Review Questions and Problems Chapter 3

1. Assume that a single event is transmitted in a one byte state message with a period of 50 msec. What is the finest temporal resolution of the time of event occurrence that can be encoded in this one-byte message?

2. What is the *difference between sampling and polling*?

3. Why is an interrupt potentially dangerous and when is it needed?

4. How can you protect a computer system from the occurrence of sporadic *erroneous interrupts*?

5. What are *accuracy limits* of an analog signal in typical industrial applications?

6. Sketch the *software* routine for a contact switch *eliminating* the contact *bounce*.

7. For the Dragon12 board, write a routine that *detects* the maximum *number of contact bounces*, and the maximum bounce time for switch SW5, and display these values on the LCD.

8. Explain the differences between the DI-part, I-part, and S-part of an interrupt service routine in the RMOS operating system.

9. What are the characteristics of a *fail-silent* and *triple-modular redundancy* actuator?

10. What are the advantages of an *intelligent instrument*?

11. Give an example of a *fault-tolerant sensor*.

Review Questions and Problems Chapter 7

1. Why is *concurrency* so important in real-time computer systems?

2. There are three ways to *implement concurrency*. Name them and explain how they work.

3. Explain the difference between *POSIX processes* and POSIX *threads*. In most real-time systems you do not find POSIX processes. Please explain why not.

4. What is the purpose of the *task control block* (TCB)? Please list some of the more important parts of a TCB and explain their function.

5. In a real-time operating system, what makes up the *context* and what happens during a *context switch*?

6. Is it possible that a task that has suspended itself can resume itself again? Please explain.

7. What criteria are needed to determine the size of the run-time stack for a task and for the system stack in a multiple (nested) interrupt system?

8. Tasks in a real-time operating system can reside in a number of *states*. Sketch a typical *task state diagram* and explain the different states. Why is it not possible for a task to switch from “runnable” to “blocked”?

9. Explain the difference between an *active* and a *passive object*.

10. Consider a software system with an RTOS and a priority based scheduler. The main task has priority 60. It creates two new tasks T1 and T2 with priorities 70 and 50, respectively. It then starts Task T2, which in turn starts Task T1.

    Draw the *activity diagram* for this scenario. Draw the *timing diagram* for this scenario.

11. For the following code fragment for a priority based scheduling on a real-time operating system, how many different outputs are possible, considering different priorities?

```
void task1(void)
{
   createAndStartTask(task2, prio2);
   printf("1");
}

void task2(void)
{
   printf("2");
}

int main()
{
  createAndStartTask(task1, prio1);
  printf("0");
  exit(0);
}
```

12. In many general purpose operating systems, context switches from one task to the next are enforced when a task has exceeded a certain run time, called the “quantum”. Assuming that a context switch takes TC seconds, and the quantum is TQ seconds, give an expression for the computational efficiency η in terms of TC and TQ, assuming that the execution times of the tasks are much larger than TQ.

Review Questions and Problems Chapter 9

1. Consider a task set T composed of the following three periodic tasks {T1, T2, T3}:

    - T1(0,1,3,3) (release time, computation time, deadline, period)
    - T2(0,1,4,4)
    - T3(0,2,3,6)

    a) Compute the processor utilization factor and the major cycle for the task set.

    b) Build the schedule for the task set under the three scheduling algorithms RM, DM, and EDF.

2. Consider a task set with the following three periodic tasks {T1, T2, T3}:

    - T1(0, 2, 7) (release time, computation time, period)
    - T2(0, 1, 4)
    - T3(0, 1, 8)
    - T4(0, x, 5) (x is to be determined)

    a) Compute the relative priorities for this task set according the RM algorithm.

    b) How much computation time x can be granted to T4 such that schedulability can be guaranteed for all four tasks under the RM algorithm?

    c) How much computation time x can be granted to T4 such that schedulability can be guaranteed under the EDF algorithm?

3. On a given system with a round-robin scheduler (time slice scheduler) an average task runs for a time T until it is blocked by an input/output operation. A context switch takes a fixed time S; this time is lost for the actual purpose of the system.

    Please compute the CPU efficiency for this round-robin scheduler with a quantum Q (the time slice granted to each task before a context switch is enforced), the efficiency being the ratio of usable CPU time to total CPU time for the following cases:

    (a) $Q = \infty$
    (b) $Q > T$
    (c) $S < Q < T$
    (d) $Q = S$
    (e) Q almost 0

4. Four Tasks, T1 to T4, are released simultaneously to run on a single CPU. Their computation times are 7, 4, 8, for T1, T2, and T3, respectively, and some time X for T4.

    In what order should the scheduler execute the four tasks such that the average response time for this task set is minimized?

    Solution hints:

    - To compute the average response time, first assume four tasks Ta, Tb, Tc, and Td are being executed in that order.
    - Compute the actual response time for task Ta with computation time a, for task Tb with computation time b, and so on.
    - For this case, develop a formula for the average response time.
    - Now assign the computation times of T1 to T4 to their proxies Ta to Td such that the average response time is minimized.
    - The answer depends on X, thus there is more than one solution.

Review Questions and Problems Chapter 10

1. Consider a task set T composed of the following three periodic tasks:

    - T1(0, 5, 25, 30) (release time, computation time, deadline, period)
    - T2(0, 10, 40, 50)
    - T3(0, 20, 55, 75)

    The task set is scheduled with the EDF algorithm.

    a) Verify the schedulability under the EDF algorithm. Build the corresponding schedule. What are the idle times of the processor?

    Consider the following aperiodic tasks:

    - T4(40, 10, 15) (release time, computation time, deadline)
    - T5(70, 15, 35)
    - T6(100, 20, 40)
    - T7(105, 5, 25)
    - T8(120, 5, 15)

    b) Can these requests be guaranteed in the idle times of the processor?

2. Consider a task set T composed of the following three periodic tasks:

    - T1(0, 5, 30) (release time, computation time, period)
    - T2(0, 10, 50)
    - T3(0, 25, 75)

    a) Compute the major cycle of the task set. Verify the schedulability under the RM algorithm. Build the schedule.

    Consider the following aperiodic tasks:

    - T4(5, 12) (release time, computation time)
    - T5(40, 7)
    - T6(105, 20)

    b) The aperiodic tasks are scheduled in the background. Compute the response times of tasks T4, T5, and T6.
    c) The periodic tasks are scheduled with a server. The server capacity is set to 5 and its period is set to 25. Verify the schedulability of the new task set. Build the schedule. Consider that the server is a polling server. Compute the response times of tasks T4, T5, and T6.