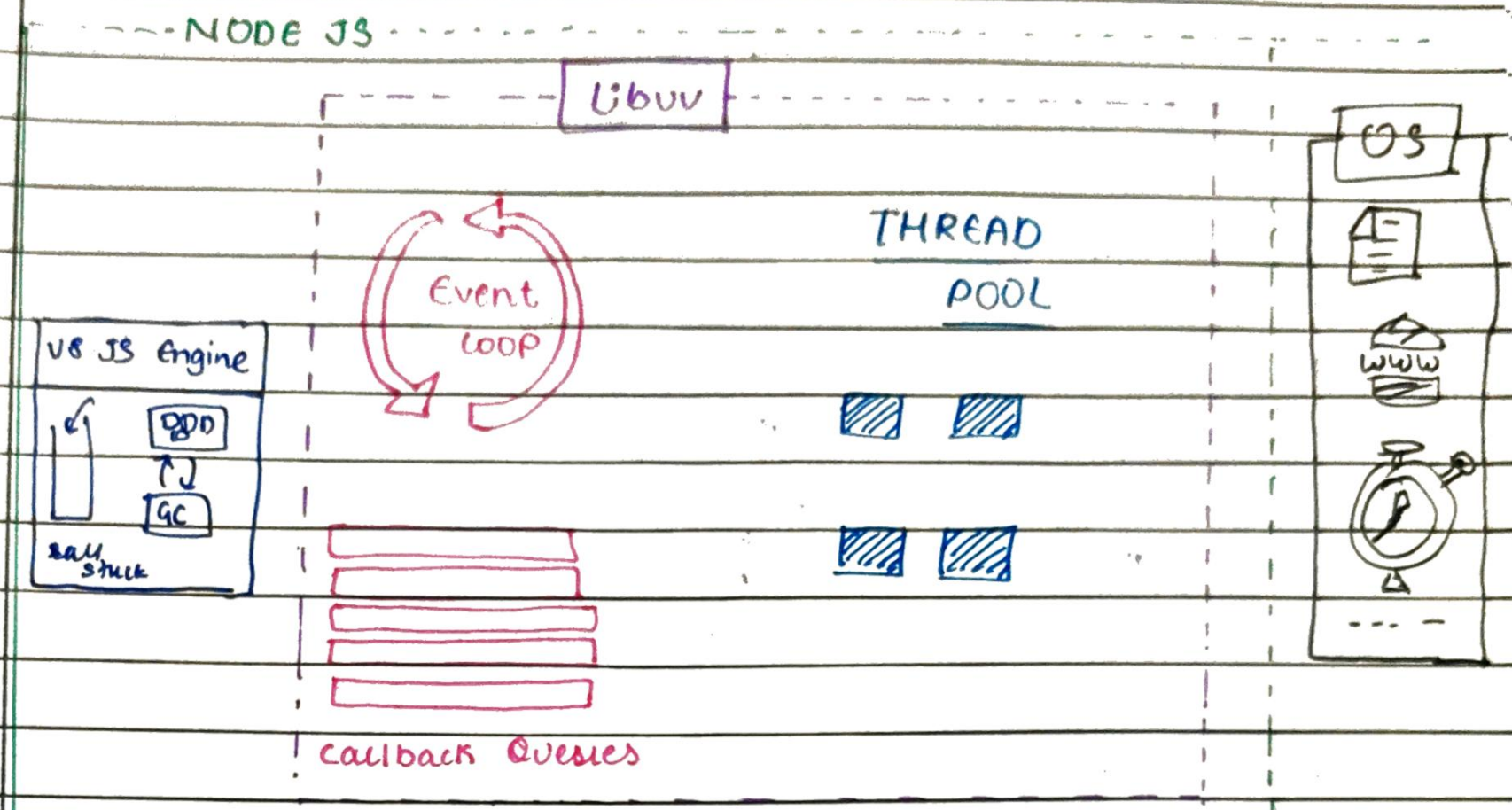
10/12/25

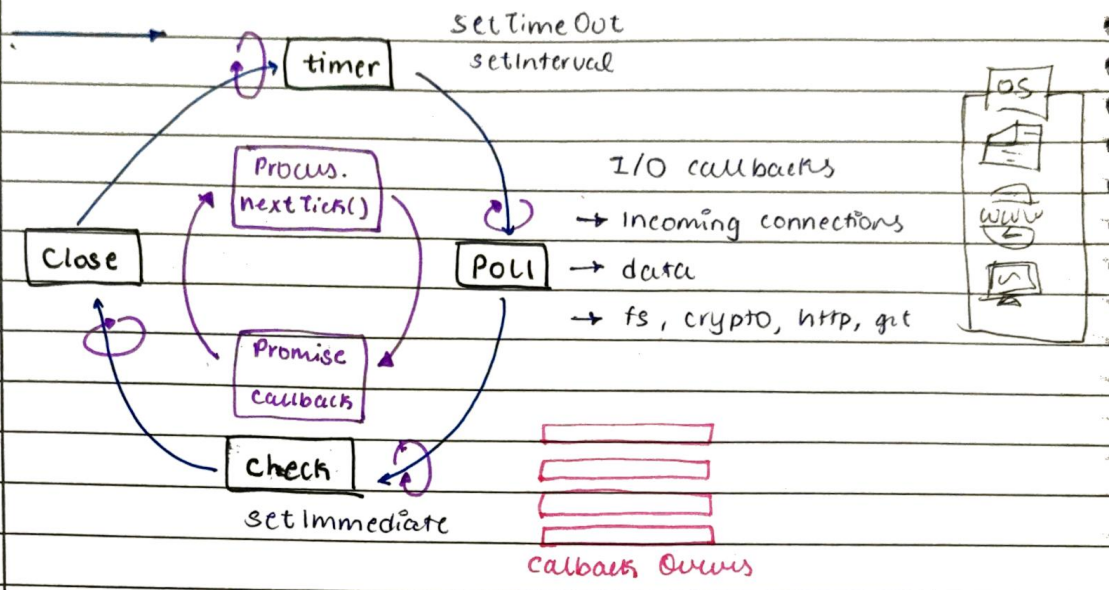
Libuv Event Loop



Asynchronous I/O (Non-blocking I/O)

__/__/____

EVENT LOOP



Example:

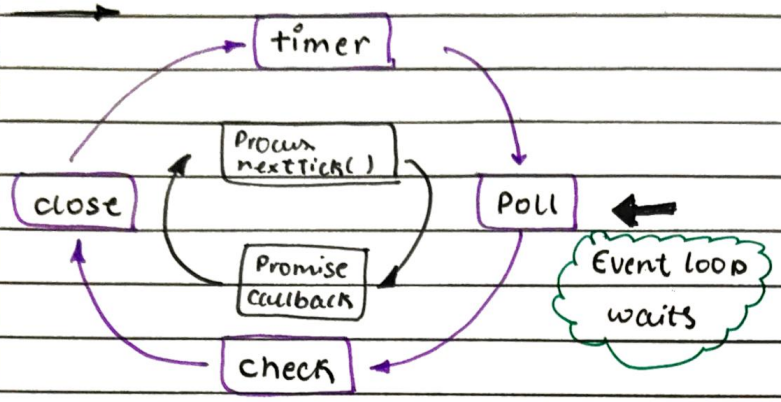
```
procss.nextTick(cb);
Promise.resolve(cb);
setTimeout(cb, 0);
setImmediate(cb);
fs.readFile(".file.txt", cb);
https.get("URL", cb);
```

▼ How event loop works:

→ Phele ton ye procss.next() aur uske baad promise callbacks ko ~~jag~~ run karega.

→ fr time ko firs wapas procss.next() aur promise callback() ~~ko~~ aur uske baad Poll ko mile run hone ko aur ye procss yese chalta jayega.

__/__/__



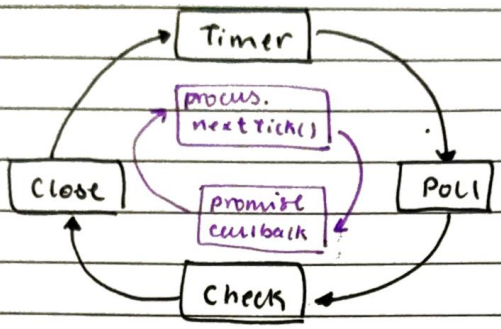
⇒ In nodejs the poll phase is the only phase where the event loop may actually blocks (i.e wait) for:

1. New I/O events
2. New callbacks to be added to the poll queue.
3. A timer to expire.

- When did poll phase wait? (If both are true)

1. There is are NO timer scheduled.
   (no setTimeout, setInterval, etc)
2. The poll query is empty:
   (no completed I/O callbacks waiting to be executed)

__/__/____



set - - -

- — —

- — —

fs.readFile(".\File.txt", 'utf-8', () => {

setTimeOut . .

procss.nextTick . .

setImmediate

console.log

})

procss.nextTick()

console.

=) ye sab run hone ke baad event loop poll phase mai wait karega aur fir jab file read ho jayega toh event loop wapas run hoga per (starting se mai poll phase se run hoga).