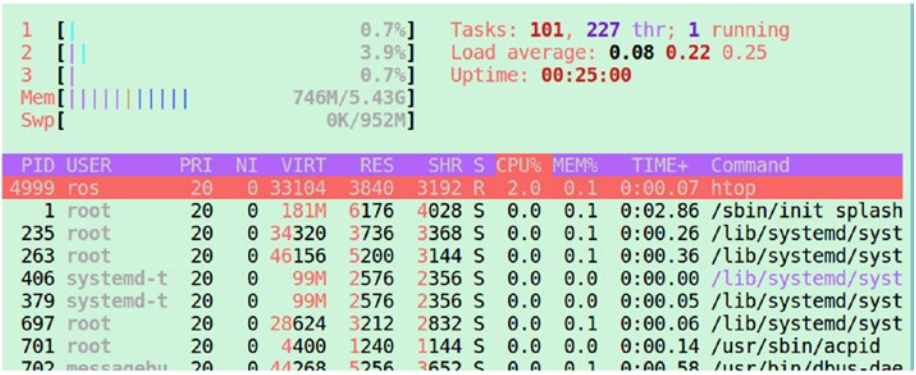

Figure 1-56. *Terminal process viewer*

nano: Text Editor in Terminal

There is a useful text editor that you can use while working in the terminal. You can create code inside the terminal (see Figure 1-57).

Usage: `$ nano file_name`

Example: `$ nano test.txt`

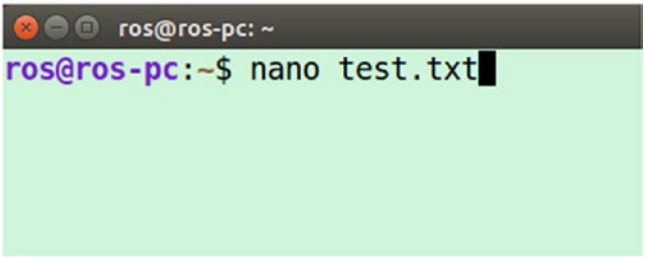

Figure 1-57. *Text editor in the terminal*

Figure 1-58 shows the resulting screen. In this editor, you can enter your code.

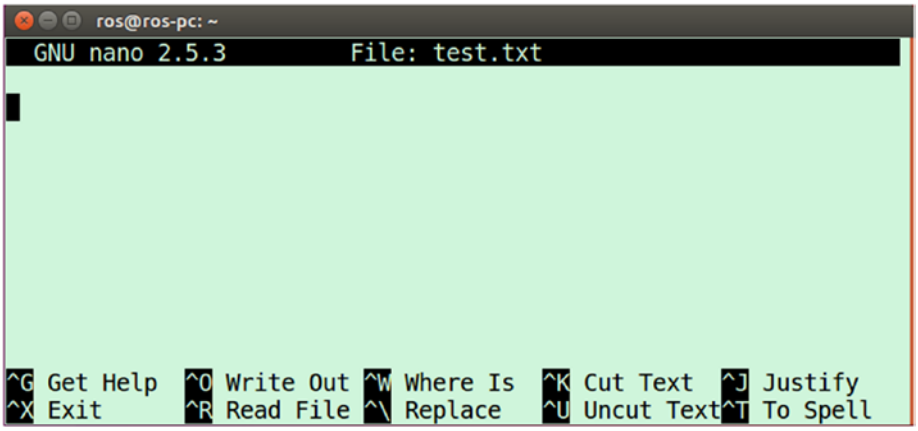

Figure 1-58. *Nano text editor in terminal*

After completing the code, press Ctrl+O to save the file. You are asked to enter the file name. You can enter a new file name or use an existing name. Press Enter to save (see Figure 1-59).

Press Ctrl+X to exit from the editor. To open the file again, use `nano file_name`.

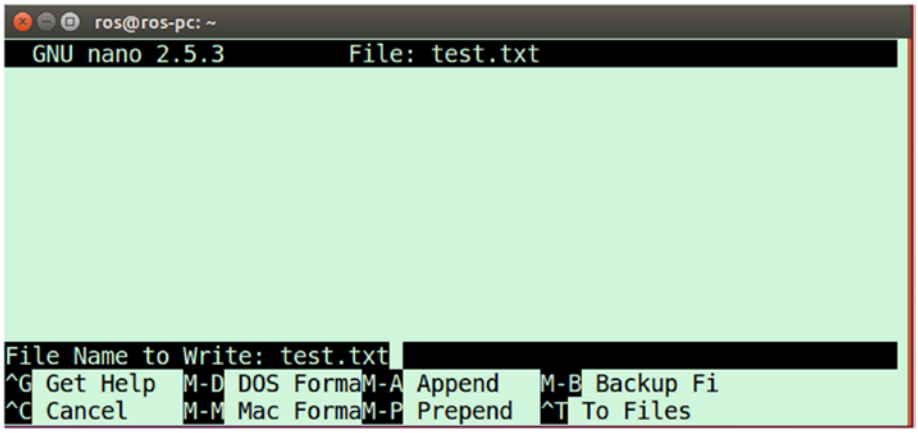

Figure 1-59. *Saving a file in the nano text editor in the terminal*

Summary

This chapter discussed the fundamentals of the Ubuntu operating system, its installation, and the important shell commands that we need for working with robots. This chapter is important because, before working with ROS-based applications, you should have a basic understanding of Linux and its commands. Understanding the Linux environment and its commands is one of the prerequisites for learning ROS. This book discusses all the prerequisites needed for learning ROS. This chapter is the first step in learning ROS.