References:
   RTC Chapters 5 and 6
   CUG Chapter 12

Where appropriate, give specific RTEMS directives and lines of code.

Please type all responses. Hand written work will not be accepted.

1. When a semaphore is released and there are tasks in the task-waiting list, does RTEMS allow the highest priority task or the task that has been waiting longest acquire the semaphore?

2. With regard to mutexes as described in the RTC, does RTEMS support mutex ownership? Recursive locking? Task deletion safety?

3. Which protocol does RTEMS support for priority inversion avoidance, priority inheritance or ceiling priority?

4. How would you create an RTEMS counting semaphore with an initial value of 10 that uses a priority wait queue?

5. How would you create an RTEMS mutex that uses the priority inheritance protocol?

6. Assume you use the RTEMS acquire directive with a 10 tick timeout. Upon return, how can you determine whether you obtained the semaphore or the timeout occurred?

7. Some RTOSes allow you to poll a semaphore (check availability without acquiring the semaphore). RTEMS does not directly support polling (being able to poll a semaphore is of questionable value), but it can be effectively implemented. Write an rtems_semaphore_poll() routine that takes a single argument of type rtems_id (the semaphore id) and returns an rtems_status_code type. It should return RTEMS_SUCCESSFUL if the semaphore is available and RTEMS_UNSATISFIED if it is not available.