From Zero to Terminal Hero

32nd TaCoS conference

July 1, 2023

Heinrich Heine University
Düsseldorf, Germany

Akhilesh Kakolu Ramarao

Contact

- Email: kakolura@hhu.de
- Website: `https://akkikek.xyz`

1. **Command line**

2. Vim

Intro

- The Linux command line is a text interface to your computer.
- Often referred to as the shell, terminal, console, prompt or various other names, it can give the appearance of being complex and confusing to use.

History

- Adopted from this guide.
- During the formative years of the computer industry, one of the early operating systems was called Unix.
- It was designed to run as a multi-user system on mainframe computers, with users connecting to it remotely via individual terminals.
- These terminals were pretty basic by modern standards: just a keyboard and screen, with no power to run programs locally. Instead they would just send keystrokes to the server and display any data they received on the screen.
- There was no mouse, no fancy graphics, not even any choice of colour.

Why use terminals?

- Compared with graphics, text is very light on resources.
- The commands are kept very terse to reduce the number of keystrokes needed, speeding up people's use of the terminal even more.
- This speed and efficiency is one reason why this text interface is still widely used today.

WSL

- Windows Subsystem for Linux (WSL) is a compatibility layer for running Linux binary executables natively on Windows 10 and Windows 11.
- Hit the windows key and open your WSL by searching for **Ubuntu**

Mac Users

- Hit **cmd** + **space** and search for **terminal** and press *Enter*.

Launching the terminal

- When you launch your terminal, you should end up with a rather dull looking window with an odd bit of text at the top.
- Let's run our first command. Click the mouse into the window to make sure that's where your keystrokes will go, then type the following command, all in lower case, before pressing the *Enter* or *Return* key to run it.
- type **pwd** to see a directory path printed out: probably something like /home/YOUR USERNAME
- **pwd** is an abbreviation of **print working directory**

Before we continue...

- There is one thing you need to understand here, before we get into the detail of what the command actually did. First is that when you type a command it appears on the same line as the odd text.
- That text is there to tell you the computer is ready to accept a command, it's the computer's way of prompting you.
- In fact it's usually referred to as the prompt, and you might sometimes see instructions that say 'bring up a prompt', 'open a command prompt', 'at the bash prompt' or similar. They're all just different ways of asking you to open a terminal to get to a shell.

pwd

- One important concept to understand is that the shell has a notion of a default location in which any file operations will take place.
- This is its working directory. If you try to create new files or directories, view existing files, or even delete them, the shell will assume you're looking for them in the current working directory unless you take steps to specify otherwise.
- So it's quite important to keep an idea of what directory the shell is "in" at any given time, after all, deleting files from the wrong directory could be disastrous. If you're ever in any doubt, the pwd command will tell you exactly what the current working directory is.

Creating folders

- Create a directory called iot using **mkdir iot**
- **mkdir** is short for **make directory**
- You can change the working directory using the **cd** command, an abbreviation for **change directory**.
- Now, go to iot directory using: **cd iot**

Creating files

- Create a text file called tutorial1 using **touch tutorial1.txt**
- Create another text file called tutorial2 using **touch tutorial2.txt**
- Now, let's look at all the files in the directory iot using **ls**
- ls is short for **list**

Additional resources

- Video tutorial on Linux Terminal
- Linux commands cheatsheet is uploaded in ILIAS under Reading Materials > Linux
- The Linux command line for beginners
- An Introduction to the Linux Terminal
- Introduction to WSL2

1. Command line

2. Vim

Intro

- Vim is a simple and powerful computer program used for writing.
- The first version of Vim was released in 1991.
- It is still one of the most popular text editor!

Writing a file

Open Vim using the ‘vim‘ command. If you get an error saying ‘command not found: vim‘, you will have to do the following –

WSL and Linux users: `sudo apt install vim`

- Type `:e hello_world.txt` and press `Enter` key
- Press **i**
- Type `Hello World!`
- Press the `<Esc>` key
- Type `:w` and press the ‘Enter‘ key
- Close Vim by typing `:q`

Modes

There are 3 basic types of modes in Vim - **Normal**, **Insert** and **Visual**

Normal Mode Normal mode is where you can run the commands. This is also the default mode when you open Vim.

Insert Mode Insert mode is where you write the text.

Visual Mode Visual Mode is where you select a bunch of text so that you can run a command only on a part of text.

Working with a python file

- Download the spellchecker code by Peter Norvig using:

  ```
  wget https://norvig.com/spell.py
  ```

- Open the file with Vim using:

  ```
  vim spell.py
  ```

Editing...

- → Move to the next word: `w`
- → Go to line 81: `:81`
- → Delete the import: `dd`
- → Go to line 10: `:10`
- → Paste the last deleted line here: `p`
- → Delete the 4 spaces to de-indent: `d4h`

h,j,k,l

- Place your index finger (of your right hand) to press `h`. This is how you go left.
- `j` to go down.
- `k` to go up.
- `l` to go right.

h,j,k,l

- Move 6 characters to the right by pressing `6l`.
- Press `3h` to move left by 3 characters.
- Press `2j` to go to the 2nd line.
- Press `2k` to go back to where you were.

Commonly used keystrokes

- Move the cursor to the last line of the file by pressing `G`
- Move the cursor to the first line of the file by pressing `1G`
- Go to the last character of the line by pressing `$`
- Go to the first character of the line by pressing`^`

Copy and pasting

- First select the block of text that you want to copy using: `v`
- The operator for copy is `y` which stands for 'yank'.
- Paste it at your desired location using `p` which stands for 'paste'.

Best ways to get better at vim

- `vimtutor` which comes as a default program for unix-based operating systems. Just type `vimtutor` in your command line.
- Improve your muscle memory using `https://www.openvim.com/tutorial.html` and `https://vim-adventures.com/`.
- Most importantly, use Vim as your daily driver!

Other forks of Vim

- https://github.com/neovim/neovim
- https://spacevim.org/

  ... and many more

Thank you!