



KodeGo

File Navigation

Command Line Interface



Overview

1. Linux File System
2. Navigation Commands
 - Print Directory
 - Change Directory

Understanding the basics

Before we get into commands, let's talk about important special characters. The dot (`.`), dot-dot (`..`), forward slash (`/`), and tilde (`~`), all have special functionality in the Linux filesystem:

Meaning:

- ☐ The dot (`.`) represents the current directory in the filesystem.
- ☐ The dot-dot (`..`) represents one level above the current directory.
- ☐ The forward slash (`/`) represents the "root" of the filesystem. (Every directory/file in the Linux filesystem is nested under the root / directory.)
- ☐ The tilde (`~`) represents the home directory of the currently logged in user.

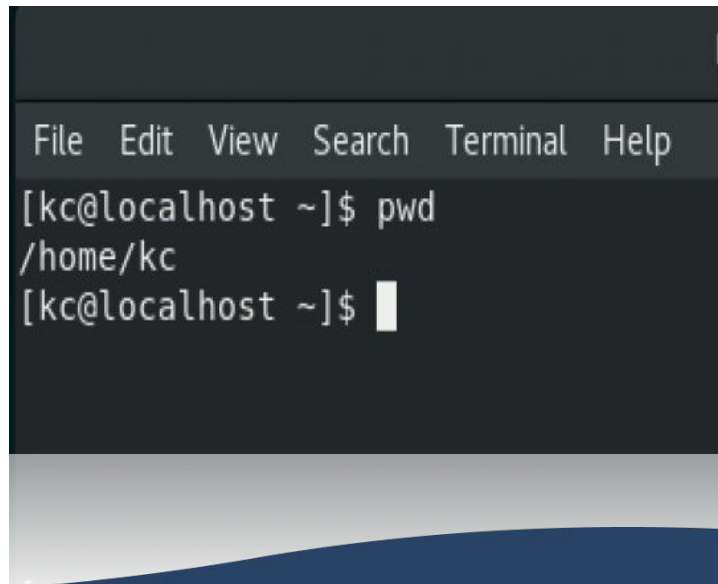
Let's put things in to practice

Print working directory (pwd)

The `pwd` command prints the current/working directory, telling where you are currently located in the filesystem. This command comes to your rescue when you get lost in the filesystem, and always prints out the absolute path.

What is an absolute path? An absolute path is the full path to a file or directory. It is relative to the root directory (`/`). Note that it is a best practice to use absolute paths when you use file paths inside of scripts. For example, the absolute path to the `ls` command is: `/usr/bin/ls`.

If it's not absolute, then it's a relative path. The relative path is relative to your present working directory. If you are in your home directory, for example, the `ls` command's relative path is: `../../usr/bin/ls`.

A screenshot of a terminal window with a dark background. At the top, there is a menu bar with the options 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. Below the menu bar, the terminal shows a prompt '[kc@localhost ~]\$' followed by the command 'pwd'. The output of the command is '/home/kc'. Below the output, the prompt '[kc@localhost ~]\$' is shown again with a white cursor character (a vertical bar) at the end, indicating the terminal is ready for the next command.

```
File Edit View Search Terminal Help
[kc@localhost ~]$ pwd
/home/kc
[kc@localhost ~]$
```

Let's put things in to practice

Change directory (cd)

The `cd` command lets you change to a different directory. When you log into a Linux machine or fire up a terminal emulator, by default your working directory is your home directory. My home directory is `/home/kc`. In your case, it is probably `/home/<your username>`.

Absolute and relative paths make more sense when we look at examples for the `cd` command. If you need to move one level up from your working directory, in this case `/home`, we can do this couple of ways. One way is to issue a `cd` command relative to your `pwd`:

Let's put things in to practice

```
$ cd ..
```

Note: Remember, .. represents the directory one level above the working directory. The other way is to provide the absolute path to the directory:

```
$ cd /home
```

Either way, we are now inside /home. You can verify this by issuing pwd command. Then, you can move to the filesystem's root (/) by issuing this command:

```
$ cd /
```

Let's put things in to practice

If your working directory is deeply nested inside the filesystem and you need to return to your home directory, this is where the ~ comes in with the cd command. Let's put this into action and see how cool this command can be. It helps you save a ton of time while navigating in the filesystem.

My present working directory is root (/). If you're following along, yours should be same, or you can find out yours by issuing pwd. Let's cd to another directory:

```
$ cd /home/kc/0/1/2/3/4/5/6/7/8/9
```

```
$ pwd
```

```
/home/kc/0/1/2/3/4/5/6/7/8/9
```

Let's put things in to practice

To navigate back to your home directory, simply issue ~ with the cd command:

```
$ cd ~
```

Again, check your present working directory with the pwd command:

```
$ pwd
```

```
/home/kc
```

The dash (-) navigates back to the previous working directory, similar to how you can navigate to your user home directory with ~. If you need to go back to our deeply nested directory 9 under your user home directory (this was my previous working directory), you would issue this command:

```
$ cd -
```


References:

1. Navigating Linux File System:
<https://www.redhat.com/sysadmin/navigating-linux-filesystem>
2. Images: <https://kodego.ph/>

