# Homework #5 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

•Grading: 3 = correct

2 = almost

1 = an attempt

0 = nothing

•Score: Points / Possible

# (41 points) (Name) (Section)

**Scheduling (Chapter 9)**

**RT Scheduling (Chapter 10)**

|  |  |
| --- | --- |
| Questions: | Answers: |
| 1. (8 points) Suppose that an automobile microcontroller has four priority levels for interrupts (level 1 being the highest). The units interrupting the controller include an impact sensor subsystem, which needs attention within 0.1 ms, along with four other subsystems as follows:   |  |  |  | | --- | --- | --- | | **Subsystem** | **Interrupt rate** | **Max service time (ms)** | | Fuel/ignition | 500/sec | 1 ms | | Engine temperature | 1/sec | 100 ms | | Dashboard display | 800/sec | 0.2 ms | | Air conditioner | 1/sec | 100 ms |   Use rate monotonic scheduling to assign a priority to each subsystem that guarantees the response time of 0.1 ms for the impact sensor and handles each of the other critical units before the next interrupt is produced.  a. Justify your priority assignments.  b. Does RMA guarantee schedulability? | Period  Level 1 Impact sensor Who cares  Level 2 Dashboard display 1.25 ms  Level 3 Fuel/ignition 2.0 ms  Level 4 Air conditioner 1000 ms  Engine temperature 1000 ms  RMA  Periods: F/I = 2 ms DD = 1.25 ms  ET = 1000 ms AC = 1000 ms  Utilization:  F/I = 1/2 = 0.5% DD = 0.2/1.25 = 0.16%  ET = 100/1000 = 0.1% AC = 100/1000 = 0.1%  U4 = (0.5 + 0.1 + 0.16 + 0.1) = 0.86 < 1.00%  W4 = 4 × (2(1/4) - 1) = 0.76 < 0.86  Hence RMA doesn’t guarantee schedulability. |
| 2. (12 points) The following C program is to be executed on a computer and a parallel version is to be executed on a 32-computer cluster.  **L1: for (i = 0; i < 1024; i++)**  **L2: { sum[i] = 0;**  **L3: for (j = 0; j <= i; j++)**  **L4: { sum[i] = sum[i] + i;**  **L5: }**  **L6: }**  Suppose lines 2 and 4 each take two machine cycle times, including all processor and memory-access activities. Ignore the overhead caused by the software loop control statements (lines 1, 3, 5, 6) and all other system overhead.  a. What is the total execution time (in machine cycle times) of the program on a single computer?  b. Divide the i-loop iterations among the 32 computers as follows: Computer 1 executes the first 32 iterations (i = 0 to 31), processor 2 executes the next 32 iterations, and so on. What is the execution time and speedup factor compared with part (a)? (Note that the computational workload, dictated by the j-loop, is unbalanced among the computers.)  c. Show how to modify the parallelizing to facilitate a balanced parallel execution of all the computational workload over 32 computers. (A balanced load means an equal number of additions assigned to each computer with respect to both loops.)  d. What is the execution time resulting from the balanced parallel execution on 32 computers? What is the resulting speedup over a single computer? (Show your calculations.) | a. T = 2 × (1024 + (1 + 2 + … + 1024))  (1 + 2 + … + 1024) = n(t1 + tn)/2 = 524,800  T = 2 × (1024 + 524800) = 1,051,648  b. C1 = 2 × (32 + (1 + 2 + … + 32))  = 2 × (32 + 32 × (1 + 32) / 2) = 592  C2 = 2 × (32 + (33 + 34 + … + 64))  = 2 × (32 + 32 × (33 + 64) / 2) = 3168  …  C32 = 2 × (32 + (993 + 994 + … + 1024))  = 2 × (32 + 32 × (993 + 1024) / 2) = 64,608  1,051,648 / 64,608 = 16.3 times faster  c. Identify:  hotspots (where the real work is being done)  data dependence  Investigate other algorithms  sum(1) = 2 × (1 + 1024) = 2050  sum(2) = 2 × (1 + 1023) = 2048  …  sum(1023) = 2 × (1 + 2) = 6  sum(1024) = 2 × (1 + 1) = 4  C1 do sum(1-16), sum(1009-1024)  C2 do sum(17-32), sum(993-1008)  …  C32 do sum(497-528)  d. C32 = 2 × (32 + (497 + 498 + … + 528))  = 2 × (32 + 32 × (497 + 528) / 2) = 32,864    1,051,648 / 32,864= 32 times faster |
| 3. (9 points) Use Rate-monotonic Analysis (RMA) to answer the following questions:   |  |  |  | | --- | --- | --- | | **Process** | **Execution Time** | **Period** | | P1 | 1 ms | 8 ms | | P2 | 2 ms | 5 ms | | P3 | 2 ms | 10 ms |   a. What is the CPU utilization?  b. What is the theoretical limit for 3 processes?  c. Is this system schedulable? | a. U3 = (1/8 + 2/5 + 2/10) = 0.725  b. W3 = 3 × (2(1/3) - 1) = 0.78  c. Since 0.725 < 0.780, the system is schedulable! |
| 4. (10.1, page 477) (12 points) Consider a set of three periodic tasks with the execution profiles and arrival order shown below. Develop scheduling diagrams similar to those of Figure 10.5 for this set of tasks. Round-robin the two lower priority tasks. (Use ascending alphabetical order for simultaneous arrivals.) Preempt only on task arrivals. | |
| |  |  |  |  | | --- | --- | --- | --- | | **Process** | **Arrival Time** | **Execution Time** | **Ending Deadline** | | **A(1)** | **0** | **10** | **20** | | **A(2)** | **20** | **10** | **40** | | **A(3)** | **40** | **10** | **60** | | **A(4)** | **60** | **10** | **80** | | **A(5)** | **80** | **10** | **100** | | **…** | **…** | **…** | **…** | | **B(1)** | **0** | **10** | **50** | | **B(2)** | **50** | **10** | **100** | | **…** | **…** | **…** | **…** | | **C(1)** | **0** | **15** | **50** | | **C(2)** | **50** | **15** | **100** | | **…** | **…** | **…** | **…** | | **10_5** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | A1 | |  |  | A2 | |  |  | A3 | |  |  | A4 | |  |  | A5 | |  |  | A6 | |  |  | |  | B1 | |  |  |  |  |  |  |  |  | B2 | |  |  |  |  |  |  |  |  | B3 | |  |  | |  | C1 | | |  |  |  |  |  |  |  | C2 | | |  |  |  |  |  |  |  | C3 | | |  | |  | 0 | | 10 | | 20 | | 30 | | 40 | | 50 | | 60 | | 70 | | 80 | | 90 | | 100 | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | A1 | |  |  | A2 | |  |  | A3 | |  |  |  | |  |  |  | |  |  |  | |  |  | |  |  |  | B1 | |  |  |  |  |  |  | ↓ |  |  |  |  |  |  |  |  |  |  |  |  |  | | A Priority |  |  |  |  |  |  | C1 | |  |  | C2 |  |  |  |  | |  |  |  |  |  |  |  |  | |  | 0 | | 10 | | 20 | | 30 | | 40 | | 50 | | 60 | | 70 | | 80 | | 90 | | 100 | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  | A1 | |  |  |  | A2 | A3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | B1 | |  |  |  |  |  |  | ↑ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | B Priority |  |  |  |  | C1 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | 0 | | 10 | | 20 | | 30 | | 40 | | 50 | | 60 | | 70 | | 80 | | 90 | | 100 | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | A1 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | C1 | | |  | ↑ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | C Priority |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | 0 | | 10 | | 20 | | 30 | | 40 | | 50 | | 60 | | 70 | | 80 | | 90 | | 100 | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | A1 | |  |  | A2 | |  |  |  | A3 | |  | A4 | |  |  |  |  | A5 | |  |  |  |  | | Earliest |  |  | B1 | |  |  |  |  |  |  |  | B2 |  |  |  |  |  | B2 |  |  |  |  |  |  | | Deadline |  |  |  |  |  |  | C1 | | |  |  |  |  |  | C2 | | |  |  |  |  |  |  |  | |  | 0 | | 10 | | 20 | | 30 | | 40 | | 50 | | 60 | | 70 | | 80 | | 90 | | 100 | |  | | | |