Aayahna Herbert Dr. Adam Hoover ECE 4680 23 April 2020

Lab 8: Real Time Scheduling Using RMA

After creating the max blocking table and writing a program to perform RMA with overhead and blocking, it was found that the theorem passes, and the system is schedulable. The table of the data used (the max blocking table), the values of (k,l) for which the theorem passes, and the code written is attached below.

Compute attitude data 1.30 1.055 0.20 0.30 0.00 2.00 3.00 3.33 Compute evelocity data 4.70 40.96 0.20 0.30 0.00 3.00 3.00 3.33 Compose attitude message 9.00 61.44 0.30 3.00 6.00 3.00 9.33 Display data 22.00 100.00 0.30 0.20 6.00 3.00 9.20 Compose navigation message 38.30 165.00 0.20 6.00 2.00 3.00 5.20 Run-time Built-in Test (BIT) 10.00 285.00 0.20 2.00 1.00 3.00 5.20 Compute position data 3.00 350.00 0.20 0.00 2.00 3.00 0.00 2.00										
Compute velocity data	Task	Execution Time (R) [ms]	Period (T) [ms]	Time Using R1 [ms]	Max Blocking [ms]	Time Using R2 [ms]	Max Blocking [ms]		Max Blocking [ms]	Total Max Blocking [ms]
Compose attitude message										3.30
Display data 23.00 100.00 0.30 0.20 6.00 3.00 5.20				0.20	0.30			3.00	3.00	
Compose navigation message 38.30 165.00 0.20 0.20 0.20 1.00 3.00 5.27	Compose attitude message	9.00	61.44		0.30	3.00	6.00		3.00	9.30
Resource #1 (R1)	Display data	23.00	100.00	0.30	0.20				3.00	9.20
Compute position data 3.00 350.00 0.20 0.00 0.	Compose navigation message	38.30	165.00		0.20	6.00			3.00	5.20
Resource #1 (R1): Result Table Usage - Calculating Mus Blocking Task Time Using Resource Max Blocking (Plash-Through) Max Blocking (Plash-Through	Run-time Built-In Test (BIT)	10.00	285.00		0.20		2.00	1.00	3.00	5.20
Resource #1 (R1): Result Table Usage - Calculating Max Blocking Task Time Using Resource [ms] Max Blocking (Direct) Max Blocking (Pash-Through) Max Blocking Campute velocity data 0.20 0.3	Compute position data	3.00	350.00	0.20	0.00		2.00	3.00	0.00	2.00
Task	Compose test message	2.00	700.00		0.00	2.00	0.00		0.00	0.00
Task										
Compute attitude data										
Compute velocity data 0.20 0.30		Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking					
Compose attitude message	Compute attitude data	0.20	0.30	0.00	0.30					
Display data	Compute velocity data	0.20	0.30	0.30	0.30					
Compose navigation message	Compose attitude message		0.00	0.30	0.30					
Run-time Built-In Test (BIT)	Display data	0.30	0.20	0.20	0.20					
Compute position data	Compose navigation message		0.00	0.20	0.20					
Resource #2 (R2): I/O Channel Usage - Calculating Max Blocking Task Time Using Resource ms Max Blocking (Direct) Max Block	Run-time Built-In Test (BIT)		0.00	0.20	0.20					
Task	Compute position data	0.20	0.00	0.00	0.00					
Task	Compose test message		0.00	0.00	0.00					
Task										
Compute attitude data		Resource #2 (R2): I/O Channel Usage - Calculating Max Blocking								
Compute velocity data	Task	Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking					
Compose attitude message 3.00 6.00 0.00 6.00 6.00	Compute attitude data		0.00	0.00	0.00					
Display data Disp	Compute velocity data		0.00	0.00	0.00					
Compose navigation message	Compose attitude message	3.00	6.00	0.00	6.00					
Run-time Built-in Test (BIT)	Display data		0.00	6.00	6.00					
Compuse position data 0.00 2.00 2.00 0.00	Compose navigation message	6.00	2.00	2.00	2.00					
Compose test message 2.00 0.00	Run-time Built-In Test (BIT)		0.00	2.00	2.00	i				
Resource #3 (R3): Disk Usage - Calculating Max Blocking Task Time Using Resource [ms] Max Blocking (Dush-Through) Max Blocking Compute stituted data 2.00 3.00 0.00 3.00 3.00 0.00 3.00 0.00	Compute position data		0.00	2.00	2.00	i				
Task Time Using Resource [ms] Max Blocking (Push-Through) Max Blocking Dush-Through	Compose test message	2.00	0.00	0.00	0.00	i				
Task Time Using Resource [ms] Max Blocking (Push-Through) Max Blocking Dush-Through										
Compute attitude data 2.00 3.00 0.00 Compute velocity data 3.00 3.00 3.00 Compose attitude message 0.00 3.00 3.00 Display data 0.00 3.00 3.00 Compose navigation message 0.00 3.00 3.00 Run-time Built-in Test (BIT) 1.00 3.00 3.00 Compute position data 3.00 0.00 0.00										
Compute velocity data 3.00 3.00 3.00 Compose attitude message 0.00 3.00 3.00 Display data 0.00 3.00 3.00 Compose navigation message 0.00 3.00 3.00 Run-time Built-In Test (BIT) 1.00 3.00 3.00 Compute position data 3.00 0.00 0.00	Task	Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking	ĺ				
Compose attitude message 0.00 3.00 3.00 Display data 0.00 3.00 3.00 Compose navigation message 0.00 3.00 3.00 Run-time Built-in Test (BIT) 1.00 3.00 3.00 Compute position data 3.00 0.00 0.00 Compute position data 3.00 0.00 0.00	Compute attitude data	2.00	3.00	0.00	3.00	i				
Compose attitude message 0.00 3.00 3.00 Display data 0.00 3.00 3.00 Compose navigation message 0.00 3.00 3.00 Run-time Built-in Test (BIT) 1.00 3.00 3.00 Compute position data 3.00 0.00 0.00 Compute position data 3.00 0.00 0.00		3.00	3.00			İ				
Display data			0.00	3.00		İ				
Compose navigation message 0.00 3.00 3.00 Run-time Bull+in Tex (BIT) 1.00 3.00 3.00 Compute position data 3.00 0.00 0.00	Display data		0.00	3.00	3.00	ĺ				
Run-time Built-in Test (BIT) 1.00 3.00 3.00 3.00 3.00 Compute position data 3.00 0.00 0.00 0.00			0.00	3.00	3.00	İ				
		1.00	3.00	3.00	3.00	İ				
	Compute position data	3.00	0.00	0.00	0.00	İ				
			0.00			İ				

Table 1: Max blocking table of data used

		schedulable! and 1 values collected:
i 1	k 1	1 1
2	1	1
3	1	3
4	1	7
5	2	4
6	6	1
7	4	3
8	1	43

Figure 1: Values of (k,l) for which the theorem passes

```
Aayahna Herbert
       Dr. Adam Hoover
       ECE 4680, Spring 20
       23 April 2020
       Lab 8: Real Time Scheduling Using RMA
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define VERY_SMALL 0.00000001
#define 0 0.153
 int main(void)
    int i, j, k, l, m;
    int schedulable;
    int kvalues[8];
    int lvalues[8];
    double sum;
    double R[8] = \{1.30, 4.70, 9.00, 23.00, 38.30, 10.00, 3.00, 2.00\};
    double T[8] = {10.56, 40.96, 61.44, 100.00, 165.00, 285.00, 350.00, 700.00};
    double B[8] = \{3.30, 3.30, 9.30, 9.20, 5.20, 5.20, 2.00, 0.00\};
    for (i = 0; i < 8; i++)
        schedulable = 0;
         for (k = 0; (k \le i) \&\& (schedulable == 0); k++)
           for (1 = 0; (1 \le floor(T[i]/T[k])) && (schedulable == 0); 1++)
             sum = 0.0;
             for (j = 0; j \le (i-1); j++)
                 sum += ((R[j] + O)*ceil(((double)l*T[k])/T[j]) - VERY SMALL));
             sum += R[i] + B[i];
             if (sum \le ((double)1*T[k]))
                  schedulable = 1;
                 kvalues[i] = k;
                 lvalues[i] = 1;
           }
        if (schedulable == 0)
             printf("\nThis system is not schedulable\n");
printf("Now printing k and l values collected:\n");
printf("\ni\tk\tl\n");
for (m = 0; m <= i; m++)</pre>
             {
                 printf("%d\t%d\n\n", m+1, kvalues[m]+1, lvalues[m]);
             }
             printf("\nExiting RMA program\n");
```

```
return 0;
}

printf("\nThis system is schedulable!\n");
printf("Now printing k and l values collected:\n");
printf("\ni\tk\tl\n");
for (m = 0; m < i; m++)
{
    printf("%d\t%d\t%d\n\n", m+1, kvalues[m]+1, lvalues[m]);
}

return 1;
}</pre>
```