Laboratory 4

Part I
Prepare (at least 4) workload mixes having different characteristics, ranging from all compute-intensive benchmarks to all I/O-intensive benchmarks. Each workload should spawn around 5 processes. You can compose these workloads of benchmarks from the UnixBench suite, or write your own.

Study how the standard Minix3 scheduler (https://minixnitc.github.io/scheduling.html) schedules your workloads. Your study should not only encompass the order of the processes, but also the time quanta spent in the CPU by the processes. Record your observations and your inferences.

(Use the *fork*, *exit*, and *swap* logging mechanisms from the previous assignments for analysis/ debugging; turn off the prints while measuring time).

Part II
Modify the user-level scheduler in Minix3 to the following "*Pseudo-FIFO*" policy: among the user-level processes that are ready to execute, the one that entered the earliest must be scheduled.

You may not be able to implement it *in totality*. Justify why your version is a reasonable approximation.

(Another Clarification - let's simplify things and focus on CPU intensive processes. Given a workload consisting of only CPU intensive processes, we wish that the process that was spawned the earliest finishes first. And when it finishes, processes that were spawned after it should have made little (ideally it should be zero, but 'little' is acceptable) progress. Also, discuss in your report how your scheduler will behave when there is a workload mix containing some IO intensive processes.)

Study how the workloads you prepared are scheduled under the new scheduler. Compare with the original Minix3 scheduler. Discuss your observations.

Submission: same as before: a zip file containing the modified source files, a script that copies the files and builds the new kernel, and a report of you findings.

Hints:
minix/servers/sched/schedule.c
minix/kernel/proc.c
http://www.minix3.org/docs/scheduling/report.pdf
Git: You may use the same branch as the previous assignment.