Operating Systems

Endre Bangerter

Lab 02 - version 1: Due date 15.05.16

Lab policy:

- You may work alone or in groups of two.
- The deliverable of this lab is a written report answering the questions below. The report shall be written in a concise manner (clear, short, simple).
- You are free to have technical discussions etc. with any of your peers (of course). However, the lab report must be individual per group, and there is no (partial) copying allowed. Groups that participate in "report copying" will get 0 points for this lab.
- To avoid plagiarism, you must properly cite references when using material from books, websites etc. Please also refer to the BFH policies on plagiarism: https://www.bfh.ch/fileadmin/docs/recht/bfh/Richtlinie_Plagiate_d.pdf and https://www.bfh.ch/fileadmin/docs/recht/bfh/Richtlinie_Plagiate_f.pdf
- The deadline for handing in the report is strict.

1. Implement a simulator for a simplified version of the Windows scheduler

We have discussed the Windows scheduler in the lecture. It is a multi level queue scheduler that combines priority queues with round Robin scheduling. That is, it supports priorities in the range 0..15, whereas 15 is the highest and 0 the lowest priority. At each priority threads are scheduled using the Round Robin algorithm with a respective queue. When a thread with a higher priority than the actually running thread arrives, the latter is preempted. Also threads that use up a quantum are preempted. For simplicity, we do not consider priority boosts and other improvements used by the Windows scheduler.

The algorithms parameters are as follows:

- the quantum q (in abstract units)
- context switch time t (in abstract units)

The scheduler is invoked for the following events:

- When a new thread arrives.
- When a thread finishes a CPU burst or blocks on IO.
- When a thread finishes its IO burst, unblocks and thus arrives in the ready queue.
- When the quantum expires.

As an input the algorithm reads as input an ASCII file containing one line per thread of the following generic form:

```
THREADNAME[priority], arrival_time, (CPU_burst[1], IO_burst[1]), (CPU_burst[2], IO_burst[2]),...
```

A practical entry would look like:

```
T1[2]: 5, (10,20), (17,0), (10,12)
T2[15]: 9, (5,4), (5,4), (10,12)
...
```

The algorithms output will be the schedule of the form `thread id, start time, end time`. The output shall also contain the total overhead incurred by context switches.

```
T1: 5, 9
T2: 11, 16
...
...
...

----
Time spent on context switches: 16
Time spent on running threads: 200
```

Implement the algorithm in the programming language of your choice. Hand in the source code of the algorithm, as well as a compiled version, including exact installation instructions.

Usage of your code shall be as follows:

```
> lab2_solution INPUTFILE.TXT QUANTUM CONTEXT_SWITCH_TIME
```

A concrete invocation of the program would e.g,. be as follows:

```
> lab2_solution test.txt 10 5
```