

## CS2518 - P3 - PRIORITY SCHEDULER

### OVERVIEW

In this practical it is the aim to extend the scheduler of the *minios* that was introduced in the lectures. When registering a new process it should be possible to also specify a priority for a process.

### TOOLS

Please use the *minios* assembler program provided on Canvas as starting point. Use the MIPS Mars assembler simulator to complete the assignment.

### THE SCHEDULER MODIFICATION TASK

A new process is registered using *syscall 100*. The process address is passed on to the syscall in register \$a0. The syscall should be modified such that in \$a1 we can pass a priority to the operating system scheduler for this process.

The priority of each process should then be stored in the process control block (the pcb needs to be extended).

Next the interrupt handler needs to be modified. A process of priority 0 should only run for one quantum. A process of priority 1 should run 2 quanta, a process with priority 2 should run 3 quanta and so on. Thus, a counter is required to track how many quanta the active process has been running. Only if the allocated number of quanta has been used the next process is scheduled. Otherwise the current process will run again.

If a process will run again it is not necessary to complete all steps in the interrupt handler. In this case it is not necessary to save registers or to restore registers.

---

### CS2518 CONTINUOUS ASSESSMENT - PART 3

Please submit your modified and commented minios assembler program you created.

In addition, answer the following question in detail: *Why can a process of priority 1 complete slightly more than twice the work of a process with priority 0?*

The programs and your answer has to be submitted at the end of the semester together with questions from the other practicals (You do not have to complete all tasks/answers within the practical slot).