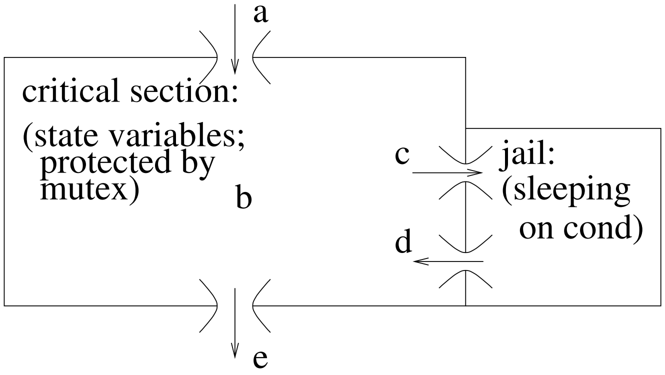
NOTE: We assume that using a resource may be slow, but acquiring and releasing a resource is fast.

ACQUIRE RESOURCE:

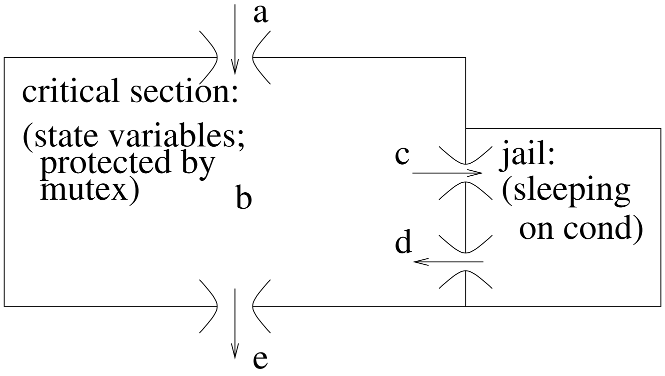


```
pthread_mutex_t mutex;
pthread_cond_t cond;
        a. pthread_mutex_lock(&mutex)
        b. modify state variables;
           while (! condition(state)) {
              pthread_cond_wait(&cond, &mutex);
           }
        c. enters "jail" due to pthread_cond_wait()
        d. leaves "jail" due to pthread_cond_signal()
           or pthread_cond_broadcast()
        e. pthread_mutex_unlock(&mutex)
           # And if releasing a resource, then
           # call pthread_cond_signal() or
           # pthread_cond_broadcast to allow
           # other threads to leave the "jail".
```

USE RESOURCE:

Read, write, create new resource, delete resource, or whatever else

RELEASE RESOURCE:



NOTE: When releasing a resource, we will still modify the state variables in part b, and it is important to call pthread_cond_signal/broadcast, but we usually don't need to test a condition and wait.