Section-4

How NodeJS Works

First Off when we use nodejs on computer and that means there is a node process running on a computer Process a program in execution that and that is Single thread running on a computer.

Single Thread:

Initialize Program

Execute Top level code

Requires modules

Register event Call Backs

Event Loop Work (it is the Heart of entire Node Architecture)

Some tasks are expensive to do it in event loop so there we use Thread pool that is provided by Libuv library.

Event Loop

All the Application code that is inside the callback functions that get executed.

NodeJs Is Built all around call back functions.

Event-driven Architecture:

1. Events Are Emitted
2. Event loop pick them up
3. Callbacks are called

Event loop has many phases each phase has one callback queue.

Expired Timer Callbacks

I/O Polling and Callbacks

setImmediate callbacks- These timers are used when we have special function to execute.

Close callbacks this is for the sake of completeness.

Process.NextTick() and other MicroTasks Queue.

If there are any callbacks in the above two queues, then it will immediately get executed after the current phase.

Event Loop makes Asynchronous Programming Possible in Nodejs.

Event – Event Driven Architecture

(Observer pattern)

Event Emitter(Emit event) 🡪 Event Listener(calls) 🡪 Attached Callback Functions

Listener listen to the request made by the observer and respond with the attached Callback functions.

Streams:

Used to process (read and write) data piece by piece (chunks) without completing the whole read and write operation and therefore without completing all the data in the memory.

Perfect for large volumes of data such as videos as used by many companies like Netflix and YouTube there no need to download full file we can read file in small chunks.

More Efficient in data processing no need to keep all the data in the memory and time.

4 types of streams in nodejs

Readable- Streams from which we can read data. http request and fs read stream, pipe () and read () functions are more important in this readable streams.

Writeable-Streams to which we can write data, http responses and fs write stream, write () and end () are important in writeable streams.

Duplex-Streams that are both readable and writeable net web socket ().

Transforms-Duplex streams that transform data as it is written or read

Readable and writeable are most important one.

How Requiring Modules Works:

Each JavaScript file is treated as a separate module;

Node.js uses the Commons module system: require (), exports or module. Exports;

ES module system is used in browsers: import/export;

There have been attempts to bring ES modules to node.js (. mgs).

What happens when we require a Module?

require: function to require modules;

module: reference to the current module;

exports: a reference to module. Exports, used to export object from a module;

\_\_filename: absolute path of the current module’s file;

\_\_dirname: directory name of the current module.

require function returns exports of the required module;

module. Exports is the returned object (important!);

Use module.exports to export one single variable, e.g. one class or one function (module.exports = Calculator);

Use exports to export multiple named variables (exports.add = (a, b) => a + b);

This is how we import data from one module into another.