

Programming-Assignment-3 Report

Gustavo Hammerschmidt*
CS433 - Assignment 3

March 14 2020

1 Name and Description of the submitted files:

1.1 PCB.cpp

This class represents a process; it has all the necessary attributes for this project and methods to operate on them. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.2 LinkedList.cpp

This class represents a list of processes; it is used to as a base class by all of the Scheduling Algorithms implemented, it provides methods to all operations made by all of the algorithms(some of them overwrite these methods to implement a different procedure). It is worth mention that the insert and remove methods of this LinkedList are named push and pop respectively: when reading the documentation, understand that the push means a procedure will push the process into the LinkedList according to its defined procedure. I decide to name them this way because, initially, I named this class as ReadyQueue, but I've read the prog3.pdf file and, in it, it was mentioned that I should mind the difference between both lists. The final conclusion is that I used a LinkedList to manage the processes and the insert function is named push, also I did not make use of the pop function in any procedure in these project, so, there is not much to say about it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.3 FCFS.cpp

This class implements the LinkedList class and its functionalities. Any detail that is not explicitly defined here is well-documented on the class files, please

*hamme032@cougars.csusm.edu

check them for more information.

1.4 SJF.cpp

This class implements the LinkedList class and its functionalities. It overwrites the push method to insert processes according to their burst time. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.5 PriorityScheduling.cpp

This class implements the LinkedList class and its functionalities. It overwrites the push method to insert processes according to their priority. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.6 RoundRobin.cpp

This class implements the LinkedList class and its functionalities. It overwrites the push method to insert slices of the processes, based on its time quantum. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.7 PriorityAndRR.cpp

This class implements the LinkedList class and its functionalities. It overwrites the push method to insert slices of the processes, based on its time quantum and their priority. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.8 MainFCFS.cpp

This class implements the FCFS class and its functionalities. This class is used as a execution class to the FCFS Scheduling Algorithm. It, basically, orders the execution of it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.9 MainSJF.cpp

This class implements the SJF class and its functionalities. This class is used as a execution class to the SJF Scheduling Algorithm. It, basically, orders the execution of it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.10 MainPriorityScheduling.cpp

This class implements the PriorityScheduling class and its functionalities. This class is used as a execution class to the Priority Scheduling Algorithm. It, basically, orders the execution of it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.11 MainRoundRobin.cpp

This class implements the RoundRobin class and its functionalities. This class is used as a execution class to the Round-Robin Scheduling Algorithm. It, basically, orders the execution of it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.12 MainPrAndRR.cpp

This class implements the PriorityAndRR class and its functionalities. This class is used as a execution class to the Priority-with-Round-Robin Scheduling Algorithm. It, basically, orders the execution of it. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

Observation all of the above mentioned classes have also a header file.

1.13 m_fcfs.cpp

This class is the main class for the FCFS Scheduling Algorithm. It takes the input and passes it to the Algorithm Execution class. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.14 m_sjf.cpp

This class is the main class for the SJF Scheduling Algorithm. It takes the input and passes it to the Algorithm Execution class. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.15 m_priority.cpp

This class is the main class for the Priority Scheduling Algorithm. It takes the input and passes it to the Algorithm Execution class. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.16 m_rr.cpp

This class is the main class for the Round-Robin Scheduling Algorithm. It takes the input and passes it to the Algorithm Execution class. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.17 m_priority_and_rr.cpp

This class is the main class for the Priority-with-Round-Robin Scheduling Algorithm. It takes the input and passes it to the Algorithm Execution class. Any detail that is not explicitly defined here is well-documented on the class files, please check them for more information.

1.18 Makefile

To make this Makefile, I used as a base the makefile provided assignment 3 example code on the server. I've made some changes on it: defined the classes that would make each executable. In general, the Makefile is pretty similar to the Makefile provided.

1.19 input.txt

This .txt file was used throughout the development to test the algorithms functions and check the final result. It doesn't have the same processes with the same values, used in the example code to test it. So, basically, I did not test my algorithms with the scheduling.txt file, and I trust I did all the calculations rightly. Any file that follows the input structure defined on the scheduling.txt or the input.txt can be used on the algorithms.

1.20 README.txt

This file was made just to inform you that a folder in the submission folder does not have anything to do with the submission. I've made a mistake when transferring the files. The folder does not interfere in anything on the project execution, so, please, ignore the folder.

2 Compilation of the program

To compile the program, use the Makefile to create the object classes of each .cpp class. I used the makefile file available on the example code on the server, making small changes to it in order to compile the program and to be able to

execute it. It was compiled and executed on linux. I solely recommend executing it in Linux.

3 Results of the program

I believe that I've made all the requirements that should be done, including the additional or the extra credit functionality(a priority-with-round-robin scheduling algorithm).The program was first written on visual studio and then transposed to the unix virtual machine environment, with some adjustments made and some errors fixed. Some warnings are reported during the compilation of the .cpp files, they do not damage the code in any aspect. I'd say my executable files perform well and that they make the time measurements properly, though I did not test them with the scheduling.txt file because I trust my code and used another .txt file to test and calculate the algorithms throughout the development(this file is given on my assign3 submission folder, it is named as input.txt). I will also remind that any information not explicitly describe here is well-commented on the program's source code. This project may look extensive, due to amount of classes I've made; but I affirm that the effort was worth in the end and that classes serve a purpose on the project. I affirm that I have not made use of any means of code copy or even used other's code. I have not taken any reference on any procedure involved on my code, except that I took the example given in the Operating Systems ed. 10 book as a strong reference, which guided the development of my program with much emphasis, and that I used the assignment 3 example code on the server to understand how to pass input information to the main function on the kernel. I've executed the code both on a linux and a windows environment and the code performed well on both of them.