Event Loop

- By Tarun Singh

Agenda

- What is Event Loop ?
- Blocking Vs Non-blocking
- I/O Operations Vs CPU Operations
- JS as Single-threaded language
- Practical Hands-on (Sync)
- Sync Vs Async
- Practical Hands-on (Async)
- Callbacks
- Practical Hands-on (Callbacks)
- Callbacks Queue
- Practical Hands-on (Timers)
- Phases Of Event Loops

What is Event Loop ?

- Definition straight from the documentation:
  - The event loop is what allows Node.js to perform non-blocking I/O operations — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.
  - Since most modern kernels are multi-threaded, they can handle multiple operations executing in the background. When one of these operations completes, the kernel tells Node.js so that the appropriate callback may be added to the poll queue to eventually be executed. We'll explain this in further detail later in this topic.

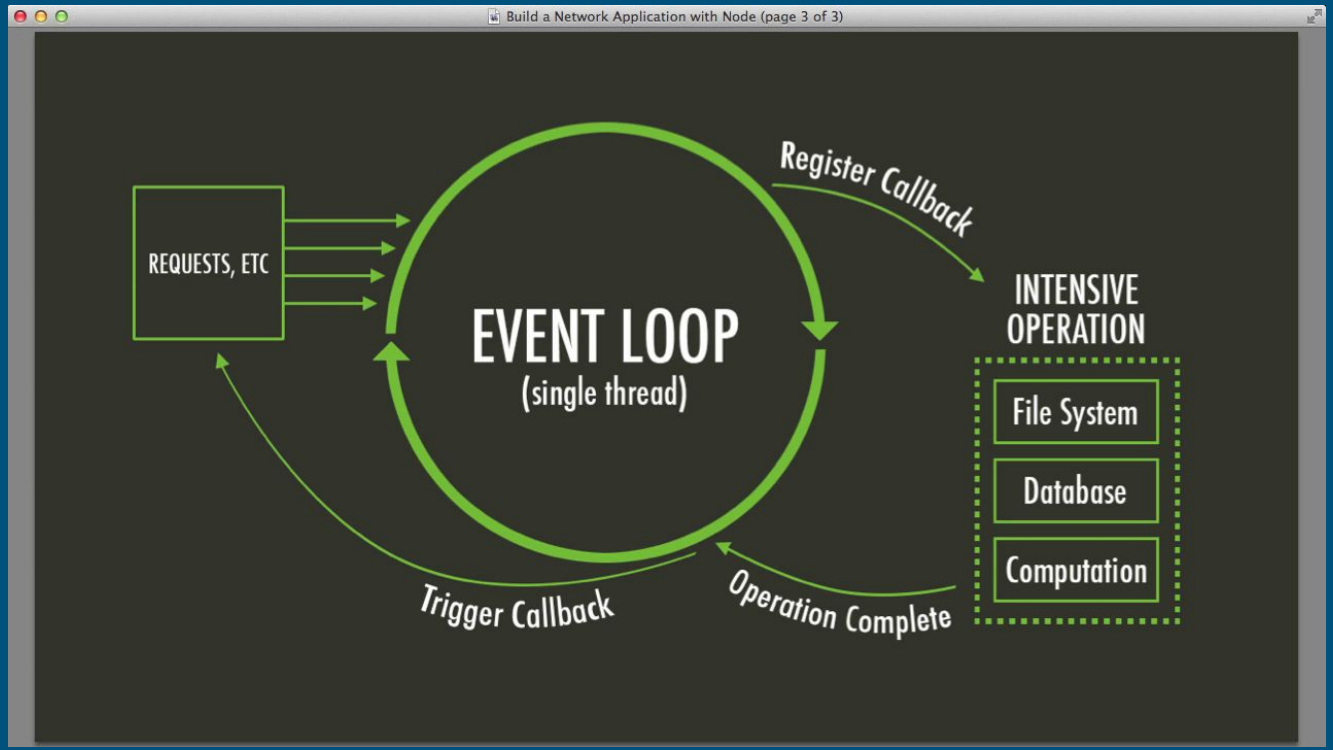
Blocking Vs Non-Blocking

- The current operation occupies the main process thread in NodeJS and blocks future operations until the current task has been completed, this is blocking behaviour.
- In Non-blocking behaviour the main process is able to take future tasks to run along with the current operations.

I/O operations Vs CPU intensive operations

- I/O is nothing but Input-Output task like reading a file or writing a file. These task does not rely on CPU and it is mostly handled by OS or Workers.
- Whereas CPU operations are handle by CPU or Thread Pool, for example calculation based tasks.

So, why JS is called single-threaded ?



Ref: StackOverFlow, NodeJS Event Loop

Sync Vs Async

Synchronous

- In sync, tasks are executed one after the other, so this means code runs in sequence and hence future functions have to wait until the current function has finished its execution.
- In sync operations, the process completes the task then it hands over back the main control, till then no other task can be performed.

Asynchronous

- In async, the tasks are delegated to service workers or thread pool which carry on the task and makes main thread free and hence it can take on new tasks or functions without making the future tasks wait.

What is callback ?

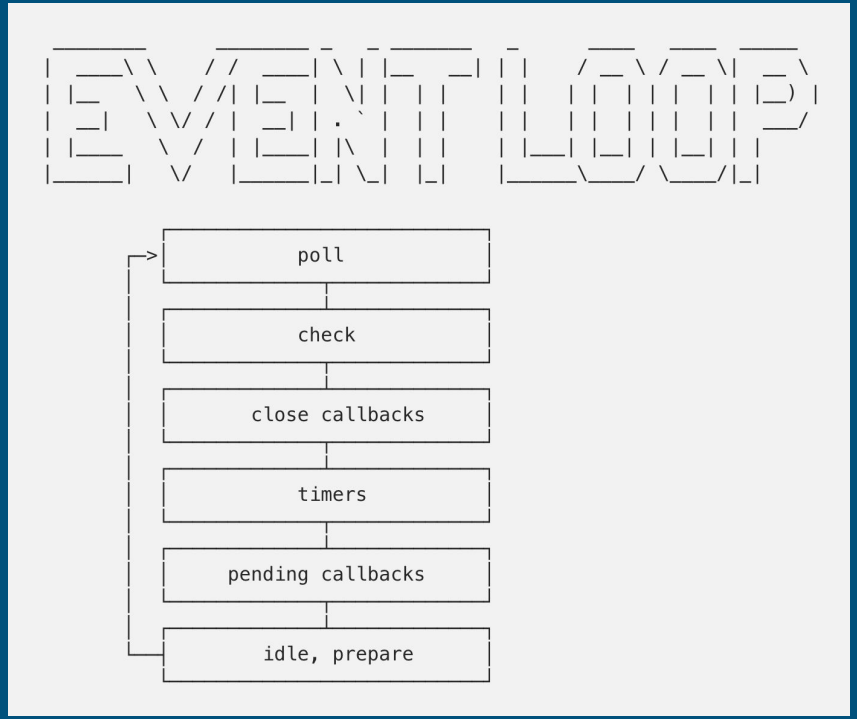
- Callbacks are function which are passed as arguments in other functions and the callback is triggered when the task/request is completed.
- A triggered callback will contain, error if any, data and other options if specified.
- This callback returns the data to main process.
- We can callbacks as signaling mechanism which lets the main thread know that the delegated task has been completed by respective worker.

Callbacks Queue

- The callback queues are queues that hold callback functions to asynchronous operations when they have been completed in the background.
- They work in a first-in-first-out (FIFO) manner.

Phases Of Event Loop

- Poll - The poll phase handles all of the input/output operations we execute.
- All the synchronous JavaScript code you write will be executed in the poll phase of the event loop.
- Timers: this phase executes callbacks scheduled by setTimeout() and setInterval()



Reference: Phases of Event Loop

Thank You!