Alice Kuang

EECS 581

10/2/22

Sprint 2 Research Deliverables

**Docker notes**

Reference: <https://youtu.be/pTFZFxd4hOI?t=2342>

Included Project: hello-docker

image: is a cut-down os with a runtime environment, libraries, and all other resources

container: process which has own file system provided by image

docker hub: provides storage for docker files, similar to the relation of git and github.

Instructions for setup w/ Node (follow along in hello-docker folder):

Install node

Open VSC, install docker extension

Create Dockerfile, with base image of node:DISTRIBUTION (search for images on hub.docker.com)

Build the docker image in terminal:

docker build -t program-name

To check if program was built:

docker image ls

To run:

docker run program-name

Upload to your docker hub:

docker push alicekuang/program-name

Test your docker:

www.labs.play-with-docker.com

docker pull ... (latest)

**Linux Shortcuts:**

pwd = print working directory

ls -l = show long listing

"tab" is auto-completion

cd .. = get to previous directory

cd ../.. = get to two levels previous directory

cd ~ = go to home directory

touch TXT.txt = create new file

mv sourceFile destFile = move / rename a file

rm fileName = remove file

rm -r directory = recursively remove in directory

echo text > file.txt = redirecting the text string into a file

Most of these commands are programs in the bin directory!

Editing files w/ nano:

1. apt install nano

2. nano fileName.type

3. apt install less --view content of long files interactively

cat fileName --view small files

tail -n # /loci

head -n # /loci

**NodeJS Notes (reference: https://nodejs.dev/en/learn/)**

Q: What is NodeJS?

A: A popular open-source, cross-platform JavaScript runtime environment

* Node.js apps run in a single process, without creating new threads
* It also uses a set of asynchronous I/O primitives in its standard library; the advantage is it doesn’t block code.
* Thread concurrency introduces a lot of bugs, NodeJS avoids this by avoiding thread creation
* A great advantage of Node.js is it makes possible writing server-side code and client-side code in one language (JavaScript)
* JavaScript package manager **npm** hosts over 1,000,000 open source packages that developers can use.
* Nodejs standard library reference: <https://nodejs.org/api/>

<https://github.com/nodejs/examples.git>

What is a server in node.js?

A: Any software built w/ node.js that serves content over http

<https://www.w3schools.com/nodejs/>

Included project: w2\_tutorials folder

NOTE: Building apps that run in the browser is a completely different thing than building a Node.js application.

About Node.js:

Node.js runs single-threaded, non-blocking, asynchronous programming, which is very memory efficient.

capable of :

- generating dynamic page content

- creating, opening, reading, writeing, deleting, and closing files on the server

- collecting from data

- adding, deleting, modifying data in your database

RUNNING NODEJS PROGRAMS

Write the server side code, save it to a file like prog.js

(1) "var http = require('http')" gives access to the http module

(2) Create a server

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

res.end('Hello World!');

}).listen(8080);

Open CMD and run "node prog.js"

Open your browser and type: http://localhost:{portNumber}/ , i.e. https://localhost:8080

CREATING MODULES

(1) Create a new file with the name of the module, i.e dateTime.js

(2) To use, wrap the module in a require statemen, i.e. **require('./dateTime);**