Lab #2 - CPU Scheduling

Karl-Johan Grinnemo
`karl-johan.grinnemo@kau.se`
Karlstad University — August 1, 2022

Preparations

Read Section 5.3, "Scheduling Algorithms", in the textbook (ninth and tenth edition).

Description

This lab aims at reinforcing the student's understanding of some common CPU scheduling algorithms: First-Come, First-Served (FCFS), Shortest-Job First (SJF), and Round Robin (RR). The students should implement a CPU-scheduling simulator, `sched`, that takes as input,

- a process information file, `<process information file>`, which contains information about *which* process arrives *when*, and *how much* CPU time it requests, i.e., *burst time*;
- the name of the employed scheduling algorithm: `FCFS`, `SJF`, or `RR`; and,
- if RR is employed, the time quantum, `<time quantum>`, in milliseconds.

The `sched` simulator computes:

- the *waiting time* and *turnaround time* for the respective process, and
- the *average waiting time* and *average turnaround time*.

The `sched` simulator should be implemented in C, C++, Java, or Python. Other programming languages might be possible to use, however, this needs to be discussed with the lab assistants.

A synopsis for the `sched` simulator is given below.

```
sched -f <process information file> -a [FCFS | SJF | RR] [-q <time quantum>]
```

The `-q` option is only applicable when the RR scheduling algorithm is selected.

The format of a process information file is as follows:

```
<PID>,<arrival time>,<burst time><newline character>
...
<PID>,<arrival time>,<burst time><newline character>
```

Both `<arrival time>` and `<burst time>` are given in milliseconds; `<PID>` denotes the process identification.

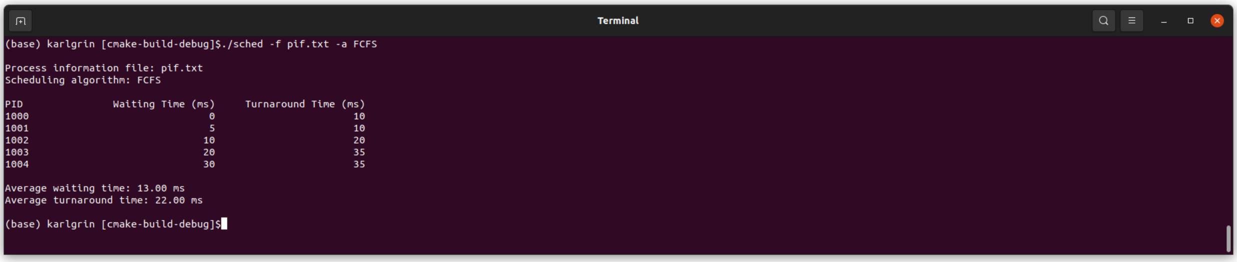
As an example, consider a process information file, `pif.txt`, with five processes with PIDs: 1000, 1001, 1002, 1003, and 1004, arrival times: 0 ms, 5 ms, 5 ms, 5 ms, and 10 ms, and with burst times: 10 ms, 5 ms, 10 ms, 15 ms, and 5 ms:

File

pif.txt

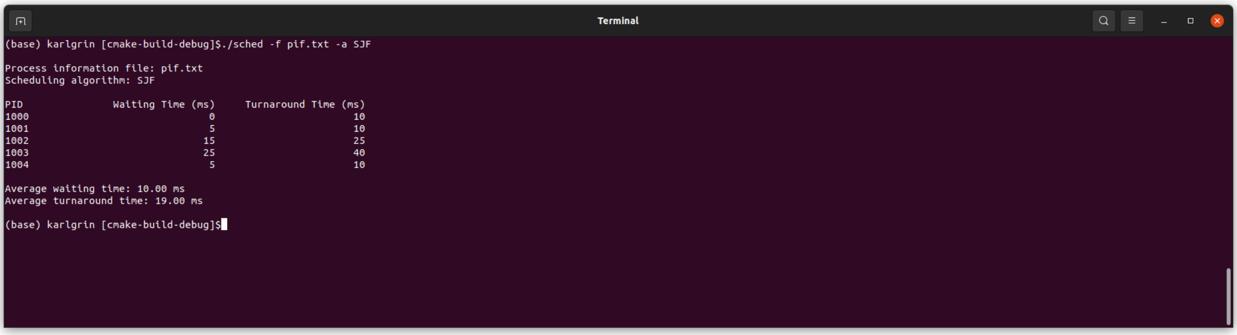
```
1000,0,10
1001,5,5
1002,5,10
1003,5,15
1004,10,5
```

The sample output for the three possible scheduling algorithms is shown in the figures below.



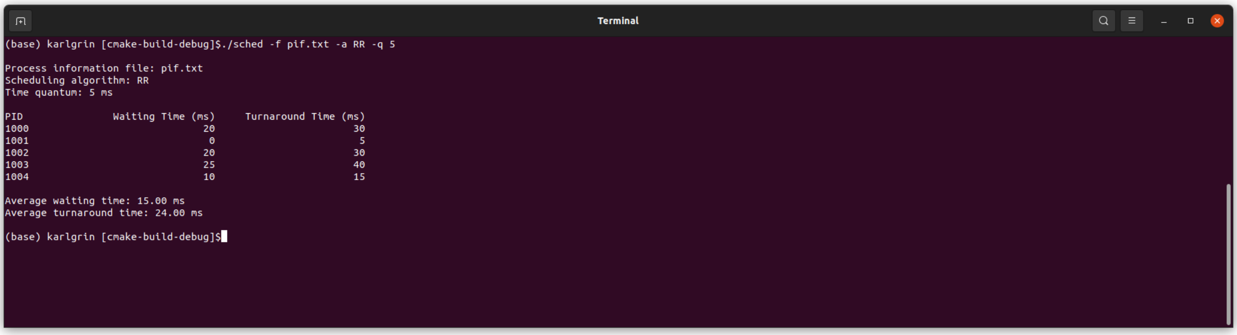
```
(base) karlgrin [cmake-build-debug]$./sched -f pif.txt -a FCFS

Process information file: pif.txt
Scheduling algorithm: FCFS

PID               Waiting Time (ms)     Turnaround Time (ms)
1000                              0                       10
1001                              5                       10
1002                             10                       20
1003                             20                       35
1004                             30                       35

Average waiting time: 13.00 ms
Average turnaround time: 22.00 ms

(base) karlgrin [cmake-build-debug]$
```

Figure 1: Screenshot from a simulation of FCFS.



```
(base) karlgrin [cmake-build-debug]$./sched -f pif.txt -a SJF

Process information file: pif.txt
Scheduling algorithm: SJF

PID               Waiting Time (ms)     Turnaround Time (ms)
1000                              0                       10
1001                              5                       10
1002                             15                       25
1003                             25                       40
1004                              5                       10

Average waiting time: 10.00 ms
Average turnaround time: 19.00 ms

(base) karlgrin [cmake-build-debug]$
```

Figure 2: Screenshot from a simulation of SJF.



```
(base) karlgrin [cmake-build-debug]$./sched -f pif.txt -a RR -q 5

Process information file: pif.txt
Scheduling algorithm: RR
Time quantum: 5 ms

PID               Waiting Time (ms)     Turnaround Time (ms)
1000                             20                       30
1001                              0                        5
1002                             20                       30
1003                             25                       40
1004                             10                       15

Average waiting time: 15.00 ms
Average turnaround time: 24.00 ms

(base) karlgrin [cmake-build-debug]$
```

Figure 3: Screenshot from a RR simulation with a time quantum of 5 ms.

Examination

The lab is graded as *pass* or *failed*. To pass, the students should demonstrate their CPU-scheduling simulator to a lab assistant.

End of Lab