|  |
| --- |
|  |
| **Assignment # 2**  **SYSC 5207 – Distributed Systems Engineering** |
| Fall 2014  Submitted To  Dr. Shikharesh Majumdar  By  **Ferhan Jamal (100953487)**  **Rudraneel chakraborty (7563772, 100992549)**  **Daman Saluja (100958057)** |

**Answer : 1**

It is given in the question that there is a single CPU-based hard time real time system on which 4 processes are running under the Rate Monotonic Scheduling discipline. The execution time and process for each process are given below:

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

1. In the 1st part, we have to determine the priority for each process with C>0. In RMS, the tasks are ordered in non-decreasing order of period. The lower is the period, the higher is the priority in RMS. Therefore, the priorities of the processes are as follows:

|  |  |  |
| --- | --- | --- |
| Process | Priority | Period |
| P2 | 1 | 10/C |
| P3 | 2 | 40/C |
| P1 | 3 | 50/C |
| P4 | 4 | 100/C |

1. In the 2nd part of the question with C=5, we have to find the maximum value of T for which all the processes are guaranteed to meet their deadlines.

A set of independent periodic tasks schedule by the RMS algorithm will always meet the deadline for all tasks if

T1/C1 + T2/C2 +…………….TN/CN <= n(21/n - 1) ---------(1)

|  |  |  |
| --- | --- | --- |
| Process | Execution Time (Ti) | Ci (ms) |
| P1 | T | 50/C or (10) |
| P2 | T/2 | 10/C or (2) |
| P3 | T/3 | 40/C or (8) |
| P4 | T/4 | 100/C or (20) |

In the equation (1); the value of C1, C2, C3 and C4 will be 10, 2, 8 and 20. Putting these values in equation (1):

T/10 + T/4 +T/24+T/80 <= 4(21/4 - 1)

By calculating, we have :

T (0.4041) <= .7568

T <= 1.8724

Maximum value of T for which all processes are guaranteed to meet deadline is 1.8724 milliseconds.

c. In the 3rd part of the question with T=12, we have to find the maximum value of C for which all the processes are guaranteed to meet their deadlines.

A set of independent periodic tasks schedule by the RMS algorithm will always meet the deadline for all tasks if

T1/C1 + T2/C2 +…………….TN/CN <= n(21/n - 1) ---------(2)

|  |  |  |
| --- | --- | --- |
| Process | Execution Time (Ti) | Ci (ms) |
| P1 | 12 | 50/C |
| P2 | 6 | 10/C |
| P3 | 4 | 40/C |
| P4 | 3 | 100/C |

In the equation (2); the value of T1, T2, T3 and T4 will be 12, 6, 4 and 3. Putting these values in equation (2), we have:

12/ (50/C) + 6/ (10/C) + 4/ (40/C) + 3/ (100/C) <= 0.7568

By calculating we have:

C(97/100) <= .7568

C <= .7802

**Answer 2:**

(a) It is given in the question that consider a single CPU-based hard real time system on which P processes are running under the Rate Monotic Scheduling discipline. The period for each process is x times its execution time.

We have to find the maximum value of P for which all the processes are guaranteed to meet their deadlines for x=1,2,3 and 4.

No. of Processors = 1

No. of Processes = P

Execution Time for Process Pi = Ti

Period (Pi) for each process is: Ci= x \* Ti

A set of independent periodic tasks schedule by the RMS algorithm will always meet the deadline for all tasks if

T1/C1 + T2/C2 +…………….TN/CN <= n(21/n - 1) ---------(1)

Putting the value of C1, C2, and CN in equation (1), we have:

T1/C1 + T2/C2 + T3/C3 + ….. TN/x\*TN <= n(21/n - 1)

T1/x\*T1 + T2/x\*T2 + T3/x\*T3 + ….. TN/x\*TN <= n(21/n - 1) -----------(2)

Putting the value of x=1, we have:

T1/T1 + T2/T2 + T3/T3 + ….. TN/TN <= n(21/n - 1)

1+1+1+1+……+1 <= n(21/n – 1)

1\*n <= n(21/n – 1) ------- (3)

Putting n=1in equation (3), we have:

1 <= 1 Inequality holds in this case

Putting n=2, in equation (3) we have:

1\*2 <= 2(21/2 - 1)

2<=0.82842 Inequality does not holds in this case

Putting the value of x=2 in equation (2), we have:

T1/x\*T1 + T2/x\*T2 + T3/x\*T3 + ….. TN/x\*TN <= n(21/n - 1)

T1/2\*T1 + T2/2\*T2 + T3/2\*T3 + ….. TN/2\*TN <= n(21/n - 1)

1/2 + 1/2 + 1/2 + ……….+ 1/2 <= n(21/n - 1)

n \* (1/2) <= n(21/n - 1) ----------------(4)

Putting n=1in equation (4), we have:

1\*(1/2) <= 1(2– 1)

0.5<= 0.82842 Inequality holds in this case

Putting n=2 in equation (4), we have:

2\*(1/2) <= 2(21/2 – 1)

1<=0.82842 Inequality does not holds in this case

Putting the value of x=3 in equation (2), we have:

T1/3\*T1 + T2/3\*T2 + T3/3\*T3 + ….. TN/3\*TN <= n(21/n - 1)

1/3+1/3+……+1/3 <= n(21/n - 1)

n\*(1/3) <= n(21/n - 1) -----------------(5)

Putting n=1in equation (5), we have:

1\*(1/3) <= 1(21/1-1)

1/3 <= 1

0.333 <=1 Inequality holds in this case

Putting n=2 in equation (5), we have:

2\*(1/3) <= 2(21/2 – 1)

0.6667 <= 0.82842 Inequality holds in this case

Putting the value of x=4 in equation (2), we have:

T1/4\*T1 + T2/4\*T2 + T3/4\*T3 + ….. TN/4\*TN <= n(21/n - 1)

n\*(1/4) <= n(21/n - 1) -------------(6)

Putting n=1 in equation (6), we have:

1\*(1/4) <= 1(2-1)

0.25 <= 1 Inequality holds in this case

Putting n=2 in equation (6), we have:

2\*(1/4) <= 2(21/2 – 1)

2\*(0.25) <= 0.82842

0.50 <= 0.82842 Inequality holds in this case

Putting n=3 in equation (6), we have:

3\*(1/4) <= 3(21/3 – 1)

0.75 <= 0.7797 Inequality holds in this case

Putting n=4 in equation (6), we have:

4\*(1/4) <= 4(21/4 - 1)

1 <= 0.7568 Inequality does not holds in this case

(b) In the 2nd part of the question, it is given that we have to consider the same system as in 2(a) but the processor speed is half of that used in 2(a). We have to find the maximum value of P for x=2. Everything will be same as in the 1st part, only the value of Ci will change.

No. of Processors = 1

No. of Processes = P

Execution Time for each process Pi = 2Ti

Period (Pi) for each process is: Ci= x \*2 Ti

Again, a set of independent periodic tasks schedule by the RMS algorithm will always meet the deadline for all tasks if

T1/C1 + T2/C2 +…………….TN/CN <= n(21/n - 1) ---------(a)

Putting the values in equation (a), we have :

2T1/x\*2T1 + 2T2/x\*2T2 +…………….+2TN/x\*2TN <= n(21/n - 1)

1/x + 1/x+…….+1/x <= n(21/n - 1) -------------------(b)

n/x <= n(21/n - 1) ----------------------(c)

Putting the value of x=2 in equation (c), we have:

n/2 <= n(21/n - 1) ----------------------(d)

Putting the value of n=1 in equation (d), we have:

1/2 <= 1(2-1)

1/2 <= 1 Inequality does not holds in this case

Putting the value of n=2 in equation (d), we have:

1 <=2(21/2 - 1)

1 <= 0.82842

Maximum value of **P is 1** for x=2

**Answer 3:**

It is given in the question that consider a system having 5 processes:

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

Ordering the processes on the basis of priority, we have:

**1st Processor**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Period(ms) | Execution Time (ms) |
| P1 | 1 | 50 | 10 |
| P4 | 2 | 60 | 30 |
| P3 | 3 | 75 | 25 |
| P2 | 4 | 80 | 15 |
| P5 | 5 | 140 | 35 |

For RMS to meet deadlines on 1st processor having k processes, we have:

T1/C1 + T2/C2 +T3/C3+T4/C4+T5/C5 <= k(21/k - 1)

10/50 + 15/80 + 25/75 + 30/60 + 35/140 <= k(21/k - 1)

1.47083 <= k(21/k - 1) -------------------------------------------------(a)

Putting the value of k=5 in equation (a), we have:

1.47083 <= 5(21/5 - 1)

1.47083 <= 0.74349 Inequality does not holds in this case

For 4 processes, putting the value of k=4 in R.H.S and subtracting T5/C5 =35/140 in L.H.S in equation (a), we have:

1.47083 – (35/140) <= 4(21/4 – 1)

1.22083<= 0.756821 Inequality does not holds in this case

For 3 processes, putting the value of k=3 in R.H.S and subtracting the sum of T2/C2 = 15/80 and T5/C5 =35/140 in L.H.S in equation (a), we have:

1.47083 – (35/140 + 15/80) <= 3(21/3 – 1)

1.03333<= 0.77976 Inequality does not holds in this case

For 2 processes, putting the value of k=2 in R.H.S and subtracting the sums of T2/C2 = 15/80, T5/C5 =35/140 and T3/C3 = 25/75 in L.H.S in equation (a), we have:

1.47083 - (15/80 + 35/140 + 25/75) <= 2(21/2 - 1)

0.6999 <= 0.82842 Inequality holds in this case

**2nd Processor**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Period(ms) | Execution Time (ms) |
| P3 | 3 | 75 | 50 |
| P2 | 4 | 80 | 30 |
| P5 | 5 | 140 | 70 |

For RMS to meet deadlines on 2nd processor having m processes, we have:

T3/C3 + T2/C2 +T5/C5 <= m (21/m - 1) --------------------------(b)

Putting the value of m=3 in equation (b), we have:

50/75 + 30/80 + 70/140 <= 3(21/3 - 1)

1.5416 <= 0.77976 Inequality does not holds in this case

Putting the value of m=2 in R.H.S and subtracting T5/C5=70/140 from L.H.S in equation (b), we have:

1.5416 - (70/140) <= 2 (21/2 - 1)

1.0416 <= 0.82842 Inequality does not holds in this case

Putting the value of m=1 in R.H.S and subtracting the sum of 70/140 and 30/80 from L.H.S in equation (b), we have:

1.5416 – (70/140+30/80) <= 1(2-1)

0.6666 <= 1 Inequality holds in this case

The value of m=1 signify that only 1 process will fit in 2nd processor.

**3rd Processor**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Period(ms) | Execution Time (ms) |
| P2 | 4 | 80 | 45 |
| P5 | 5 | 140 | 105 |

For RMS to meet deadlines on 3rd processor having s processes, we have:

T2/C2 +T5/C5 <= s (21/s - 1) ---------------------------(c)

45/80 + 105/140 <= 2 (21/2 - 1)

1.3125 <= 0.82842 Inequality does not holds in this case

Putting the value of s=1 in equation (c) and subtracting 105/140 in equation (c), we have:

T2/C2 <= 1 (21/1 - 1)

45/80 <= 1

0.5625 <=1 Inequality holds in this case

The value of s=1 signify that only 1 process will fit in 3rd processor.

**4th  Processor**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Period(ms) | Execution Time (ms) |
| P5 | 5 | 140 | 105 |

For RMS to meet deadlines on 4th processor having r processes, we have:

T5/C5 <= r (21/r - 1) ---------------------------(d)

105/140 <= 1

0.75 <= 1 Inequality holds in this case

The value of r=1 signify that only 1 process will fit in 4th processor.

Finally, 4 CPUs are needed such that all processes are guaranteed to meet their deadlines when RMS is used on each CPU.

**Answer 4:**

Yes, in some cases work-conserving schedule can be the optimal schedule that produces the highest speed up for some concurrent applications running on a non- preemptive environment.

Let us consider the following 2-level precedence graph representation of a system that has 8 processes and 3 processors.

T1/3 T8/9

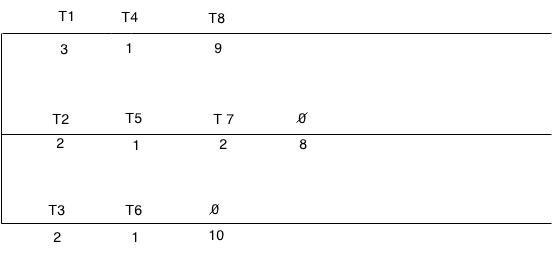
T2/2

T4/1

T3/2 T5/1

T6/1 T7/2

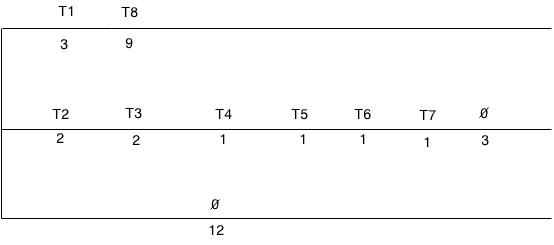
First, we are considering the work-conserving schedule and calculate w’ with the gain chart.



Here w’=13 and total idle time is 18

.

Now, in the case of non-work conserving schedule, we are forcing processor 3 to be in the idle state for the entire time.



We can above that w’=12 and total idle time is 15 units so we have presented an example where initial system holds.

**Answer-5**

App1 takes T unit of time when executed on a single processor

So, Tapp1(1)=T

App1 takes t1(N) units of time when executed on N processor

Tapp1(N)=t1(N)

So speedup for app1 is Sapp1(N)=T/t1(N)------------------------------------(1)

App2 takes T unit of time when executed on a single processor

So, Tapp2(1)=T

App2 takes t2(N) units of time when executed on N processor

Tapp2(N)=t2(N)

o speedup for app2 is Sapp2(N)=T/t2(N)----------------------------(2)

We know that , if the average parallelism A is known , following is hold

NA/N+a-1<=S(N)<=min(N,A)

app1 and app 2have same average parallelism they will have the same upper bound

so Sapp1(N)<=min(N,A) ----------------------------------------(3)

and

Sapp2(N)<=min(N,A) ----------------------------------------(4)

using eq 3 and 4

Sapp1(N)/ Sapp2(N)<=1

we can express the inequality in the following way (using eq 1 and 2 )

(T/t1(N)) /( T/t2(N)) <=1

or t2(N)/t1(N) <= 1