

Ahmad Malik

12/7/22

ECE357

PSET 6

Problem 1:

- a) It is valid because it can block all signals before entering the critical region and then later restore them to keep other tasks from disrupting the CPU.
- b) Not valid because although the code can try to block signals, there are many other threads in a region of memory that can access control other than signal handlers.
- c) Not valid because there are many other threads that can access control, only signal handlers won't work.
- d) It is valid since because all signals can be masked before entering the crucial region and then restore them once we have left to stop any other jobs from influencing the CPU.
- e) Not valid because the code would use the local CPU and any other process that uses the local CPU would affect it.

Problem 2:

The code is susceptible to deadlock since there are several instances in which a single job would have to wait for the completion of another if both are running concurrently. If tasks A and B lock R1 and R2, respectively, when task A or tasks B tries to lock R1 or R2, respectively, they both have to wait and become trapped in a deadlock. It isn't livelock because the jobs never loop and never return to their initial state after completing their processes. So the tasks wait indefinitely to unlock one another's locks.