Implementing monitors in a single-processor kernel

```
Monitor Entry:
  Disable interrupts
  if (!inMonitor)
    inMonitor := 1
  else {
    Append(entryQ, currentThread)
    Sleep
  }
  Enable interrupts

Monitor Exit:
  Disable interrupts
  if (isEmpty(entryQ))
    inMonitor := 0
  else
    Append(readyQ, Remove(entryQ))
  Enable interrupts

Wait:
  Disable interrupts
  Append(waitQ, currentThread)
  if (isEmpty(entryQ))
    inMonitor := 0
  else
    Append(readyQ, Remove(entryQ))
  Sleep
  Enable interrupts

Signal:
  Disable interrupts
  if (!isEmpty(waitQ))
    Append(entryQ, Remove(waitQ))
  Enable interrupts
```