Independent Study

Xterm/Bash

What is Bash?

Bash is a Unix shell and command language

It is a lower-level way to interact with your computer, making it really powerful

In Windows, it is usually called **Command Prompt**

Bash Terminologies

A command is usually called a **prompt**, and the bash interface will show **$** where you can start writing your prompt.

A folder is usually called a **directory** in Bash. Meanwhile, your current position (how deep you are as you go through your folders) is usually called a **path**.

You can either use built-in prompts to interact with your computer, or create your own prompt.

(For this project, we'll be exploring the basic built-in prompts in Bash)

Basic Bash Prompts

$ pwd - Stands for ‘print_wdir’ or print working directory. It prints the current path that you’re at right now.

$ ls - Stands for ‘list’. It lists all the files and directories in your current working directory.

$ cd *<directory>* - Stands for ‘change dir(ectory)’. It allows you to change your path by going into the directory that you have specified,

$ echo *<string>* - As it says on the tin, it echoes/repeats the string that you have provided

$ cat *<filename>* - Stands for ‘concatenate’. It has many purposes, but a very simple and handy usage of it is to view the contents of the given file.

XTerm.js

XTerm.js is a JavaScript library that we will be using to emulate a Terminal in our project.

Source: https://xtermjs.org/

Although it emulates a Terminal, it is not a fully working terminal. Much of the backend functionality has to be provided by the user.

It also require a **WebSocket**, a two-way interactive communication session that allows the client to interact with the server using TCP connection protocol.

For the project, the WebSocket is served in **http://localhost:1337**. It takes in a JSON with the user prompt, and returns a JSON with the appropriate data to be displayed depending on the prompt.

XTerm Implementation

Each stroke of the key is captured using ‘term.onKey()’. Each of the key stroke is stored in ‘curline’ variable.

Upon encountering ‘enter’ keystroke (code 127), the ‘curline’ will be bundled into one JSON message and sent to the websocket.

The websocket receives and processes the given message, and return the appropriate data to show to client.

The client receives the return message, and displays the content of the message on the terminal.