James Donaldson, Tyler Desharnais

CS433: Operating Systems 10:30 am

February 2, 2022

**Assignment 1 - Report**

**Submitted Files:**

* **PCB.cpp:** Source file that contains the functions for the PCB class
* **Driver.cpp:** A source file that contains the test functions
* **MaxHeap.cpp:** Source file that contains the main functions for the Data structures
* **ReadyQueue.cpp:** Source file to control the PCB nodes being added to the Heap

**How to Compile and Run the Program:**

**To compile the program, use command: g++ driver.cpp**

**To run the program, use command: ./a.out**

**Results and runtime of test 2:**

Our program was able to complete the test cases provided for the classes to run and all the functions seemed to work as expected. We conducted test to determine the average time output of the program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Time 1** | **Time 2** | **Time 3** | **Time 4** | **Time 5** |
| .45 sec | .48 sec | .47 sec | .48 sec | .43 sec |

Avg : .462 sec

**Features Implemented (by class):**

* **PCB:** The first functions that were established was the constructors and the getters and setters. We also used the enum to label the state of the PCB. We created a os to create a simple print out for the information on PCB’s. The last function created was the state helper to identify the states of the PCB.
* **MaxHeap:** The max heap class is the core data structure for our PCB program. The start of the program has the basic information such as size and capacity. After that we have the constructors to create your heap. Now we have the getter functions for the heap, along with the parent function and the left and right child.We also have the functions to build the heap and balance the heap as PCBs are being added. After all of the basic heap functions we have listed a insert PCB function to properly add PCBs to the heap.
* **ReadyQueue:** The ready queue function was created to control the communication between the pcb function and max heap. We can add and remove pcbs from the heap through this function and check the size, if empty, and display the heap through this source file.
* **Driver:** In this cpp file we have the main function that calls the two test functions to test the program and how it functions.

**Design and Implementation:**

We decided to use a heap function to control the PCBs that we are entering into the program. The struggles we came across the most was figuring out how to create the PCBs and to create the enum types to represent the states of the PCB’s. The Ready queue function helped us with the ability to control the heap better and made our code easier for us to read and understand. It took us a few days to determine what we wanted to for our data structure and how to properly connect the cpp files.

**Lessons Learned:**

This first program showed that we need to take the time to plan better, and we can’t rush into the coding portion of these assignments. This program was far more complicated than what we have had to deal with in previous classes. When learning new data structures, we need to take a day or two researching how this data structure works and how to implement it. We also had to review how the heap data structure worked to properly understand what our code was doing.

**References:**

We used the previous code that was written from CS 311 for the heap and the communication with the ready queue. Also had to use this website <https://www.cplusplus.com/reference/ctime/> to figure out the proper way to implement the time function.