Libuv

Q1) Is node JS single threaded ?

Q2) Is event loop part of node JS ?

Threading nature of Node JS.

Node JS runs on a single thread

if nodejs is single threaded how is it performant?

Nodejs can utilize multithreading as well. (libuv allows for it).

Java overall is faster than JS. It is also more secure.

What is libuv?

→ written in C

→ provides cross platform support (supports both windows & unix)

→ It is the main driving force to make nodejs support async Programming

→ might have separate implementation for windows & unix platforms

→ TCP connections

→ UDP connection

Where is the implementation of event loop?

Ans - Libuv. ← windows, unix

src/unix/core.c func uv_run. ← logic of event loop.

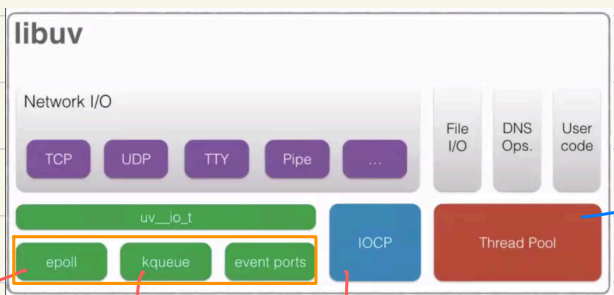
README | MIT license | CC-BY-4.0 license | License | Security

- Asynchronous DNS resolution
- Asynchronous file and file system operations
- File system events
- ANSI escape code controlled TTY — terminal.
- IPC with socket sharing, using Unix domain sockets or named pipes (Windows)
- Child processes
- Thread pool
- Signal handling
- High resolution clock
- Threading and synchronization primitives

Thread pool allows you to do multithreading

→ default 4 (can be changed manually) UV_THREADPOOL_SIZE

Architecture of libuv



event handlers

linux

macos

windows

OS level feature (4)

Is is single threaded but can harness multi threading by power of the runtime environment.

Libuv

Event loop

handles

requests

used in backend servers

handles → long lived resource ex. TCP socket. (stays in memory for long periods)
timers

↳ handles are the main objects that keep nodejs alive.

instructions might be done but process is still alive.

requests → short lived resources. ex file io (file i/o for large files takes more resources)

Libuv allows access to the OS for nodejs.

all async code depends on the OS. Libuv gives access to that part.

* event driven architecture *

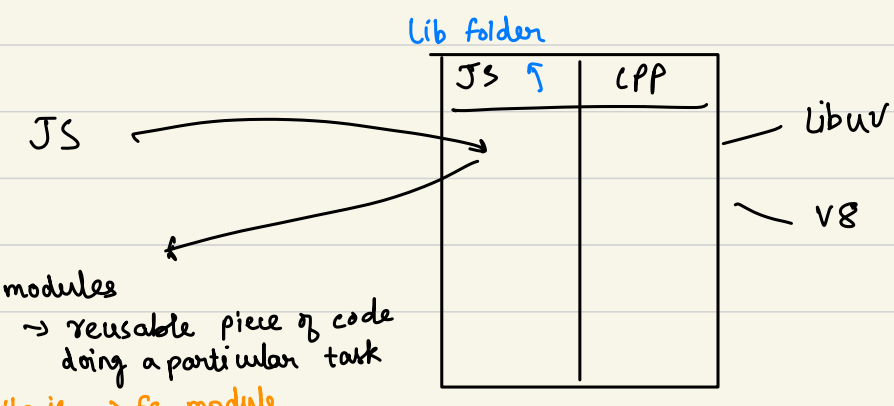
→ Not every operation of libuv accesses multiple threads.

for file i/o or DNS ✓ thread pool

for TCP, UDP ✗ thread pool.

DB query ?

1) TCP connection → libuv.



file io → fs module

fs.js copyfile method

binding.copyfile calls the cpp layer

JS code → cpp layer

↓

libuv implementation

async func^s of crypto library are able to access threads.

Node JS internals - threads

context switching.

sending an http request is just making a TCP connection.

Libuv allocates this task to OS automatically & OS use epoll here as this is an event.

APIs which use the thread pool:

- Every filesystem operation (fs module)
- dns.lookup
- Pipes (edge cases)

APIs which are backed by kernel async operations:

- TCP/UDP servers and clients
- Pipes
- dns.resolve
- Child processes

Basics.