NODEJS

Shikha Verma
22105031

CONTENTS

WHAT IS NODEJS?

Node.js is an open-source and cross-platform JavaScript runtime environment.Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

KEY FEATURES

(KEYWORDS)

Single Threaded

One thread handles everything.

Asynchro-nous

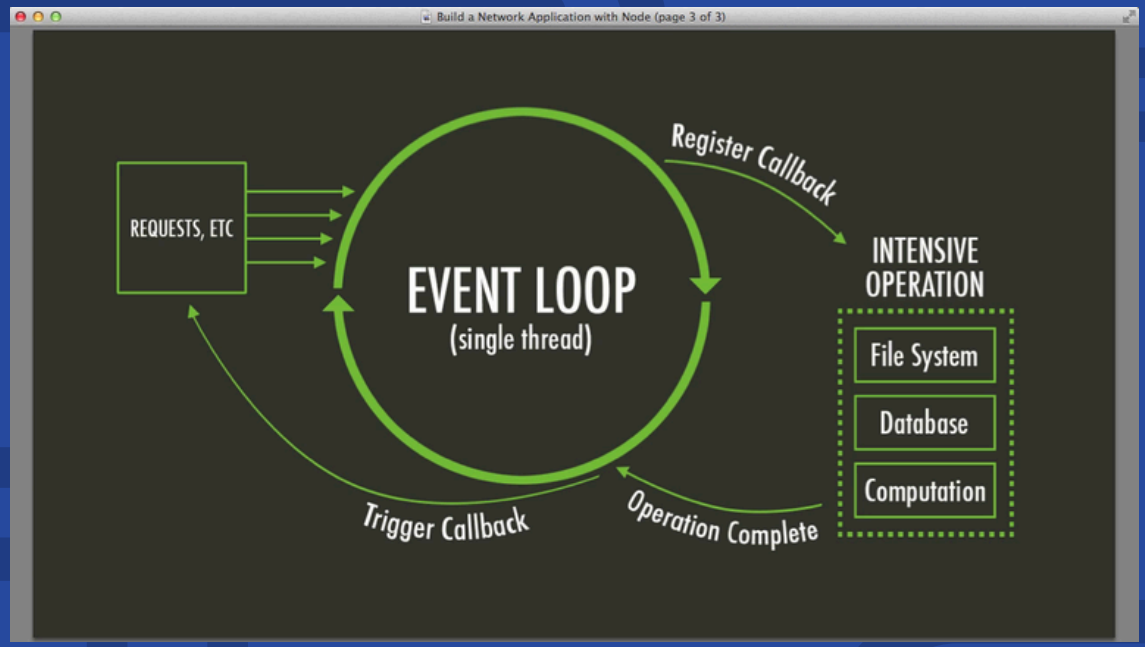
Non-blocking I/O.

Event Driven

Reacts to events efficiently.

FIRSTLY WHAT IS EVENT LOOP?

When running an application, callbacks are entities that have to be handled, in the case of Node.js, event loops are used for this purpose.



The working of an event loop begins with the occurrence of a callback wherever an event begins. This is usually run by a specific listener, Node.js will keep executing the code after the functions have been called, without expecting the output prior to the beginning.

Once, all of the code is executed, outputs are obtained and the function is mecuted,

SINGLE THREADED

NodeJS works on **single thread cocept** i.e. uses single thread to execute a JavaScript

Has single set of resources.

All operations, including handling requests, are managed by this one thread.

If CPU intensive works come then **thread pool** is used ,making it **efficient**.

ASYNCHRONOUS

NodeJS uses concept of **libUV**, special library which provides concept of non-blocking I/O.

It uses **kernel** which has multiple threads.

when CPU intensive work comes, then NodeJS **delegates** these tasks (like reading a file or querying a database) to the event loop.

Thus not blocking the thread.

EVENT DRIVEN

It means that the flow of the program is **determined by events.**

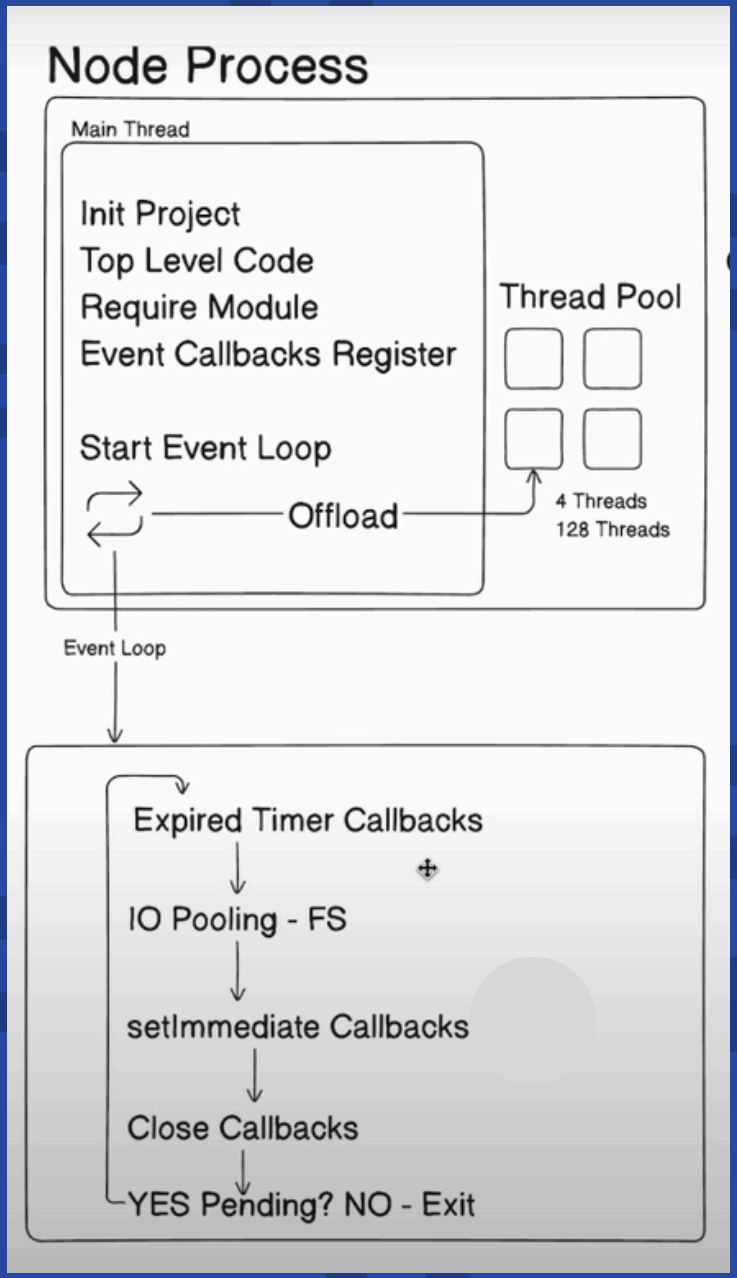
It continuously listens for events, like incoming requests or the completion of I/O operations.

When an event occurs, it triggers a corresponding callback or function to handle it.

This approach allows to efficiently manage multiple tasks, making it responsive and scalable, even on a single thread.

HOW ALL THE THREE FEATURES WORK TOGETHER?

- All operations, including handling requests, are managed by a single thread.
- If there is any CPU intensive work, which will take time, NodeJS does not makes the user wait for the previous task to finish, rather it delegates the task to the thread pool, which contains multiple threads which act like a worker for the main thread. On completion of task, the thread returns the output to the main thread thus not blocking the operation.
- When an operation completes, an event is triggered, and the corresponding callback or event handler is executed. This allows Node.js to efficiently handle multiple operations without waiting for each one to finish before starting the next.



1. Initialise Project
2. executes firstly top level code,i.e.,code is not within callback.
3. Import require modules.
4. Register Event Callbacks.
5. start Event Loop which offloads to thread pool.
    - first priority is given to expired timer callbacks
    - second,I/O pooling such as reading a file etc.
    - thirdly, setImmediate callbacks
    - close callbacks
    - if any callbacks are pending then start from the first, if not, then exit.

THANK YOU!