

## 1.General Information

In this Project, I learnt a real CPU scheduling algorithm in C language with Linux. How I did this is I used CFS algorithm. I implemented and learnt with this scheduling Project and from the course book;

For the time-sharing class, CFS can not directly prioritize. No separate running queues.

Keep track of how long each process lasts as a virtual runtime.

Virtual run time is advanced for a process with a lower priority (higher value), and less for a higher priority process.

For this reason, the lower your priority is higher.

Example: The actual running time of 200 ms will be more for a low priority operation, less for a high priority operation, and 0 for a good value.

Select the operation that has the lowest virtual working time.

Keep the operations executable in a red-black tree (balanced tree) according to the virtual run times.

The process is on the left.

Selection:  $O(1)$ ;

Insert a process into the tree:  $O(\log N)$

I practiced a simulation program in c, in extra I got familiar with executing files, practiced with writing and reading files by inputs and outputs. I interpreted results which is given before, learnt how to generate a random variate.

## 2. Environment and setup

The experiments were run in Linux, Ubuntu 17.04 that is up to date with 2 processes(2 CPU cores), with ASUS N61JQ-JX050V.

There are fixed set of inputs which are given by the instructor. I used those inputs by reading from text file. The input text file is "sample-workload-1" that should be in same directory with simulator.c :

*1 start 2000 prio 20*

*1 cpu 400*

*1 io 100*

*1 cpu 300*

*1 end*

*2 start 1000 prio 20*

*2 cpu 200*

*2 io 300*

*2 cpu 200*

*2 end*

### 3. Results and discussions

The program is beginning with Linked List implementation. I used Linked List instead of Red-black tree because I find it less complicated and much clear, understandable. I also used "time.h" for calculation and interpreting the results. Took running and waiting variables for interpreting and initialized them to NULL value which will be used later. After that I implement CFS algorithm which is requirement of the implementation. In this, it is running variable is checked for realizing thread's condition. Then I am taking thread with its id which has minimum vcruntime. In this, I used usleep() function because I wanted to suspend the execution for microsecond intervals otherwise the problem in my implementation occurs. So I found this solution and it works. And then given input text file is read by the program, with checking nullity. I used atoi for parsing to integer. I used flags for implementation, calculation. I used clock for time as I defined before with "time.h". I allocated memory using malloc() in Linked List implementation. I add node in linked list and assigned my calculations to it. Then I wrote them to output file. After that, in startOperationthread() function I let these calculations to be done by threads by pthread\_create() and pthread\_join() functions. As a result I am getting these results and writing them to file:

*1 20 2000 2457 457 351 196*

*2 20 1000 1349 349 118 196*

*Statistics:*

*Total Process:2*

*Average Turn Around Time(ms):403.200000*

*Average Wait Time(ms):234.500000*

*Average Response Time(ms):196.000000*

---

*Results are similar with outputs given by the instructor with some fixes:*

```
"1 start 0 prio 20
1 cpu 400
1 end
2 start 0 prio 20
2 cpu 400
2 end
```

output:

time: 0.0 pid: 1 started in cpu  
time: 0.0 pid: 1 started cpu burst  
time: 0.0 pid: 1 timer ticked  
time: 10.0 pid: 1 timer ticked  
time: 20.0 pid: 1 timer ticked  
time: 30.0 pid: 1 timer ticked  
time: 40.0 pid: 1 timer ticked  
time: 50.0 pid: 1 timer ticked  
time: 60.0 pid: 1 timer ticked  
time: 70.0 pid: 1 timer ticked  
time: 80.0 pid: 1 timer ticked  
time: 90.0 pid: 1 timer ticked  
time: 100.0 pid: 1 timer ticked  
time: 100.0 pid: 1 removed from cpu - timeslice expired - suspended  
time: 100.0 pid: 1 removed from cpu - suspended  
time: 100.0 pid: 2 started in cpu  
time: 100.0 pid: 2 started cpu burst  
time: 110.0 pid: 2 timer ticked  
time: 120.0 pid: 2 timer ticked  
time: 130.0 pid: 2 timer ticked  
time: 140.0 pid: 2 timer ticked  
time: 150.0 pid: 2 timer ticked  
time: 160.0 pid: 2 timer ticked  
time: 170.0 pid: 2 timer ticked  
time: 180.0 pid: 2 timer ticked  
time: 190.0 pid: 2 timer ticked  
time: 200.0 pid: 2 timer ticked  
time: 200.0 pid: 2 removed from cpu - timeslice expired - suspended  
time: 200.0 pid: 2 removed from cpu - suspended  
time: 200.0 pid: 1 started in cpu  
time: 210.0 pid: 1 timer ticked  
time: 220.0 pid: 1 timer ticked  
time: 230.0 pid: 1 timer ticked  
time: 240.0 pid: 1 timer ticked  
time: 250.0 pid: 1 timer ticked  
time: 260.0 pid: 1 timer ticked  
time: 270.0 pid: 1 timer ticked  
time: 280.0 pid: 1 timer ticked  
time: 290.0 pid: 1 timer ticked  
time: 300.0 pid: 1 timer ticked  
time: 300.0 pid: 1 removed from cpu - timeslice expired - suspended  
time: 300.0 pid: 1 removed from cpu - suspended  
time: 300.0 pid: 2 started in cpu  
time: 310.0 pid: 2 timer ticked  
time: 320.0 pid: 2 timer ticked  
time: 330.0 pid: 2 timer ticked  
time: 340.0 pid: 2 timer ticked  
time: 350.0 pid: 2 timer ticked  
time: 360.0 pid: 2 timer ticked  
time: 370.0 pid: 2 timer ticked  
time: 380.0 pid: 2 timer ticked  
time: 390.0 pid: 2 timer ticked  
time: 400.0 pid: 2 timer ticked  
time: 400.0 pid: 2 removed from cpu - timeslice expired - suspended  
time: 400.0 pid: 2 removed from cpu - suspended  
time: 400.0 pid: 1 started in cpu  
time: 410.0 pid: 1 timer ticked  
time: 420.0 pid: 1 timer ticked  
time: 430.0 pid: 1 timer ticked  
time: 440.0 pid: 1 timer ticked

time: 450.0 pid: 1 timer ticked  
time: 460.0 pid: 1 timer ticked  
time: 470.0 pid: 1 timer ticked  
time: 480.0 pid: 1 timer ticked  
time: 490.0 pid: 1 timer ticked  
time: 500.0 pid: 1 timer ticked  
time: 500.0 pid: 1 removed from cpu - timeslice expired - suspended  
time: 500.0 pid: 1 removed from cpu - suspended  
time: 500.0 pid: 2 started in cpu  
time: 510.0 pid: 2 timer ticked  
time: 520.0 pid: 2 timer ticked  
time: 530.0 pid: 2 timer ticked  
time: 540.0 pid: 2 timer ticked  
time: 550.0 pid: 2 timer ticked  
time: 560.0 pid: 2 timer ticked  
time: 570.0 pid: 2 timer ticked  
time: 580.0 pid: 2 timer ticked  
time: 590.0 pid: 2 timer ticked  
time: 600.0 pid: 2 timer ticked  
time: 600.0 pid: 2 removed from cpu - timeslice expired - suspended  
time: 600.0 pid: 2 removed from cpu - suspended  
time: 600.0 pid: 1 started in cpu  
time: 610.0 pid: 1 timer ticked  
time: 620.0 pid: 1 timer ticked  
time: 630.0 pid: 1 timer ticked  
time: 640.0 pid: 1 timer ticked  
time: 650.0 pid: 1 timer ticked  
time: 660.0 pid: 1 timer ticked  
time: 670.0 pid: 1 timer ticked  
time: 680.0 pid: 1 timer ticked  
time: 690.0 pid: 1 timer ticked  
time: 700.0 pid: 1 timer ticked  
time: 700.0 pid: 1 removed from cpu - finished  
time: 700.0 pid: 1 finished process  
time: 700.0 pid: 2 started in cpu  
time: 710.0 pid: 2 timer ticked  
time: 720.0 pid: 2 timer ticked  
time: 730.0 pid: 2 timer ticked  
time: 740.0 pid: 2 timer ticked  
time: 750.0 pid: 2 timer ticked  
time: 760.0 pid: 2 timer ticked  
time: 770.0 pid: 2 timer ticked  
time: 780.0 pid: 2 timer ticked  
time: 790.0 pid: 2 timer ticked  
time: 800.0 pid: 2 timer ticked  
time: 800.0 pid: 2 removed from cpu - finished  
time: 800.0 pid: 2 finished process

”

