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.

Task	Execution Time (R) [ms]	Period (T) [ms]	Time Using R1 [ms]	Max Blocking [ms]	Time Using R2 [ms]	Max Blocking [ms]	Time Using R3 [ms]	Max Blocking [ms]	Total Max Blocking [ms]
Compute attitude data	1.30	10.56	0.20	0.30		0.00	2.00	3.00	3.30
Compute velocity data	4.70	40.96	0.20	0.30		0.00	3.00	3.00	3.30
Compose attitude message	9.00	61.44		0.30	3.00	6.00		3.00	9.30
Display data	23.00	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
Compose test message	2.00	700.00		0.00	2.00	0.00		0.00	0.00

Resource #1 (R1): Result Table Usage - Calculating Max Blocking				
Task	Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking
Compute attitude data	0.20	0.30	0.00	0.30
Compute velocity data	0.20	0.30	0.30	0.30
Compose attitude message		0.00	0.30	0.30
Display data	0.30	0.20	0.20	0.20
Compose navigation message		0.00	0.20	0.20
Run-time Built-In Test (BIT)		0.00	0.20	0.20
Compute position data	0.20	0.00	0.00	0.00
Compose test message		0.00	0.00	0.00

Resource #2 (R2): I/O Channel Usage - Calculating Max Blocking				
Task	Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking
Compute attitude data		0.00	0.00	0.00
Compute velocity data		0.00	0.00	0.00
Compose attitude message	3.00	6.00	0.00	6.00
Display data		0.00	6.00	6.00
Compose navigation message	6.00	2.00	2.00	2.00
Run-time Built-In Test (BIT)		0.00	2.00	2.00
Compute position data		0.00	2.00	2.00
Compose test message	2.00	0.00	0.00	0.00

Resource #3 (R3): Disk Usage - Calculating Max Blocking				
Task	Time Using Resource [ms]	Max Blocking (Direct)	Max Blocking (Push-Through)	Max Blocking
Compute attitude data	2.00	3.00	0.00	3.00
Compute velocity data	3.00	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	3.00
Compute position data	3.00	0.00	0.00	0.00
Compose test message		0.00	0.00	0.00

Table 1: Max blocking table of data used

```
This system is schedulable!
Now printing k and l values collected:

i       k       l
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 0.000000001
#define O 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 <= i) && (schedulable == 0); k++)
        {
          for (l = 0; (l <= floor(T[i]/T[k])) && (schedulable == 0); l++)
          {
            sum = 0.0;

            for (j = 0; j <= (i-1); j++)
            {
                sum += ((R[j] + O)*ceil((((double)l*T[k])/T[j]) - VERY_SMALL));
            }

            sum += R[i] + B[i];

            if (sum <= ((double)l*T[k]))
            {
                schedulable = 1;
                kvalues[i] = k;
                lvalues[i] = l;
            }
          }
        }

        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++)
            {
                printf("%d\t%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;
}
```