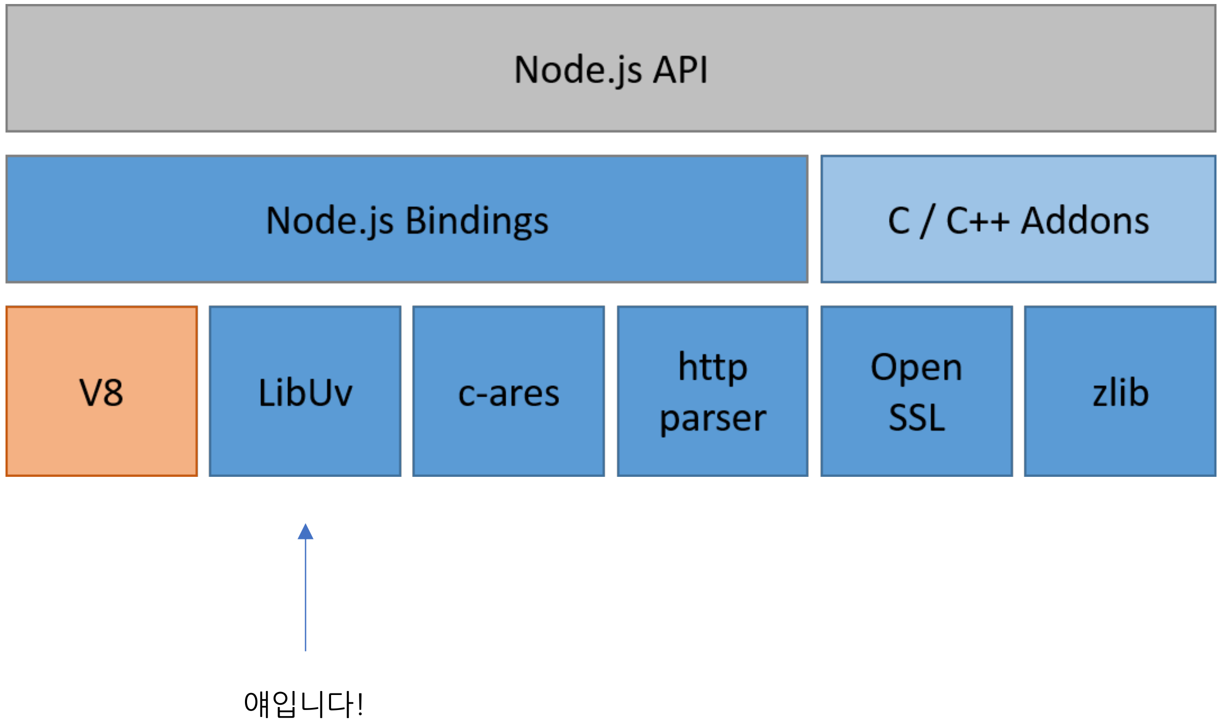
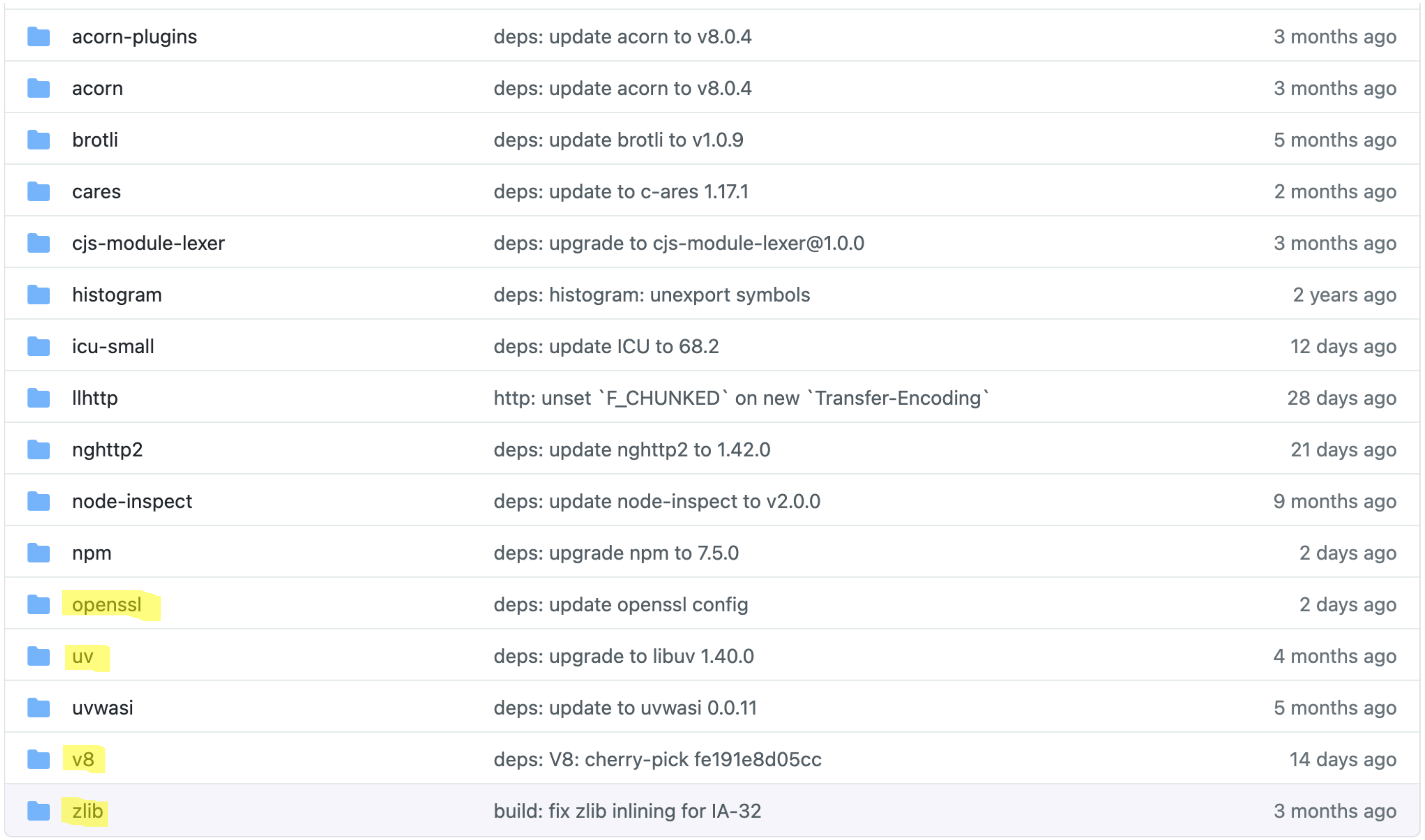
libuv

리뷰비가 크아앙 하고 울부지저따

libuv란?


Node.js API
Node.js Bindings
C / C++ Addons
V8
LibUv
c-ares
http parser
Open SSL
zlib
얘입니다!

libuv란?

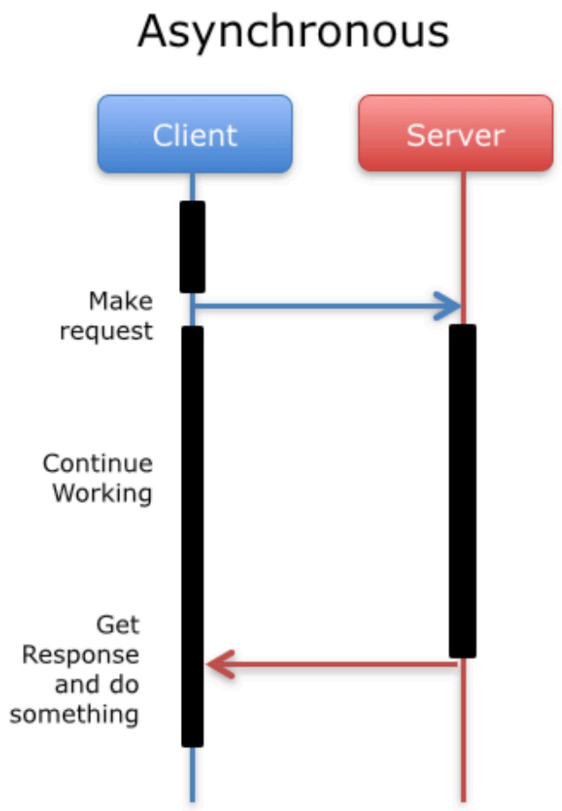


acorn-plugins	deps: update acorn to v8.0.4	3 months ago
acorn	deps: update acorn to v8.0.4	3 months ago
brotli	deps: update brotli to v1.0.9	5 months ago
cares	deps: update to c-ares 1.17.1	2 months ago
cjs-module-lexer	deps: upgrade to cjs-module-lexer@1.0.0	3 months ago
histogram	deps: histogram: unexport symbols	2 years ago
icu-small	deps: update ICU to 68.2	12 days ago
llhttp	http: unset \`F_CHUNKED\` on new \`Transfer-Encoding\`	28 days ago
nghttp2	deps: update nghttp2 to 1.42.0	21 days ago
node-inspect	deps: update node-inspect to v2.0.0	9 months ago
npm	deps: upgrade npm to 7.5.0	2 days ago
openssl	deps: update openssl config	2 days ago
uv	deps: upgrade to libuv 1.40.0	4 months ago
uvwasi	deps: update to uvwasi 0.0.11	5 months ago
v8	deps: V8: cherry-pick fe191e8d05cc	14 days ago
zlib	build: fix zlib inlining for IA-32	3 months ago

libuv란?

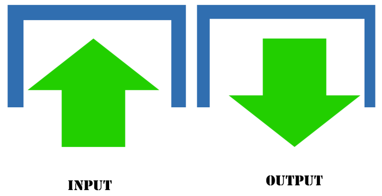
C로 작성되었으며,

노드의 특징인 EVENT LOOP, 비동기 I/O를 가능하게 해준다.



libuv의 구성요소

Event Loop



uv_io

libuv의 구성요소

uv_io

- 작업에 따라 **커널 API 호출 또는 워커 스레드 pool에 넘겨**주는 역할을 한다.
- 작업이 완료되면, 콜백을 큐에 등록한다.
- 워커 스레드는 **Default 4개**이다.

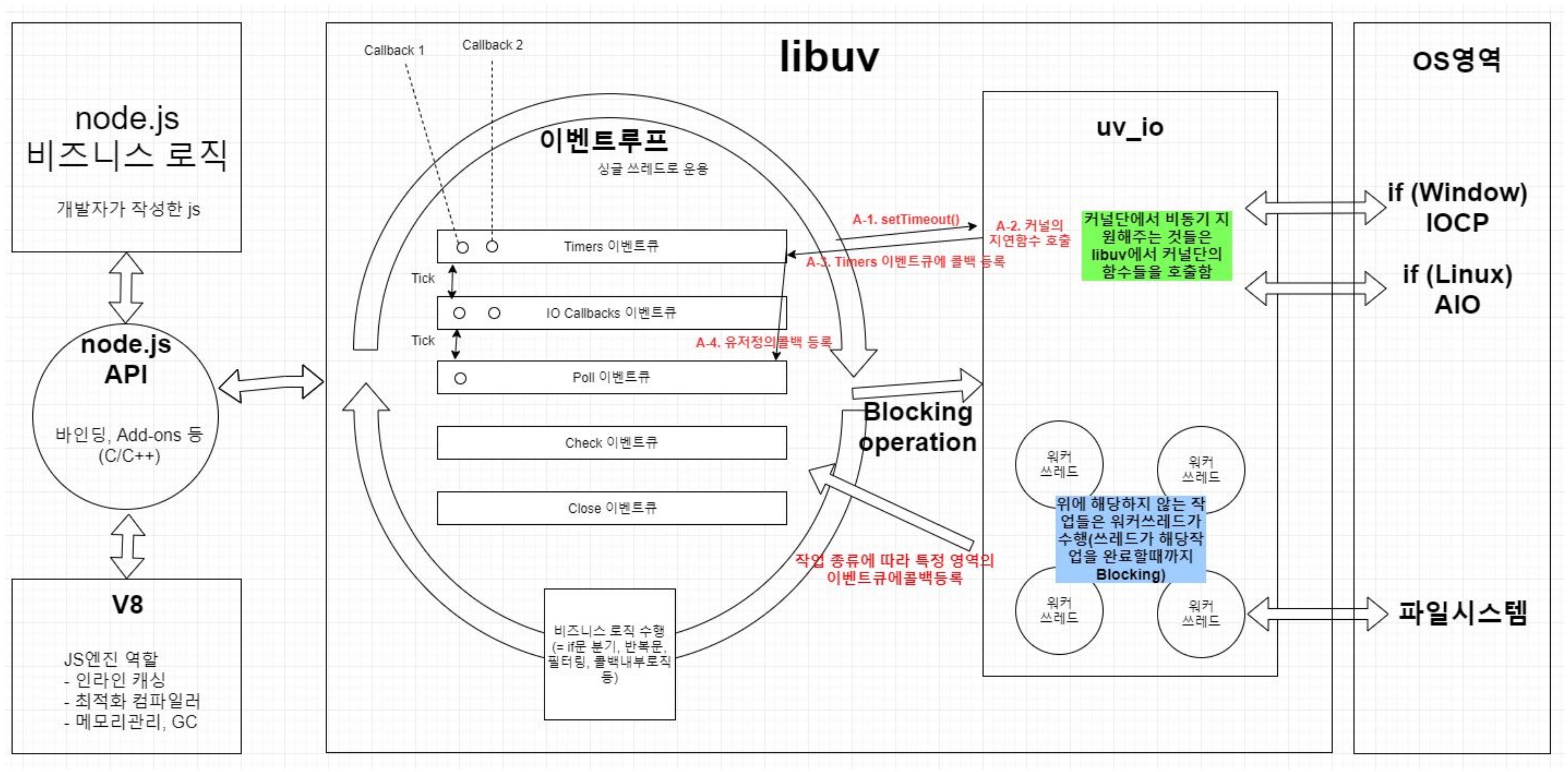
> 커널에서 작업이 끝나면 SIGNAL을 통해 작업 완료를 알린다!

libuv의 구성요소

event loop

- **uv_run() 을 통해 호출된다.**
- **메인 스레드**이자, **싱글 스레드**이다.
- **비지니스 로직**을 수행한다.
- phase를 넘어가며 콜백함수를 실행한다.

구성도


node.js
비즈니스 로직
개발자가 작성한 js
node.js
API
바인딩, Add-ons 등
(C/C++)
V8
JS엔진 역할
- 인라인 캐싱
- 최적화 컴파일러
- 메모리관리, GC
libuv
Callback 1
Callback 2
이벤트루프
싱글 쓰레드로 운용
Timers 이벤트큐
Tick
IO Callbacks 이벤트큐
Tick
Poll 이벤트큐
Check 이벤트큐
Close 이벤트큐
비즈니스 로직 수행
(= if문 분기, 반복문,
필터링, 콜백내부로직
등)
A-1. setTimeout()
A-2. 커널의
지연함수 호출
A-3. Timers 이벤트큐에 콜백 등록
A-4. 유저정의콜백 등록
Blocking
operation
작업 종류에 따라 특정 영역의
이벤트큐에콜백등록
uv_io
커널단에서 비동기 지
원해주는 것들은
libuv에서 커널단의
함수들을 호출함
워커
쓰레드
워커
쓰레드
위에 해당하지 않는 작
업들은 워커쓰레드가
수행(쓰레드가 해당작
업을 완료할때까지
Blocking)
워커
쓰레드
워커
쓰레드
OS영역
if (Window)
IOCP
if (Linux)
AIO
파일시스템

event loop 코드

setTimeout(Interval)에 등록된 콜백을 가지는 큐

setTimeout(Interval), setImmediate를 제외한 콜백

setImmeditae에 등록된 콜백을 가지는 큐

```c
while (r != 0 && loop->stop_flag == 0) {
  uv__update_time(loop);
  uv__run_timers(loop);
  ran_pending = uv__run_pending(loop);
  uv__run_idle(loop);
  uv__run_prepare(loop);

  timeout = 0;
  if ((mode == UV_RUN_ONCE && !ran_pending) || mode == UV_RUN_DEFAULT)
    timeout = uv_backend_timeout(loop);

  uv__io_poll(loop, timeout);

  /* Run one final update on the provider_idle_time in case uv__io_poll
   * returned because the timeout expired, but no events were received. This
   * call will be ignored if the provider_entry_time was either never set (if
   * the timeout == 0) or was already updated b/c an event was received.
   */
  uv__metrics_update_idle_time(loop);

  uv__run_check(loop);
  uv__run_closing_handles(loop);

  if (mode == UV_RUN_ONCE) {
    /* UV_RUN_ONCE implies forward progress: at least one callback must have
     * been invoked when it returns. uv__io_poll() can return without doing
     * I/O (meaning: no callbacks) when its timeout expires - which means we
     * have pending timers that satisfy the forward progress constraint.
     *
     * UV_RUN_NOWAIT makes no guarantees about progress so it's omitted from
     * the check.
     */
    uv__update_time(loop);
    uv__run_timers(loop);
  }

  r = uv__loop_alive(loop);
  if (mode == UV_RUN_ONCE || mode == UV_RUN_NOWAIT)
    break;
}
```

감사합니다