SYSC-5207 Distributed Systems Engineering (Fall 2014)

Assignment 2

Show your work. Justify your answers.

[Students will work in groups (of two) that have already been determined. Each group must hand in only ONE set of solutions to the problems]

1. Consider a single CPU-based hard real time system on which 4 processes are running under the Rate Monotonic Scheduling discipline. The execution time and period for each process are given below:

Process	Execution Time	Period (ms)
P1	T	50/C
P2	T/2	10/C
P3	T/3	40/C
P4	T/4	100/C

(a) Given that C>0, determine the priority for each process
(b) With C = 5, Find the maximum value of T for which all the processes are guaranteed to meet their deadlines.
(c) With T = 12, find the maximum value of C for which all the processes are guaranteed to meet their deadlines.

Note that the deadline for each process is equal to its period

2. (a) Consider a single CPU-based hard real time system on which P processes are running under the Rate Monotonic Scheduling discipline. The period of each process is x times its execution time. Determine the maximum value of P for which all the processes are guaranteed to meet their deadlines for x = 1, 2, 3 and 4. Plot your results in the form of a line graph.

(b) Consider the system described in 2(a) but the processor speed is half of that used in 2(a). Find the maximum value of P for x = 2.

Note that for both (a) and (b) the deadline for each process is equal to its period

3. Consider a system with 5 processes:

Process	Execution Time (ms)	Period (ms)
P1	10	50
P2	15	80
P3	25	75
P4	30	60
P5	35	140

Determine the minimum number of CPUs you need such that all processes are guaranteed to meet their deadlines when RMS is used on each CPU. Note that the speed for each CPU is different. The first CPU achieves the execution times described earlier. The speed of the ith CPU in the system is 1/i of the speed of the first CPU.

Note that the deadline for each process is equal to its period.

4. In *work-conserving* scheduling a processor is never kept idle if there are one or more ready processes on the system.

Can a non-work-conserving schedule be the optimal schedule that produces the highest speedup for some concurrent application running on a non-pre-emptive environment on some given number of processors?

If your answer is NO then prove that a non-work-conserving schedule can never be the optimal non-pre-emptive schedule for any concurrent application. If your answer is YES, show with the help of an example that such a non-work-conserving schedule can actually be the optimal non-pre-emptive schedule for a given application and system.

[Assume that the system you are concerned with contains three processors and the applications that run on the system contain 8 or more processes and are characterized by at least two levels of precedence relationships among some of the processes].

5. Two parallel applications App1 and App2 have the same average parallelism A. Each of them takes T units of time to complete when executed on 1 processor. Consider running each of them separately on N processors leading to a completion time of t1(N) for App1 and t2(N) for App2. Derive an expression for the upper bound on the ratio of t1(N) and t2(N).

Posted: October 22
Due: November 5 (in class)