Lecture 27

25 September 2018 10:40

Synchronization

Spin locks

```
getlock:  mov R0, 1
          Test and Set LOCK, R0
          JNZ getlock
          RET

unlock:   mov LOCK, 0
```

① Spin lock + disabling interrupts

② Q. Is disabling interrupts required for user space synchronization?
└ user threads, user data.

mutex / condition variables
sleep – wakeup locks

```
getlock:  mov R0, 1
          Test and Set LOCK, R0
          JZ DONE
          CALL YIELD        ← move process from RUNNING to READY
          ⇒ JMP getlock
DONE      : RET

unlock:   ??
          mov Lock, 0       option 0
```

design! condition variables.

reason for wakeup or sleep.

└ identifier / on what are you contending for?

① mutex —— condition & if the mutex lock 0/1

```
struct mutex {
    mutex_lock ;     ← state of the lock 0/1
    variable   id;   ← identifier variable
};

getlock:
    CheckLock ();
    if not available
        sleep (m->id);   ← changed from yield to sleep.
        jmp getlock();   ← RUNNING to
```

unlock;

$m \rightarrow L = 0;$

wakeup $(m \rightarrow id)$;

// move thread(s) to READY

↑ RUNNING to BLOCKED

process