Operating Systems Lab 3: Scheduling and page replacement

Note: For this lab, submissions on Themis will not be checked for correctness! In other words, there is no 'expected output'. We only check that the program is not crashing and does not contain memory leaks. Functionality will be checked by the TA's when grading.

Note: For this lab, you will have to write a short report (pdf or README). In there, briefly describe your design decisions and the implementation of your code.

1 Scheduling

In this exercise you are going to do some measurements on your own virtual schedulers. You will make both a simple non-preemptive FCFS scheduler, and a preemptive Round-Robin scheduler with three priority queues. Queue 1 will be highest priority and queue 3 will be the lowest priority. Use 10ms as the quantum for preemptive scheduling.

Remember that most processes are a mix of CPU and I/O. Assume that there is only one CPU and one I/O device. The scheduler will receive its input on `stdin` that will look like:

```
1 0 100 25 100 20 50 20 100 10 200 -1
3 15 30 15 30 10 40 10 50 -1
...
```

Each line represents one process. The first column will be arrival time of the process, the second is its priority (this one can be ignored by the FCFS-scheduler). After that, you have alternating CPU- and I/O-times, starting with CPU time. A -1 indicates the end of a process. When all processes are done, the scheduler should print the calculation of the average turnaround time for all processes to `stdout` (do not forget the newline!). Remember that turnaround time is the time between creation of the process (i.e. entering the queue) and finishing the process.

On Themis, submit two separate implementations, one for the Round-Robin implementation and one for the FCFS implementation. Both have the exact same test cases.

2 Page replacement algorithm

In this exercise you are going to do some measurements on your own page replacement algorithms. These will be the FIFO page replacement algorithm and the clock page replacement algorithm. Again, make two separate implementations to hand in on Themis.

The program will receive the input on its `stdin`, for example:

```
4
1 2 3 4 7 2 5 2 3 1 7 6 4
```

The first line indicates the amount of physical memory frames. After that, there will be a stream of page references. You may assume a maximum of 10 frames, 50 pages and 100 references. When the program is finished, it should print the number of page faults to `stdout`.