Kyle Kolodziej

CS 5343 – Dr. Hakki Cankaya

Operating Systems Project – Part 1

June 22nd, 2021

Operating Systems Project Part 1

Data Structures

For the Process Control Board Queue, I decided to use a Singly Linked List. I decided on using a Singly Linked List as this data structure does a great job of efficiently fulfilling the functionality of the Process Control Board queue. In comparison to a Doubly Linked List, the Singly Linked List takes up less memory. For deleting a process, we will need to locate the process with a specific process ID. Similarly, we will need to locate where a process' priority falls within the queue for inserting. Having to locate these processes mitigates the advantages a Doubly Linked List has with inserting and deleting over a Singly Linked List.

I created a class called Process. Objects of type Process are created for each Process and added to the Process Control Board Queue. Each Process contains a process ID and a priority value. The process ID is required for the Process while the priority is optional. If not given a priority value, the Process will set its priority value to -1 which is used to indicate that a Process does not contain a priority. Since I decided to use a Singly Linked List, each Process object only

has a next pointer. This will be used for operations performed on the Process Control Block Queue such as inserting and deleting.

Additionally, I created a class called ProcessControlBlock. A Process Control Block can be initialized with or without a Process such that it could be an empty queue. Each Process Control Block object contains Process(es) that are connected by a Singly Linked List. The Process Control Board also contains a pointer for the head and tail of the queue. These pointers are useful in the functions that need to be performed on the queue.

Algorithms Used

For adding processes to the process control board queue, I created a function called addProcess(self, processID, priority=None). Processes are required to have a process ID, but the priority is optional. I decided the best way to accommodate this with providing full functionality for inserting processes was to make a function that requires a process ID but the priority is optional. This way, I can always use this function regardless of if the process contains a priority or not. If a process being added does not have a priority, the default position for this process to be added is at the end of the process control board queue. Additionally, if multiple processes have the same priority value, the ordering for those processes with the same priority will function as a FIFO queue. For example, the processes that were in the process control board queue before will be farther up front than the process that is just added in with the same priority.

Upon inserting a process, I first check whether the queue is empty. If the queue is empty, I can just initialize the head and tail of the queue with this process. Otherwise, I will take one of two paths for inserting the process. First, I check if a priority was passed in. If no priority value was passed in, then I just need to add the process at the end of the queue. To do this, I just update the current tail's next pointer to point to the new process being added. Then, I update the tail to be pointing to the process that was just added at the end of the queue. If a priority value was passed into the process and the queue is not empty, then I need to figure out where the new process falls in terms of priority. I start by checking if the head's priority value. If it does not have a priority or if the incoming process' priority value is lower (meaning it has a higher priority), the process being added becomes the new head of the queue. If this is not the case, then I iterate until either the incoming priority's value is lower than that "current" process' priority in the queue, the "current" process does not have a priority, or I reach the end of the queue. I insert the new process at this position, also updating the tail in the case that I reached the end of the queue.

For deleting a process, I first check if the Process Control Block Queue is empty. If it is empty, then I will print out a message to the user informing them of this. If this is not the case, then I determine whether I need to delete a specific process. When there is no process ID passed in, then I will just delete the process from the default position at the head. Otherwise, I need to search to find if this process exists in the queue. First, I check if the process ID matches with the head's process ID. Then, I will just delete the head, printing out this process' information to the user, and update the head to its next process that it is pointing to. For both the case that no process ID was passed in and the case that a process ID was passed in and

matched with the head's process ID, I check to see if the head was the only process in the queue. Deleting this process would mean that the queue is now empty. Then, I would update the tail to reflect this. When passing in a process ID to be deleted that does not match with the head's process ID, I then check whether the head is the only process. If it is the only process in the queue, then I inform the user that they are unable to delete that process as it does not exist. On the other hand, if more process(es) exist, then I set a "curr" pointer to the head's next process and a "prev" pointer to the head. While the process ID passed in to be deleted does not equal the process ID of "curr" and "curr" is not the last process in the queue, I iterate both "curr" and "prev" forward. When this while loop breaks, this either means that I have reached the process that needs to be deleted or the end of the queue. If I reached the process that should be deleted, I delete it by setting "prev's" next pointer to "curr's" next pointer and update the tail to be "prev" if "curr" was the last process in the queue. Otherwise, I output a message to the user that I was unable to delete a process with that process ID as there was no match.

For printing the process control board queue, I start by checking if the Process Control Board Queue is empty. If it is empty, I will output a message informing the user that the queue is empty. Otherwise, I will print the position (relative to in the queue), process ID, and priority of each process with an arrow pointed towards the next process in the queue. I iterate through all the processes by using a pointer to the head process called "curr". I continue to move "curr" forward until it reaches the end of the queue, printing out the information of each process along the way.

Execution Trace

```
-----------------------------------------------------------------------------------------------------

Welcome to Kyle Kolodziej's Operating System's Project #1: Process Control Board queue Manipulation!

-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
Would you like to...
    1) Add Process(es) via an Input File
    2) Add Process(es) via a process ID and priority from your input
    3) Delete a Process
    4) Print the Process Control Board
    5) Exit
-----------------------------------------------------------------------------------------------------

Please input your option (1-5):
```

Figure 1: The welcome message and menu options shown at the start of running the program.

```
-----------------------------------------------------------------------------------------------------

Please input your option (1-5): 4

Printing the Process Control Board queue...

Process Control Board queue is empty!

-----------------------------------------------------------------------------------------------------
```

Figure 2: Attempting to print the Process Control Board queue when it is empty.

```
-----------------------------------------------------------------------------------------------------

Please input your option (1-5): 3
Would you like to delete a specific Process? If not, will default to the Process at the start of the queue
    1) Yes
    2) No
Please enter your choice (1 or 2): 2

Deleting Process from the default position (the head)...

Error! Not able to delete a Process...the Process Control Board queue is empty!


-----------------------------------------------------------------------------------------------------
```

Figure 3: Attempting to delete a default process when the Process Control Board queue is empty.

```
-----------------------------------------------------------------------------------------------------

Please input your option (1-5): 1
Please enter the input file's name: processInputFile

Successfully added Processes from:  processInputFile.txt
-----------------------------------------------------------------------------------------------------
```

Figure 4: Adding in processes from an input file.

```
-----------------------------------------------------------------------------------------------

Please input your option (1-5): 4

Printing the Process Control Board queue...

[Position: 1, Process ID = 7, Priority = 400] -----> [Position: 2, Process ID = 1, Priority = 403] -----> [Position: 3, Process ID = 16, Priority = 563]

-----------------------------------------------------------------------------------------------
```

Figure 5: Printing the Process Control Board queue after reading in the input file.

```
-----------------------------------------------------------------------------------------------

Please input your option (1-5): 2
Please enter the Process ID: INSERTING_THIS_MANUALLY
Would you like to enter a priority for this process?
    1) Yes
    2) No
Please enter your choice (1 or 2): 1

Please enter the priority (integer >= 1): 401
-----------------------------------------------------------------------------------------------
```

Figure 6: Inserting a process from a user's manual input of process ID and priority.

```
-----------------------------------------------------------------------------------------------

Please input your option (1-5): 4

Printing the Process Control Board queue...

[Position: 1, Process ID = 7, Priority = 400] -----> [Position: 2, Process ID = INSERTING_THIS_MANUALLY, Priority = 401] -----> [Position: 3, Process ID = 1, Priority = 403] --
```

Figure 7: Printing the queue after the manually entered process from above is added to the queue.

```
-----------------------------------------------------------------------------------------------------------

Please input your option (1-5): 2
Please enter the Process ID: INSERTING_MANUALLY_WITHOUT_PRIORITY
Would you like to enter a priority for this process?
    1) Yes
    2) No
Please enter your choice (1 or 2): 2
-----------------------------------------------------------------------------------------------------------
```

Figure 8: Inserting a process from a user's manual input of a process ID, but without a priority.

```
-----> [Position: 51, Process ID = 30, Priority = 9943] -----> [Position: 52, Process ID = INSERTING_MANUALLY_WITHOUT_PRIORITY, Priority = None]
```

Figure 9: Printing the queue and showing the tail after inserting the process without a priority.

```
-----------------------------------------------------------------------------------------------------------

Please input your option (1-5): 3
Would you like to delete a specific Process? If not, will default to the Process at the start of the queue
    1) Yes
    2) No
Please enter your choice (1 or 2): 2

Deleting Process from the default position (the head)...

Process Deleted: [Position: 1, Process ID: 7, Priority: 400]


-----------------------------------------------------------------------------------------------------------
```

Figure 10: Deleting a process from the default position (the head) by not passing in a process ID.

```
-----------------------------------------------------------------------------------------------------------

Please input your option (1-5): 4

Printing the Process Control Board queue...

[Position: 1, Process ID = INSERTING_THIS_MANUALLY, Priority = 401] -----> [Position: 2, Process ID = 1, Priority = 403] -----> [Position: 3, Process ID = 16, Pri

-----------------------------------------------------------------------------------------------------------
```

Figure 11: Printing the queue after deleting the default process at the head.

```
--------------------------------------------------------------------------------------------------------

Please input your option (1-5): 3
Would you like to delete a specific Process? If not, will default to the Process at the start of the queue
    1) Yes
    2) No
Please enter your choice (1 or 2): 1
Please enter the Process ID for the Process to be deleted: 1

Deleting Process with process ID '1'...

Process Deleted: [Position: 2, Process ID: 1, Priority: 403]


--------------------------------------------------------------------------------------------------------
```

Figure 12: Deleting a specific process from the queue.

```
--------------------------------------------------------------------------------------------------------

Please input your option (1-5): 4

Printing the Process Control Board queue...

[Position: 1, Process ID = INSERTING_THIS_MANUALLY, Priority = 401] -----> [Position: 2, Process ID = 16, Priority = 563] -----> [Position: 3, Process ID = 11,

--------------------------------------------------------------------------------------------------------
```

Figure 13: Printing the queue after deleting the specific process from above.

```
--------------------------------------------------------------------------------------------------------

Please input your option (1-5): 3
Would you like to delete a specific Process? If not, will default to the Process at the start of the queue
    1) Yes
    2) No
Please enter your choice (1 or 2): 1
Please enter the Process ID for the Process to be deleted: THIS_PROCESS_DOES_NOT_EXIST

Deleting Process with process ID 'THIS_PROCESS_DOES_NOT_EXIST'...

Error! Process ID 'THIS_PROCESS_DOES_NOT_EXIST' does not exist in the Process Control Board queue!


--------------------------------------------------------------------------------------------------------
```

Figure 14: Attempting to delete a specific process by a process ID that does not exist.

```
-----------------------------------------------------------------------------------------------------
Would you like to...
    1) Add Process(es) via an Input File
    2) Add Process(es) via a process ID and priority from your input
    3) Delete a Process
    4) Print the Process Control Board
    5) Exit
-----------------------------------------------------------------------------------------------------

Please input your option (1-5): 5

-----------------------------------------------------------------------------------------------------

Thank you for using Kyle Kolodziej's Process Control Board Queue! Goodbye!

-----------------------------------------------------------------------------------------------------


Process finished with exit code 0
```

Figure 15: Exiting the program.

Programming Environment

I used my Windows HP Spectre x360 laptop for this project. This laptop has an Intel® Core™ i7-8550U CPU and 16.0 GB of RAM. It uses four cores and eight threads along with the one processor. This system utilizes a 64-bit operating system. I coded in Python using the application PyCharm.