**EASTERN MEDITERRANEAN UNIVERSITY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CMPE 443**

**Real-Time Systems Design**

**Final Exam**

**2012/13 Fall Semester**

**January 8, 2013**

**Name-Surname : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Number : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Instructor:

Assoc. Prof. Dr. Alexander CHEFRANOV

Time: 120 minutes

MOBILES ARE NOT ALLOWED

YOU CAN BRING ONE A4 SIZED SHEET OF *HAND-WRITTEN* NOTES TO THE EXAM. PHOTOCOPIES ARE NOT ALLOWED AND WILL BE COLLECTED.

READ THE INSTRUCTIONS FOR EACH SECTION CAREFULLY.

**Grade**

|  |
| --- |
|  |

**Task 1.** (20 points) Build a Rate-Monotonic schedule for tasks:

|  |  |  |
| --- | --- | --- |
| Task# | E | P |
| 1 | 6 | 12 |
| 2 | 2 | 8 |
| 3 | 1 | 6 |

What are the utilization and hyperperiod for this set of tasks? Are the deadlines met by the RM schedule? Explain why they are met (not met)?

U=6/12+2/8+1/6=(12+6+4)/24=22/24, H=24

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 |
| Task | 3 | 2 | 2 | 1 | 1 | 1 | 3 | 1 | 2 | 2 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 1 |  |  |

The task releases, deadlines, and completion times are as follows

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Release time | Deadline | Completion time |
| 3 | 0 | 6 | 1 |
|  | 6 | 12 | 7 |
|  | 12 | 18 | 13 |
|  | 9 | 12 | 19 |
| 2 | 0 | 8 | 3 |
|  | 8 | 16 | 10 |
|  | 16 | 24 | 18 |
| 1 | 0 | 12 | 12 |
|  | 12 | 24 | 22 |

We see that all completion times do not exceed respective deadlines, hence all deadlines are met

**Task 2.** (20 points) Assume, two parallel processes, A1, A2, run concurrently. Process A1 waits for a command from A2, executes it and signals to A2 about its completion. A2 waits for completion of the work by A1. When the work is done, A2 processes results and then issues the next command to A1. Write a pseudo-code using binary semaphores to provide necessary synchronization of the processes A1. A2. Define necessary data structures. Show initial settings of the semaphores you use.

Sema waitA2=0/\*closed\*/, waitA1=0/\*closed\*/;

Process A1{

While(1){

P(waitA2); get(command); execute(command); V(waitA1);

}

}

Process A2{

While(1){

send(command); V(waitA2); P(waitA1); process(result);

}

}

**Task 3.** (20 points) Assume that a system has 5 processes and resources of one type: memory (total available number of memory blocks is 10). Processes’ resources required and maximal requirements are as follows:

|  |  |  |
| --- | --- | --- |
| Process | Memory required | Memory maximal required |
| 1 | 2 | 6 |
| 2 | 1 | 5 |
| 3 | 2 | 4 |
| 4 | 1 | 3 |
| 5 | 2 | 5 |

Use the Banker algorithm to decide on safety of granting required resources. Show all the steps of the algorithm you make to come to your decision.

Number of available resources if required resources are granted is r=a-(c1+c2+c3+c4+c5)=10-(2+1+2+1+2)=10-8=2

Denote maximal required resources as b1,..,b5

Safe sequence S={} initially

B1-c1=4<=r=2? no

B2-c2=4<=r=2? No

B3-c3=2<=r=2? Yes => S={P3}//extend S by process P3

B1-c1=4<=r+c3=2+2=4? Yes => S={P3,P1}// extend S by P1

B2-c2=4<=r+c3+c1=2+2+2=6?yes =>S={P3,P1,P2}

B4-c4=2<=r+c3+c1+c2=2+2+2+1=7? Yes =>S={P3,P1,P2,P4}

B5-c5=3<=r+c3+c1+c2+c4=2+2+2+1+1=8? Yes =>S={P3,P1,P2,P4,P5}

As far as S includes all the processes, there desired state is safe, and the resources might be granted.

**Task 4.** (20 points) Assume, we have a priority preemptive system (without time sharing for same priority tasks) and the following system of processes:

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Execution sequence | Release time |
| A | 2 | VQVVE | 1 |
| B | 4 (highest) | EQEQ | 4 |
| C | 1 (lowest) | QQVE | 0 |
| D | 3 | VVVVQE | 1 |

Show time diagrams of execution for each process, if Immediate Ceiling Priority Protocol is used. Calculate response time for each process A, B, C, D. Specify dynamic priority for each task and each time unit when it runs.

Response time for A is 16, for B is 4, for C is 19, for D is 11

Priority of each task is equal to its static priority all the time excepting task C having dynamic priority 4 for time in [0,2].

**Task 5.** (20 points) Assume, clocks may work in the following modes:

* normal mode;
* setting of clocks;
* setting/resetting of timer (expires after the specified time interval, once or periodically);
* setting/resetting of alarm (expires when the specified time is reached, once or periodically).

Define necessary buttons (signals) and build a Finite State Automaton modeling the clocks.

se

>,<,^,V,1,+

Set clocks

>,<,^,V

Set clocks

Set timer

Set timer

>,<,^,V,1,+

Set alarm

Set alarm

>,<,^,V,1,+

>,<,^,V,1,+

Signals <,> move cursor on clocks left-right, ^ increases clocks, V decteases, 1 is used for one time action, + for periodic action