Real Time Operating Systems  
Project

|  |  |
| --- | --- |
| Project number | 5 |
| Name | Sean Adin |
| Student ID | 259081 |
| Classes hours | Friday,8:45-11:00 |
| Date | 21.05.23 |

* ***Task 1-completed:***

**Description**:  
I wrote a task which utilizes the periodic interrupt and prints its value continuously.

**Problems**:  
Mostly editing issues to make the output look nicer.

**Testing**:

|  |  |
| --- | --- |
| **What is the expected result** | Periodic interrupt value of “%llu” |
| **How is the program tested** | Executing the program multiple times to see that the values are reasonable, and it works. |
| **Results** | Periodic interrupt value printed. |

**Conclusions**:  
The task was quite simple to do, but it required familiarizing with the built in functions of the interrupt.

* ***Task 2-completed:***

**Description**:  
The program is regarding the concurrency problem. I understood it as like synchronization issue with threads. I created a global variable, and the threads are synchronized using mutexes.

**Problems**:  
I had to read about the concurrency problem first.

**Testing**:

|  |  |
| --- | --- |
| **What is the expected result** | Having a global variable increased to some value. |
| **How is the program tested** | Executing the program multiple times and changing the value of the expected result for the global variable. |
| **Results** | Global variable was changed in a synchronous way. |

**Conclusions**:  
The program was a simple representation of the concurrency issue. Such a task was already done, but this time it required being familiar with the mutex library of the Xenomai API.

* ***Task 3-completed:***

**Description**:  
The program is performing the monte-carlo estimation based on queues. Unlike the previous task list, it had to be done with the message queue services of Xenomai.

**Problems**:  
No significant issues.

**Testing**:

|  |  |
| --- | --- |
| **What is the expected result** | The result should be approximately close to the constant value of Pi which is around 3.141592 |
| **How is the program tested** | Multiple test cases in each one we have different number of samples. |
| **Results** | The results were close to the expected values. |

**Conclusions**:  
The monte-carlo estimation was already done in previous project therefore understanding and implementing it was not really complicated.   
This time, as compared to the other tasks in this list, it required a good understanding of the Xenomai queue services.

* ***Task 4-completed:***

**Description**:  
The program is a Xenomai program to solve the readers and writers issue using the Xenomai buffer services. The buffer gets written into from an input text file, and then the buffer’s content is written into an output text file.

**Problems**:  
No significant issues.

**Testing**:

|  |  |
| --- | --- |
| **What is the expected result** | Output file created which contains a similar text to the input file. |
| **How is the program tested** | Executing the program multiple times while changing the input file text, to see if the program writes the same thing to the output text. |
| **Results** | Two similar files. |

**Conclusions**:  
In this program I had to familiarize myself with Xenomai’s buffer services to complete it. The algorithm was already done in the previous lists, therefore it was doable and mostly required adjustments to the buffer service API.