Implementing synchronization constructs

Two equivalent approaches :

- Either implement locks first
  and build semaphores and condition variable on the top

- Or implement semaphores first (Pintos approach)
  and build locks and condition variable on top

A **very bad** implementation of a lock

```
struct lock {
}

void acquire (lock) {
    disable_interrupts();
}

void release (lock) {
    enable_interrupts();
}
```

➡ Disabling interrupts blocks notification of external events that could trigger a context switch

- Can miss or delay important events
- The thread is no longer preemptive