// Example 1
// Idealized event loop usage

initialize state

while (event = get event) {
    switch (event.type) {
        case readable:
            // decide what read action is appropriate
            read (event.fd);
            // update state
            break;
        case writable:
            // decide what write action is appropriate
            write (event.fd);
            // update state
            break;
    }
}

// Example 2
// Top-level driver loop for an event-driven programming library.

list<when, callback> timeouts;
callback fds[...];

while amain() from main() {
    while (1) {
        select() for fds[] and earliest timeout;
        for each readable fd
            cb = fds[selread][fd].cb();
        for each writable fd
            cb = fds[selwrite][fd];
            cb();
        if a timeout has expired
            cb = timeouts.pop();
            cb();
    }
}

// register cb to be called when fd is ready for op (selread or selwrite)
// Set to NULL to clear
fdcb(fd, op, cb) {
    fds[op][fd] = cb;
}

// register cb to be called at specified time
delaycb(when, cb) {
    timeouts.push(list(when, cb));
}