

Unix Workbench Notes

Haggai Liu

April 24, 2018

1 Unix and Command Line Basics

1.1 What Is Unix?

Unix is an operating system and a set of tools. one particular tool is known a a **shell**, which is a computer program that provides a command line interface. Using the command line interface lets you enter lines of code into a shell (also called a console) and that code instructs your computer to perform a specific task. The terms **command line**, **shell**, and **console**, are often used interchangeably.

The shell is a very direct and powerful way to manipulate a computer. We can produce wonderful creations that help thousands of people, or we can wreak havoc on ourselves and on others.

There are several popular shell programs but Bash is the default shell program on Mac and Ubuntu.

1.2 Getting Unix

Mac and Ubuntu users can simply open up the Terminal. Windows 10 users can enable and install Bash on Ubuntu on Windows. Users of earlier Windows versions can use VirtualBox.

1.3 Command Line Basics

1.3.1 Hello Terminal!

Once we fire up a Terminal there will be a prompt like the following:
`<machine-id>:~<user>$`

Commands are typed right after the prompt. Enter is pressed after the command to execute it. If no command is typed before pressing enter, nothing happens and a new prompt appears directly below the old prompt. All those prompts appearing on the screen after several enter presses may begin to look messy. Those prompts can be cleared using the `clear` command.

Every command line command is actually a little computer program, even commands as simple as `clear`. These commands all tend to have the following structure:

`[command] [options] [arguments]`

Some simple commands like `clear` don't require any options or arguments. Options are usually preceded by a hyphen (-) and they tweak the behaviour of the command. Arguments can be names of files, raw data, or other options that the command requires. A simple command that has an argument is `echo`. The `echo` command prints a phrase to the console. An example is given below. Surround the message with single or double quotes.

```
:~$ echo "Hello World"
Hello World
:~$
```

To see the last command press the Up arrow key. We can press Up and Down in order to scroll through the history of commands that were entered. If we want to re-execute a past command, we can scroll to that command then press Enter.

Summary:

- Commands are typed after the prompt.
- `clear` cleans up the terminal screen.

- `echo` prints text to the terminal.
- Can scroll command history via Up and Down arrows.

Exercises:

1. Print your name to the console.

```
echo "Haggai Liu"
```

2. Clear your terminal after that.

```
clear
```

2 Working with Unix

3 Bash Programming

4 Git and GitHub