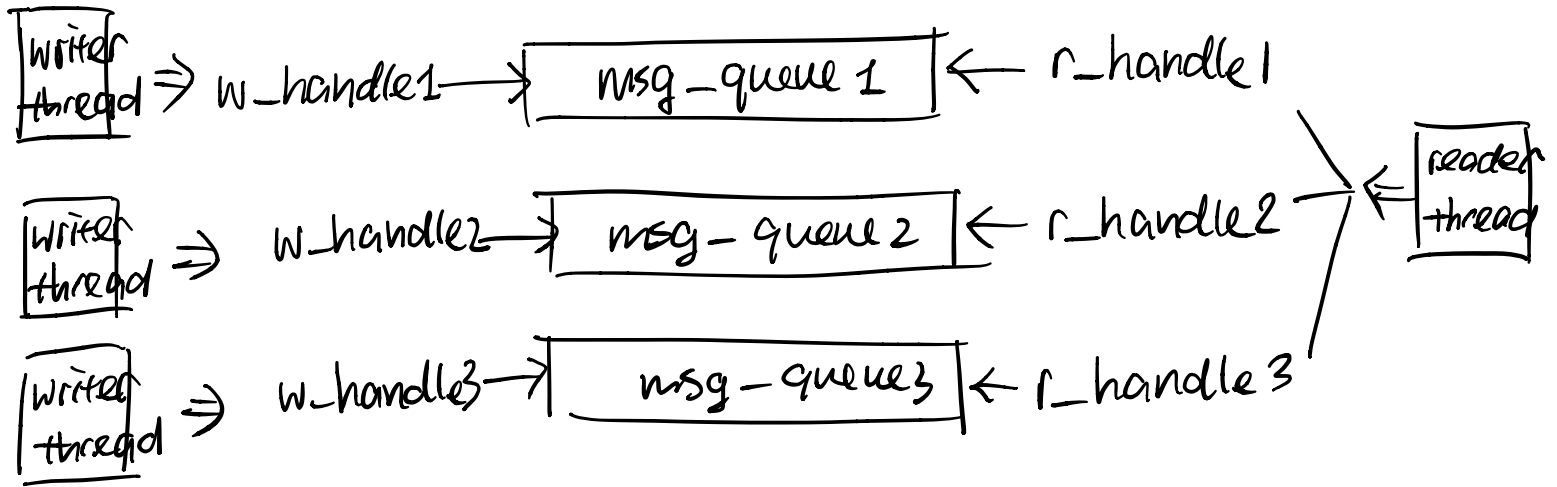
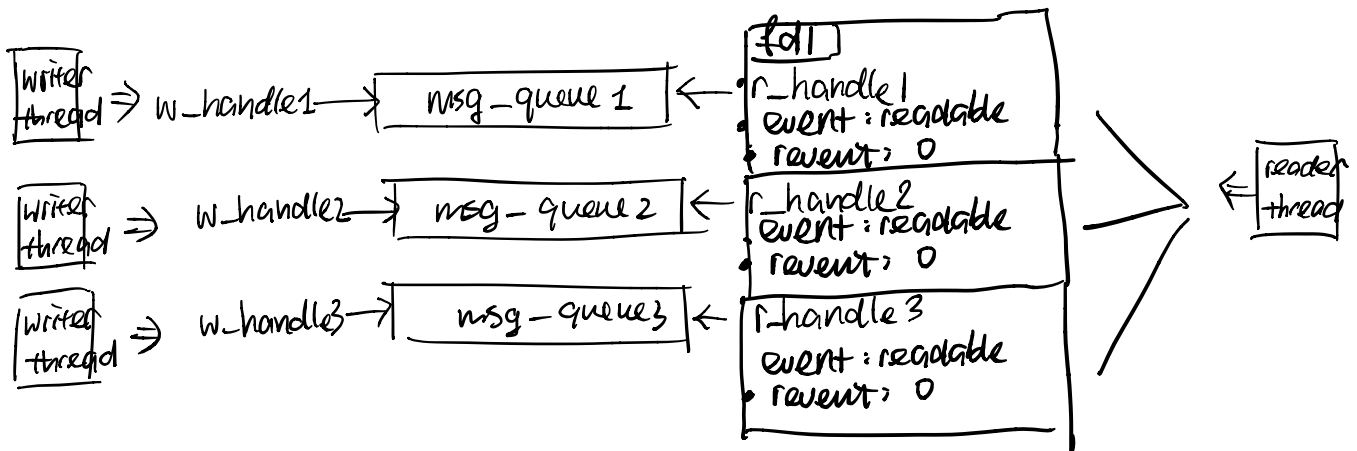


1. 已经打开 3 个 write handle 和 1 个 read handle.



当 reader thread 要 poll 时.

1). 生成 fds [3].



2). 进入 msg_poll function (reader thread 里面 call poll)

① ~~✱~~ init 一个 cond_t thread-active (假设可以放在每一个 queue wait_list 里, 只需要一个 lock)

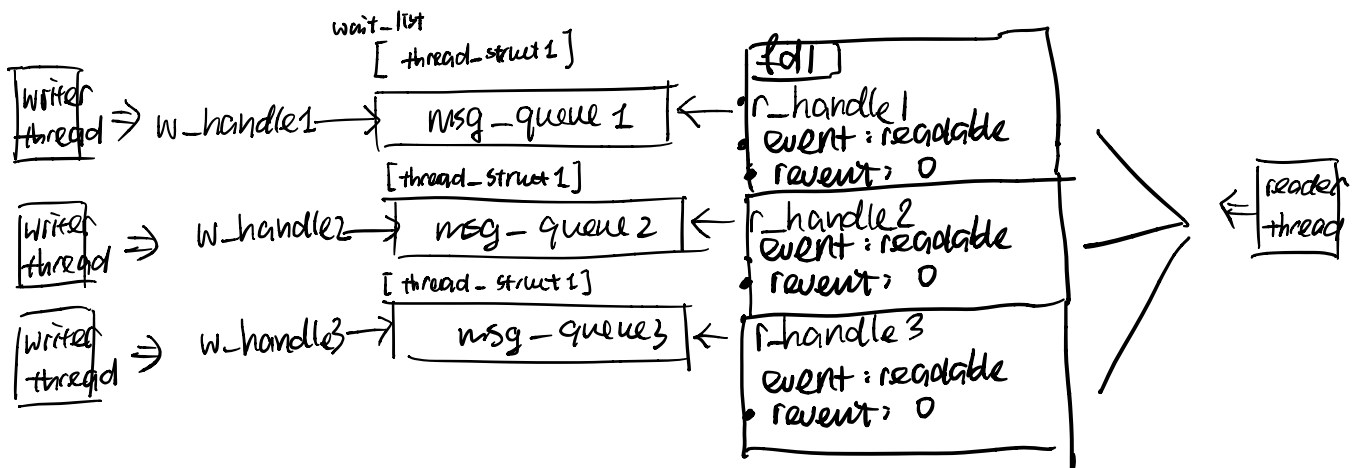
init 一个 mutex thread-lock 何 queue 发生 event, 都会激活这个 signal
(synchronize thread)

② 怎么把这个 thread-active 激活键放进每个 queue 里??

自己 define 一个 struct, 跟 list.h 连起来,

```
struct thread {
    entry (用来 implement list.h)
    cond-t active
    mutex-t thread-lock
    fd[] (用来查看 events, 和 set revents)
}
```

③ 给每个 fd[] 放进一个 thread struct.
这样就实现了 所有 queue 共用一个 active button.



④. check 有沒有 events 是沒 call poll 之前就快沒了
了的。

如果有, set revents. 直接 return. # of revents.
如果沒有, wait (thread_active, thread_lock)
return. # of revents.