Operating Systems – **Practice 9, Implementation of Priority Scheduler in NachOS**

Description

Priority Scheduler has the following characteristics:

- A priority is given to each process (thread), and this priority is an entry in the Process Control Block (PCB)
- Priorities have a fixed range of integer values and can be defined by the user
- Lower integer values mean higher priority values
- The process with the highest priority gets the CPU
- Processes with the same priority value are handled as if scheduler was FCFS
- The process voluntarily yields the CPU

Implementation

1. At the beginning of `threadtest.cc`, include the names of the students that worked in this practice, date, and what files were modified.
2. A new menu will be shown to the user in this practice, compared to the previous one:

```
MENU
1. Simple Thread
2. Threads (FCFS)
3. Producer-Consumer (FCFS)
4. Threads (Priority)
5. Exit
```

   Option chosen by the user is executed once, and the program finishes regardless what option was chosen. (Once code has been compiled, every single option can be modified without modifying the code again).
3. If user choses option 4, he needs to enter a priority value for each thread. Give values from 0 to 5 to these priorities, where 0 is the highest priority
4. When a thread is executed (option 4), priority value gets stored at its PCB
5. `fork()` will not be modified
6. For option 4, `scheduler.cc` and `scheduler.h` will be modified. Use methods implemented in `list.cc` and `list.h` To store priority value at PCB, modify `thread.cc` and/or `thread.h` `threadtest.cc` creates and begins execution of threads
7. Code must be commented with the implemented modifications

Deliverables

Submit a document to DidacTIC, following guidelines as before, that explains what needs to be done and how you did it in order to work on this practice. Highlight files that were modified and show the code, bot original and modified ones, that was modified.