Assignment 4

Operating Systems | Anthony Nguyen

round_robin.c – Source Code

I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 maybe assigned for copied work." - Anthony Nguyen | 104792283

Implementation Details and Explanation:

First what happens is that my program asks the user for the number of processes and it stores the user input into a variable. With that variable it creates 2 arrays of that size named:

burst_time – Each index corresponds to a process and the value corresponds to the burst time
track_time – This is a copy of the burst_time array and is used for calculations.

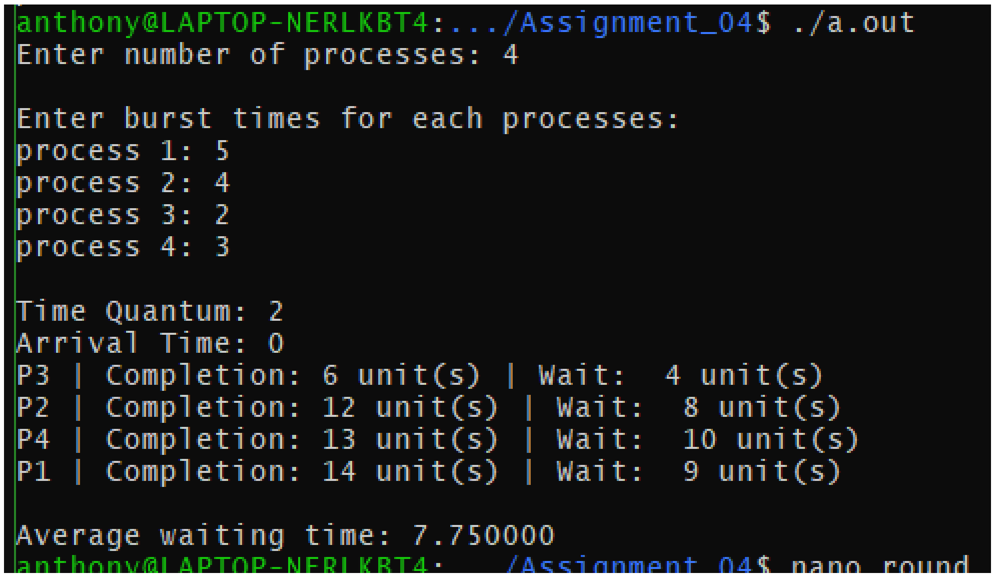
The two arrays are populated with burst times inputted by the user and my program starts doing the round robin calculations with the assumption that the time quantum is fixed to 2 and the arrival time is 0. Based on the pdf of this assignment the inputs were only the number of processes and burst times. Therefore, the quantum time was fixed at 2 using a macro, changing this to be user inputted is very simple and just 2 or 3 more lines of code.

After the user inputs, the program runs a simple round robin algorithm. This algorithm contains 2 loops. The outer loop is a while loop that checks if the number of processes left is 0, if its not 0 then it keeps looping. The inside for loop is used to loop through each process in order (kind of like a queue + when it pushes an item out o the queue it re adds it to the queue). There are three options that occur inside the inner for loop (imitating running one of the processes):

1. if the time left on that processes is more than the time quantum (2) it will subtract the time on the process by the time quantum to imitate the cpu working on that process for 2 units of time. After it subtracts the time on the process it will add that same quantum time to a variable that tracks the elapsed time.
2. If the time left on that process is less than or equal to the time quantum but more than 0 it will set the process time to 0 (meaning the process finishes its time). The actual process time will add that to the elapsed time variable (since you will not know it could be less than the quantum time) and decrements the variable that keeps track of how many processes are left. Also, given this info we can print the wait time of this process because this process completed. So, a simple calculation of taking the elapsed time variable and subtracting it to the burst time of that process will yield the wait time. Note: At this point the elapsed time = complete time = turnaround time (because arrival time = 0) and that is why this calculation works.
3. If the time left on the process is 0 then it will skip to the next process

After the above processes runs multiple times until the variable that keeps the number of process left equals 0, the outer while loop will exit, and all processes will be completed. Lasty, the average wait time is calculated by taking that elapsed time variable we were using and dividing it by the total number of processes.

Snapshots of Runs:

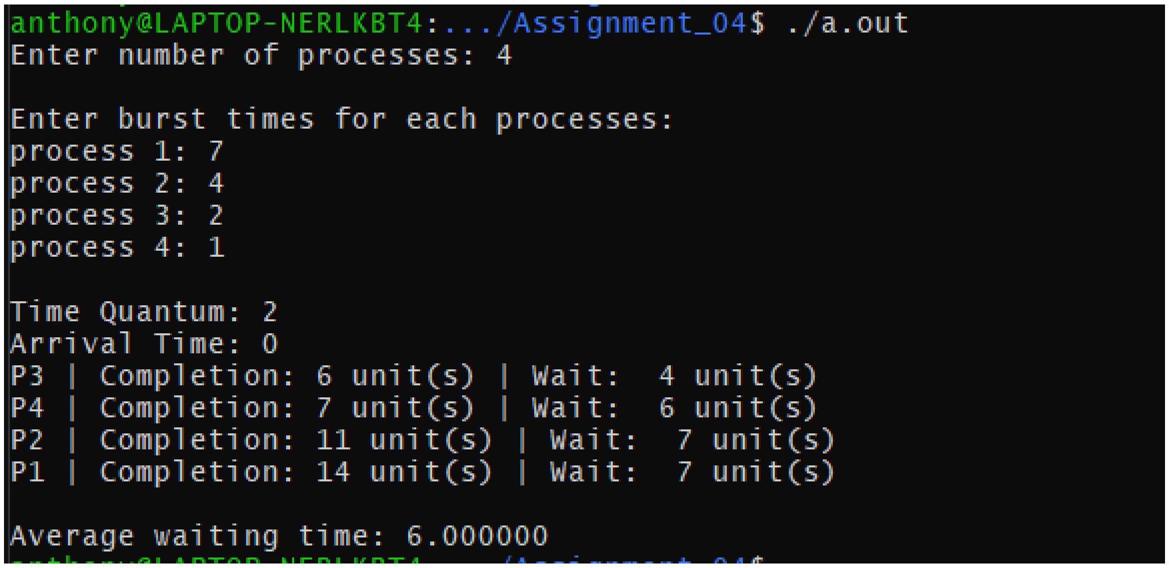


```
anthony@LAPTOP-NERLKBT4:.../Assignment_04$ ./a.out
Enter number of processes: 4

Enter burst times for each processes:
process 1: 5
process 2: 4
process 3: 2
process 4: 3

Time Quantum: 2
Arrival Time: 0
P3 | Completion: 6 unit(s) | Wait:  4 unit(s)
P2 | Completion: 12 unit(s) | Wait:  8 unit(s)
P4 | Completion: 13 unit(s) | Wait:  10 unit(s)
P1 | Completion: 14 unit(s) | Wait:  9 unit(s)

Average waiting time: 7.750000
anthony@LAPTOP-NERLKBT4:.../Assignment_04$ nano round
```



```
anthony@LAPTOP-NERLKBT4:.../Assignment_04$ ./a.out
Enter number of processes: 4

Enter burst times for each processes:
process 1: 7
process 2: 4
process 3: 2
process 4: 1

Time Quantum: 2
Arrival Time: 0
P3 | Completion: 6 unit(s) | Wait:  4 unit(s)
P4 | Completion: 7 unit(s) | Wait:  6 unit(s)
P2 | Completion: 11 unit(s) | Wait:  7 unit(s)
P1 | Completion: 14 unit(s) | Wait:  7 unit(s)

Average waiting time: 6.000000
anthony@LAPTOP-NERLKBT4:.../Assignment_04$
```