NV1 → event loop

NodeJS → net   fs

JS → CPP

libuv

V8

C-ares

JS

modules → It is a reusable piece of code doing one particular task

file I/O → fs

net

Q:⇒ Is Node JS Single threaded ??

Q:⇒ Is Event loop part of Node JS or it gets it
from Some one ?

↳ Node JS runs on a Single thread.
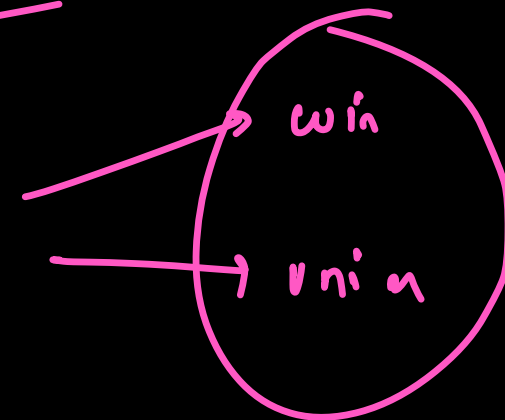
JS ⟶ Single thread

JS ⇐ [node]
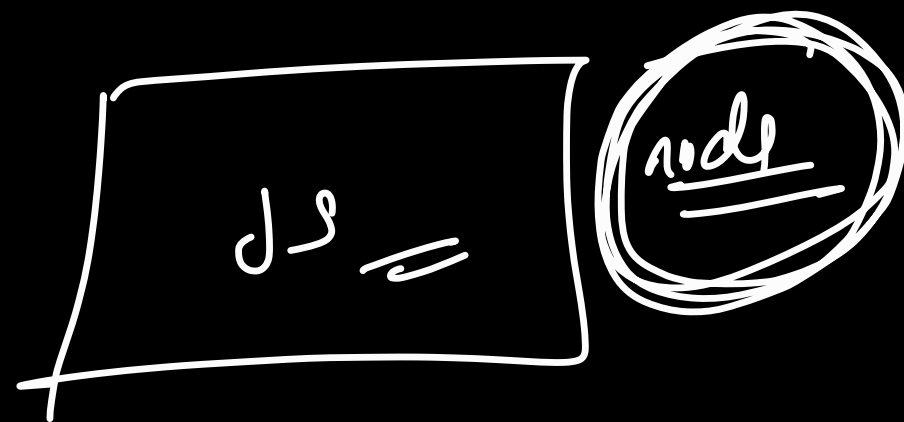
Libuv

→ It's written in C

→ It provides cross platform support

→ It is the main driving force to make node.js Support async programing.

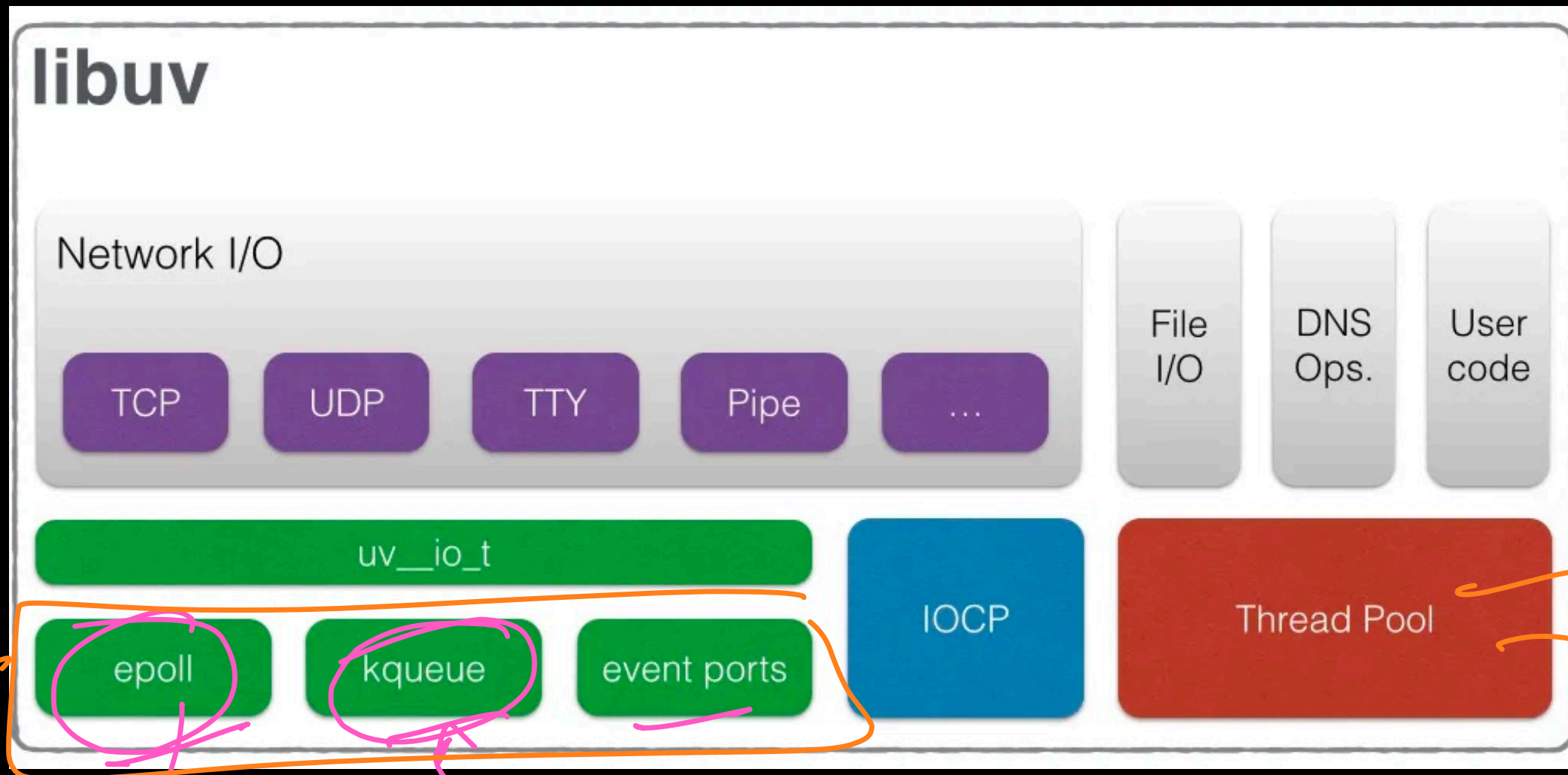→ Support of event loop. → win → unix

→ TCP connection/ TCP sockets

→ UDP ✓✓ → timers

→ file I/O → child process

Java { v m }

→ Thread pool → ④ → configurable

libuv

Network I/O
| TCP | UDP | TTY | Pipe | ... |

File I/O DNS Ops. User code

uv__io_t

epoll kqueue event ports

IOCP

Thread Pool → OS

event loop

Linux

→ macOS →

Win → IOCP

event handler

UV_THREADPOOL_SIZE

→ 4

Event
loop   handles   Requests

handles → There are long lived resources. Ex→ TCP socket
                                                    timers
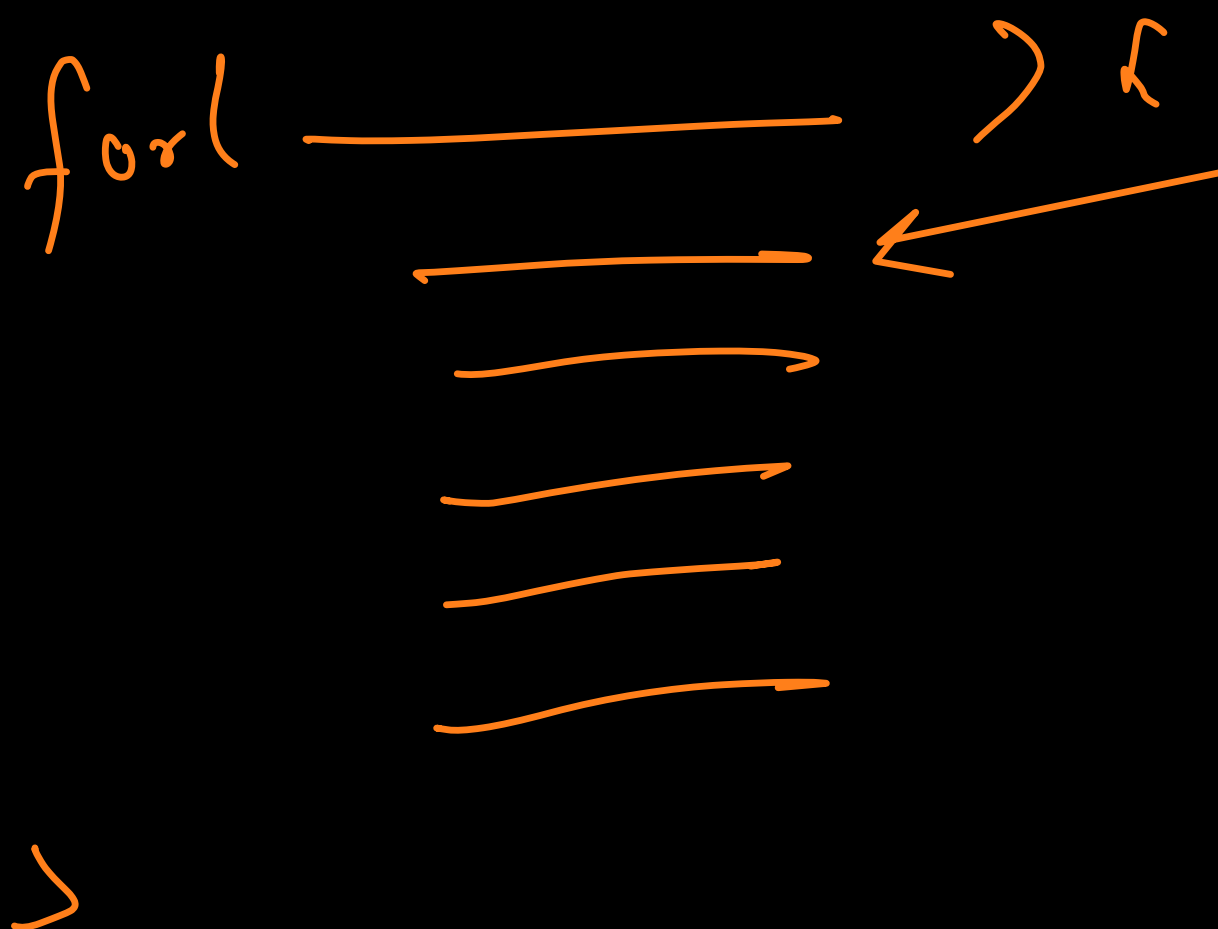   ↳ handles are the main
        objects that keeps NodeJS alive.

Requests → They are short lived resources. Ex→ file I/O

↳ libuv enables access to the OS for node

↳ Not every operation of libuv accesses the thread pool
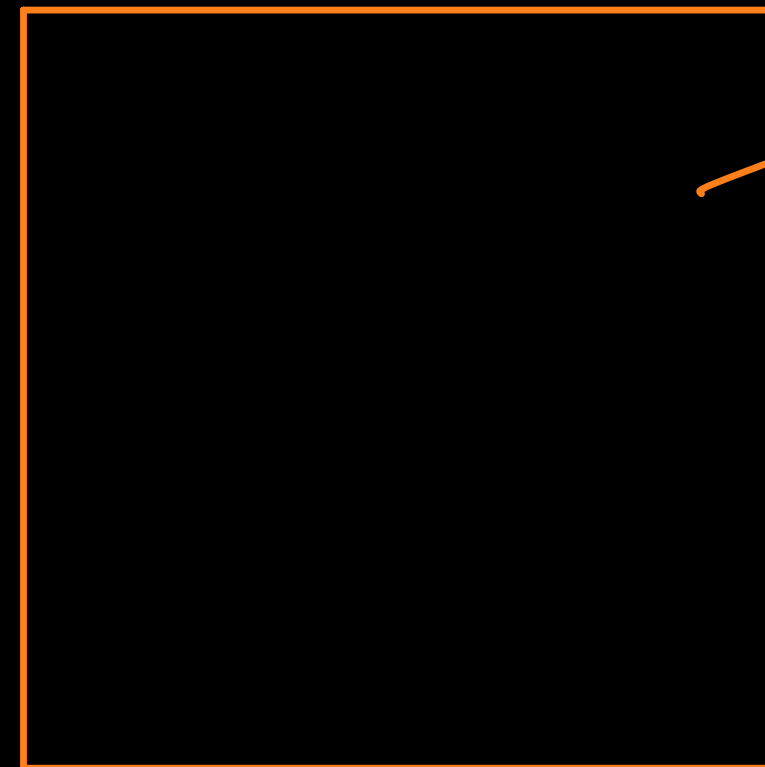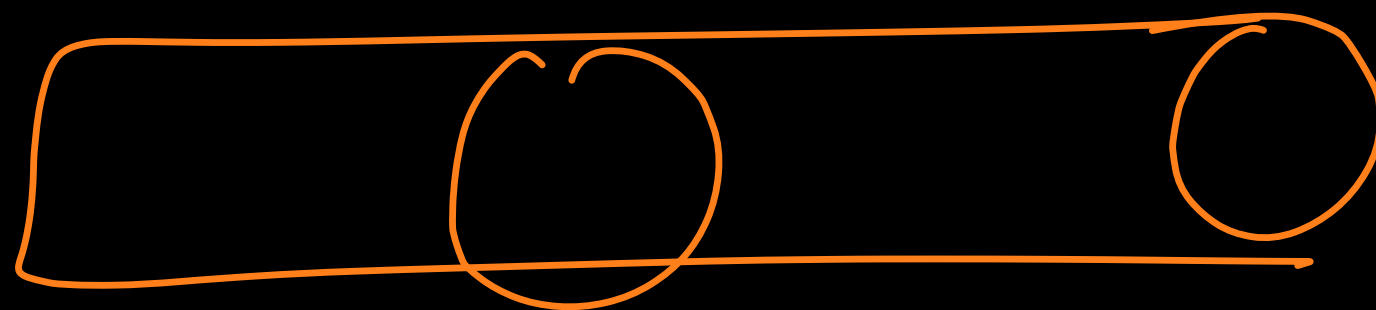
↳ file I/O or DNS → ✓

↳ TCP, UDP ✗

forl

Java (KU)
for ( )

OS

sigh car

$\longrightarrow$ instruction

$\longrightarrow$ iteration

$\longrightarrow$ Experimental Analysis

$\longrightarrow$ Asymptotic Analysis

$\longrightarrow$ Context Switch

$\longrightarrow$ CPU Core

$\longrightarrow$ Process

$\Big\}$