Introduction to the Command Line Table of Content_version_2

GETTING STARTED

Introduction

A Brief Overview

```
$ bash
% zsh
# as root using bash
```

INTRODUCTION TO THE COMMAND LINE

The Command Line Interface

What is an interface?

```
- 2 parts:
    a display of information about what the computer is doing.
    a method to tell the computer what to do.

- a program called terminal is used to provide a CLI on OS that use GUI

- PS1 = prompt = [user]@[hostname]:[current_directory]$

- 4 components of the CLi = prompt + cursor + input + output

- Anatomy of a command = [command] [argument]
```

What is the Command Line used for?

```
- system admin tasks
```

What are some common commands?

```
- cd
- ls
- pwd
- touch
- mkdir
- rm
- cp
- mv
- echo
- cat
- more
- less
- head
- tail
- man
```

Files, Directories, and Executables

Linux/Unix File System legend

```
/ root
. current directory or ./
.. one level up
../.. two leves up
~ home
* splat represents any characters
```

Executables

```
Executables:
- have special characters at the beginning to tell the computer how to execute them
- have scripts or machine language as their content
- have executables permission
```

The Environment

Environment Variables

```
- provide context to commands as well
- use env command to list environment variables
```

How to Change your Command Line Environment

```
Special Characaters:
\h Hostname
\u User name
\w Current Directory
\W Basename of current directory
\d Current date

source to run/refresh .bashrc
```

$PATH and Executables

```
- /bin is short for binary
- /bin is standard directory name for executable files

- The PATH variable determine which directories are searched when a command is entered
- PATH is an ordered, colon delimited, list of directories that contain executables
- The order of the directories is the PATH variable is first-found-first-execute
- If you use /, ., or ~ before your command, the command line will interpret that as an
  actual path to a file, and not use the PATH
- You can add to PATH to make more commands available wihtout having to memorize their
  exact path
- Modifications to PATH, on the fly will not be permanent; permanent modifications should
  be done in an environment file, like .bashrc
```

Exercice 2

```
# CREATE AN EXECUTABLE FILE
cd ~ # location of the executable
echo '#!/bin/bash' > test.sh
echo 'echo "Hello world"' >> test.sh
chmod +x test.sh # make sure test.sh has the executable permission
```

```
#RUN THE EXECUTABLE
./test.sh # specifying the path to the executable
test.sh => -bash: test.sh: command not found # THE PATH TO THE EXECUTABLE HAS TO BE ADDED
TO THE PATH ENV VARIABLE IN ORDER TO BE RUN FROM ANYWHERE

PATH = $PATH:$HOME # adding the path to the executable which is home to the env var PATH
echo $PATH # checking
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/aws/bin:/home/ubuntu/bin

$ test.sh # now can be run from anywhere
Hello world
$ cd / # now can be run from anywhere
$ test.sh
Hello world
```

Permissions

Summary

```
d|user|group|other|username|groupname

0 no permission granted
1 execute
2 write
3 execute + write
4 read
5 execute + read
6 write + read
7 read + write + execute

- Permissions are assigned to files and directories -- not users and groups
- Access levels are determined by the rwx
- A file must have the x permission to execute that file directly
- To change permissions of a file or directory, you either must be logged in as that
  file's owner or the root user, or you must use the sudo command.
```

Other Interfaces

Changing context in terminal

```
top (command for system monitoring)
vim
mysql
irb
... provide new ways to interface with computer
these types of commands change the context of the Command Line
```

CONCLUSION

Conclusion

Overview of Concepts

```
- The command line is an interface to your computer's files and directories
- Evertything you do in the command line is related to files, directories, and
executables.
- Environment variables provide context for what you do in the command line
- Access to files, directories, and executables is determined by their read, write,
and execute permissions.
There are permissions for the files's use(owner), group, and other(everyone else).
- Some programs provide a completely new context within the command line interface.
```