Lab12

1.1 What is LibUV?

LibUV is a multi-platform support library with a focus on asynchronous I/O, primarily used by Node.js. It abstracts non-blocking I/O operations (a examples: file system, networking) and offers:

- Event loop
- Thread pool
- Async TCP/UDP sockets
- File system operations
- Child processes
- Timers

It enables Node.js's event-driven architecture, allowing single-threaded code to handle multiple concurrent operations efficiently.

1.2 Difference: setImmediate(f) vs setTimeout(f, Time):

Feature	setImmediate(f)	setTimeout(f, Time)
Purpose	Schedules a function to run after the current poll phase	Schedules a function to run after a delay of at least Time milliseconds
Phase	Executes in the "check" phase of the Node.js event loop	Executes in the "timers phase"
Timing	Runs as soon as the current event loop completes (especially after I/O)	Runs after a minimum delay of Time ms, plus whatever time the event loop needs to become free
Typical Delay	Practically immediate after I/O	At least Time ms (can be longer depending on system load)

1.3 Difference: process.nextTick(f) vs setImmediate(f)

Feature	process.nextTick(f)	setImmediate(f)
Timing	Runs before the next event loop tick	Runs on the check phase
Priority	Higher priority than setImmediate	Lower than nextTick and microtasks
Blocking risk	Can block I/O if overused in loop	Doesn't block I/O
Use case	Small, urgent callbacks before async ops	Post-I/O operation callbacks

2. Output :

```
nextTick 1
Promise.resolve 1
Promise.resolve 2
nextTick inside Promise
this is setTimeout // 0ms
this is setImmediate 1
this is setImmediate 2
Promise.resolve inside setImmediate
readablStream close event
random data
this is setTimeout // 5000ms
```