Group Project 4: CPU scheduler

Antonio Hughes (015748783)
Sandra Chavez (016363540)
CECS 326 - Sec 04
Prof. Hailu Xu
12 - 06 - 2020

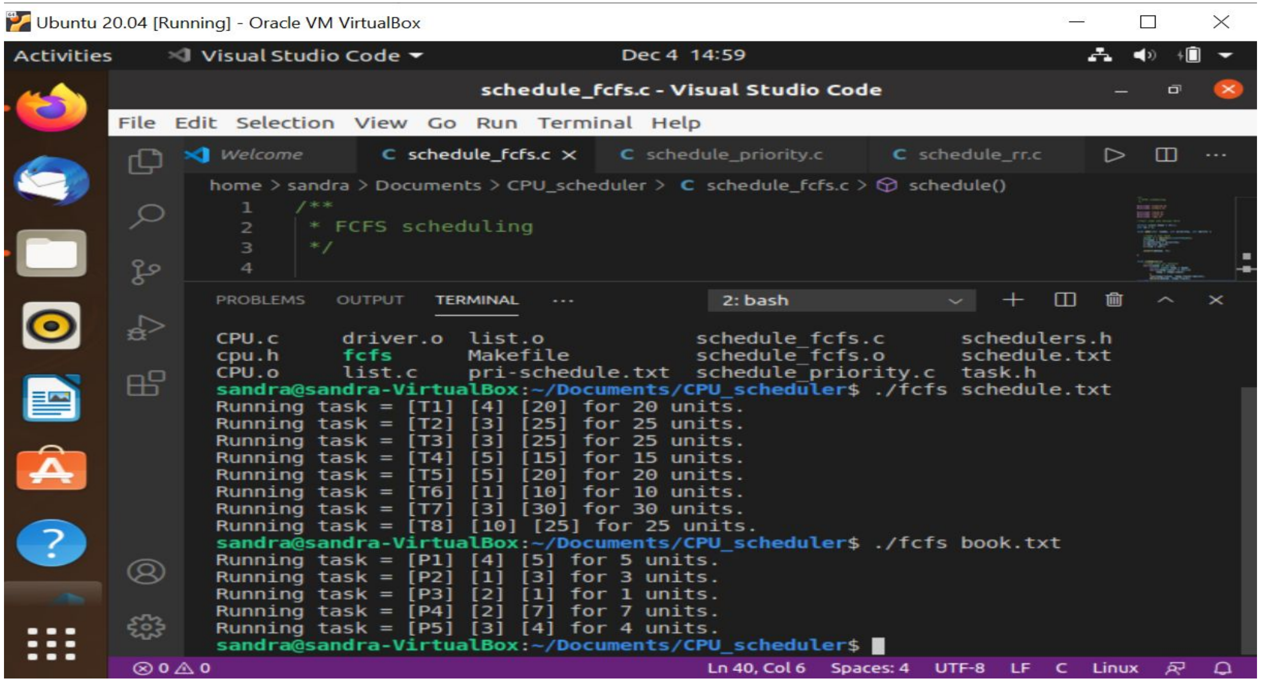
Abstract

This project focuses on the topic of CPU scheduling and involves the implementation of several process scheduling algorithms. The scheduler is assigned a predefined set of tasks and will schedule the tasks based on the selected scheduling algorithms. Each task is assigned a priority and a CPU burst. The priorities range from 1 to 10, where the higher number would indicate a higher priority. We were required to implement the scheduling algorithms: First-come, first-serve (FCFS), Priority scheduling and Round-robin (RR). Sample code was provided in the C files (*schedule_fcfs.c, schedule_priority.c, schedule_rr.c*). These files along with header files and a driver class read the schedule of tasks, insert the tasks into a list and invoke the scheduler. When printed, the schedule of tasks is displayed in the format [task name] [priority] [CPU burst].

Purpose

The purpose of this project was to implement the three CPU scheduling algorithms and to help us see what goes on in the scheduler when it is invoked. This assignment lets us see the difference between the ordering of the tasks, given the scheduling algorithm. First come first serve is the easiest CPU scheduling algorithm to implement and use. In this scheduling method, the process that requests the CPU first gets the allocation first. Priority based scheduling helps the OS involve priority assignments. Round robin scheduling helps for starvation free execution of processes.

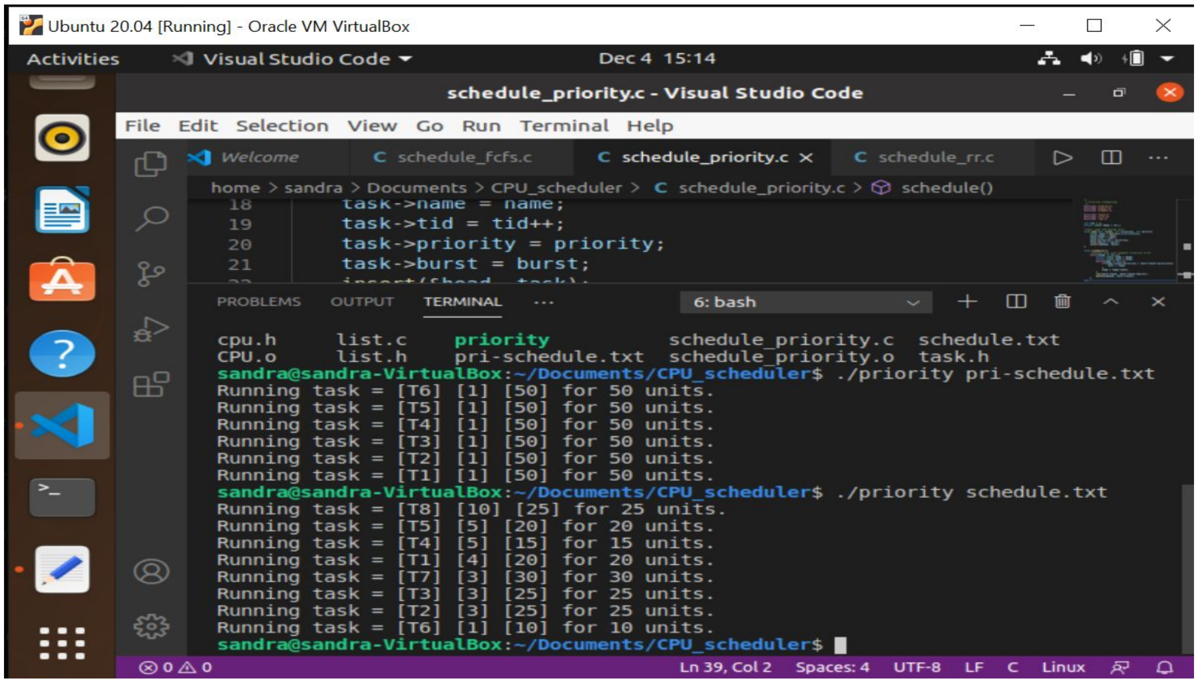
Results


Ubuntu 20.04 [Running] - Oracle VM VirtualBox
Activities
Visual Studio Code
Dec 4 14:59
schedule_fcfs.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Welcome
schedule_fcfs.c
schedule_priority.c
schedule_rr.c
home > sandra > Documents > CPU_scheduler > schedule_fcfs.c > schedule()
PROBLEMS OUTPUT TERMINAL
2: bash
Ln 40, Col 6 Spaces: 4 UTF-8 LF C Linux

```c
/**
 * FCFS scheduling
 */
```

```
CPU.c      driver.o  list.o            schedule_fcfs.c      schedulers.h
cpu.h      fcfs      Makefile          schedule_fcfs.o      schedule.txt
CPU.o      list.c    pri-schedule.txt  schedule_priority.c  task.h
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./fcfs schedule.txt
Running task = [T1] [4] [20] for 20 units.
Running task = [T2] [3] [25] for 25 units.
Running task = [T3] [3] [25] for 25 units.
Running task = [T4] [5] [15] for 15 units.
Running task = [T5] [5] [20] for 20 units.
Running task = [T6] [1] [10] for 10 units.
Running task = [T7] [3] [30] for 30 units.
Running task = [T8] [10] [25] for 25 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./fcfs book.txt
Running task = [P1] [4] [5] for 5 units.
Running task = [P2] [1] [3] for 3 units.
Running task = [P3] [2] [1] for 1 units.
Running task = [P4] [2] [7] for 7 units.
Running task = [P5] [3] [4] for 4 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$
```

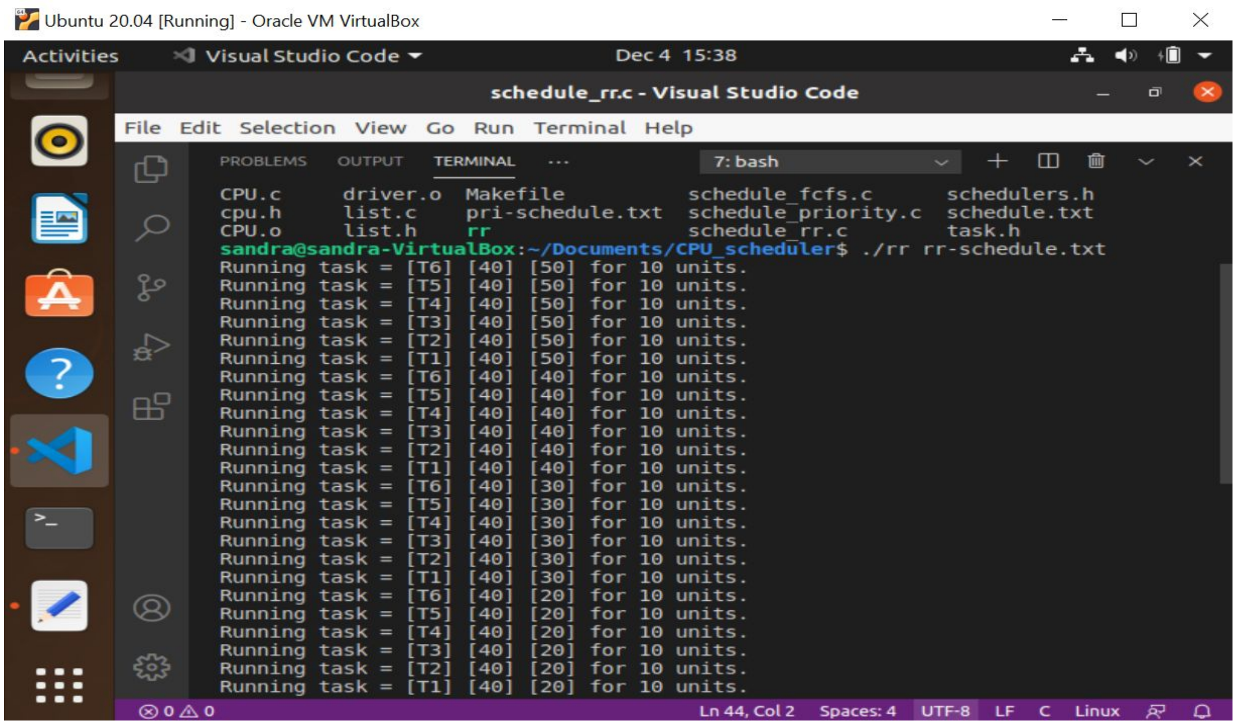
The screenshot above shows the scheduler being invoked with the fcfs algorithm using the files: schedule.txt and book.txt as arguments


Ubuntu 20.04 [Running] - Oracle VM VirtualBox
Activities
Visual Studio Code
Dec 4 15:14
schedule_priority.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Welcome
schedule_fcfs.c
schedule_priority.c
schedule_rr.c
home > sandra > Documents > CPU_scheduler > schedule_priority.c > schedule()
PROBLEMS OUTPUT TERMINAL
6: bash
Ln 39, Col 2 Spaces: 4 UTF-8 LF C Linux

```c
        task->name = name;
        task->tid = tid++;
        task->priority = priority;
        task->burst = burst;
```

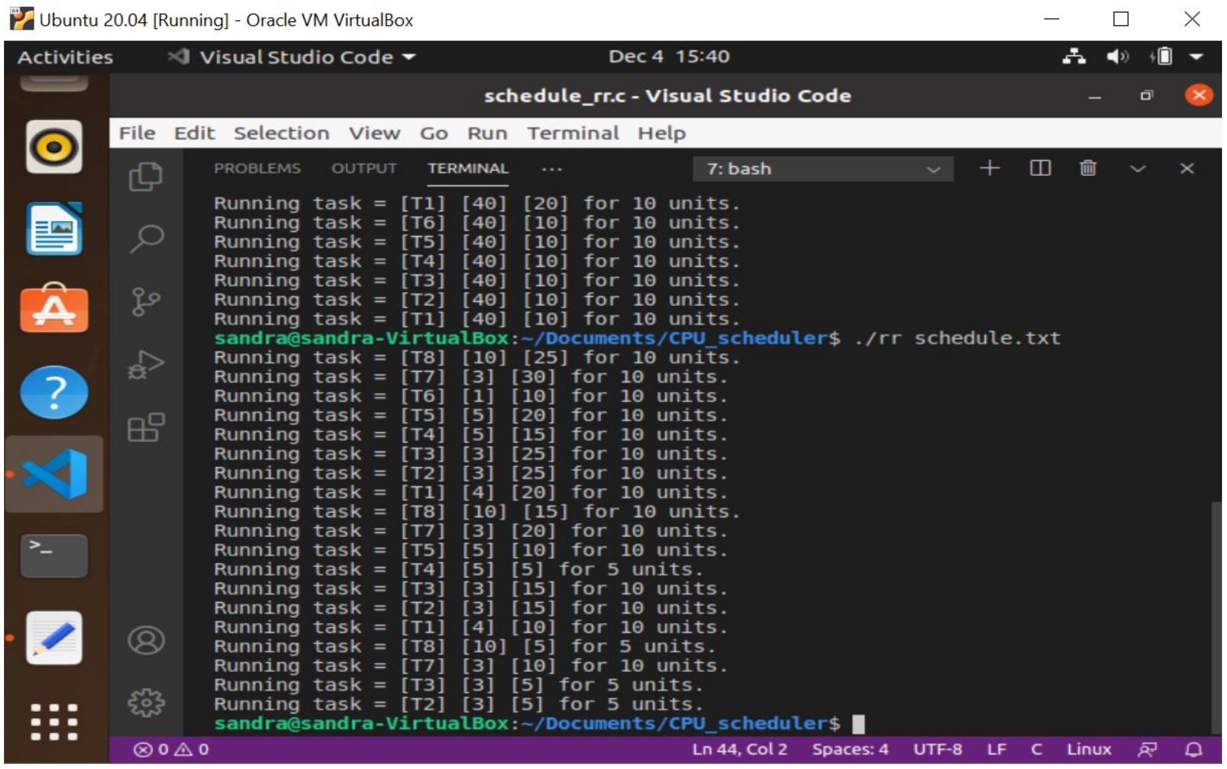
```
cpu.h     list.c    priority          schedule_priority.c  schedule.txt
CPU.o     list.h    pri-schedule.txt  schedule_priority.o  task.h
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./priority pri-schedule.txt
Running task = [T6] [1] [50] for 50 units.
Running task = [T5] [1] [50] for 50 units.
Running task = [T4] [1] [50] for 50 units.
Running task = [T3] [1] [50] for 50 units.
Running task = [T2] [1] [50] for 50 units.
Running task = [T1] [1] [50] for 50 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./priority schedule.txt
Running task = [T8] [10] [25] for 25 units.
Running task = [T5] [5] [20] for 20 units.
Running task = [T4] [5] [15] for 15 units.
Running task = [T1] [4] [20] for 20 units.
Running task = [T7] [3] [30] for 30 units.
Running task = [T3] [3] [25] for 25 units.
Running task = [T2] [3] [25] for 25 units.
Running task = [T6] [1] [10] for 10 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$
```

The screenshot above demonstrates the scheduler being invoked with the priority scheduling algorithm using schedule.txt and pri-scheduler.txt as arguments



Ubuntu 20.04 [Running] - Oracle VM VirtualBox
Activities
Visual Studio Code
Dec 4 15:38
schedule_rr.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL
7: bash
Ln 44, Col 2 Spaces: 4 UTF-8 LF C Linux

```
CPU.c      driver.o   Makefile            schedule_fcfs.c       schedulers.h
cpu.h      list.c     pri-schedule.txt    schedule_priority.c   schedule.txt
CPU.o      list.h     rr                  schedule_rr.c         task.h
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./rr rr-schedule.txt
Running task = [T6] [40] [50] for 10 units.
Running task = [T5] [40] [50] for 10 units.
Running task = [T4] [40] [50] for 10 units.
Running task = [T3] [40] [50] for 10 units.
Running task = [T2] [40] [50] for 10 units.
Running task = [T1] [40] [50] for 10 units.
Running task = [T6] [40] [40] for 10 units.
Running task = [T5] [40] [40] for 10 units.
Running task = [T4] [40] [40] for 10 units.
Running task = [T3] [40] [40] for 10 units.
Running task = [T2] [40] [40] for 10 units.
Running task = [T1] [40] [40] for 10 units.
Running task = [T6] [40] [30] for 10 units.
Running task = [T5] [40] [30] for 10 units.
Running task = [T4] [40] [30] for 10 units.
Running task = [T3] [40] [30] for 10 units.
Running task = [T2] [40] [30] for 10 units.
Running task = [T1] [40] [30] for 10 units.
Running task = [T6] [40] [20] for 10 units.
Running task = [T5] [40] [20] for 10 units.
Running task = [T4] [40] [20] for 10 units.
Running task = [T3] [40] [20] for 10 units.
Running task = [T2] [40] [20] for 10 units.
Running task = [T1] [40] [20] for 10 units.
```



Ubuntu 20.04 [Running] - Oracle VM VirtualBox
Activities
Visual Studio Code
Dec 4 15:40
schedule_rr.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL
7: bash
Ln 44, Col 2 Spaces: 4 UTF-8 LF C Linux

```
Running task = [T1] [40] [20] for 10 units.
Running task = [T6] [40] [10] for 10 units.
Running task = [T5] [40] [10] for 10 units.
Running task = [T4] [40] [10] for 10 units.
Running task = [T3] [40] [10] for 10 units.
Running task = [T2] [40] [10] for 10 units.
Running task = [T1] [40] [10] for 10 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$ ./rr schedule.txt
Running task = [T8] [10] [25] for 10 units.
Running task = [T7] [3] [30] for 10 units.
Running task = [T6] [1] [10] for 10 units.
Running task = [T5] [5] [20] for 10 units.
Running task = [T4] [5] [15] for 10 units.
Running task = [T3] [3] [25] for 10 units.
Running task = [T2] [3] [25] for 10 units.
Running task = [T1] [4] [20] for 10 units.
Running task = [T8] [10] [15] for 10 units.
Running task = [T7] [3] [20] for 10 units.
Running task = [T5] [5] [10] for 10 units.
Running task = [T4] [5] [5] for 5 units.
Running task = [T3] [3] [15] for 10 units.
Running task = [T2] [3] [15] for 10 units.
Running task = [T1] [4] [10] for 10 units.
Running task = [T8] [10] [5] for 5 units.
Running task = [T7] [3] [10] for 10 units.
Running task = [T3] [3] [5] for 5 units.
Running task = [T2] [3] [5] for 5 units.
sandra@sandra-VirtualBox:~/Documents/CPU_scheduler$
```

The two screenshots above demonstrate the tasks after the scheduler is invoked with the round robin scheduling algorithm

Contributions by group members:

- ***Antonio Hughes***
    - *Worked on the Priority scheduling*
    - *Worked on the Round robin scheduling*
- ***Sandra Chavez***
    - *Worked on the FCFS scheduling*
    - *Worked on the report (Abstract, Purpose, Results, Analysis)*
    - *Wrote the README file*

Analysis

Given the results of the different algorithms, we were able to see the different tasks when run through the scheduler. In FCFS, T1 ran first for 20 ms up until T8, which ran 25 ms. First come first serve schedules tasks in the order in which they request the CPU. In priority based, the task with the highest priority ran first and in this case, T8 ran first with priority number 10. The problem with this method is that low priority processes may never execute, resulting in starvation. In round robin, each task ran for the time quantum of 10 ms. After the time quantum has elapsed, the process is preempted and added to the end of the ready queue. Since each process is given a fixed time quantum, all processes are given the same priority. Our results matched the expected schedule of processes for each of the algorithms.