Homework 6: Scheduling

Answer all questions. Submit your hand-written answer sheets to the instructor before the lecture begins.

1 Scheduling

The goal of this homework is to understand the scheduling mechanism of Pintos. Look at the `timer_init`, `intr_register_ext`, `register_handler` routines, which set up the handler for the timer interrupt.

Look at the `intr_entry` (called on every interrupt) and `intr_handler` routines. `intr_handler` calls the `timer_interrupt` routine on timer interrupt event. If `timer_interrupt` routine decides to call schedule, it sets `yield_on_return`.

- Turn in: After how many timer interrupts `yield_on_return` is set.
- Turn in: The summary of the `thread_yield` routine.

Notice that a timer interrupt can also occur during the handling of a system call, and a system call handler can also be preempted (i.e., another thread might get scheduled during the partial execution of a system call handler).

- Turn in: Why do we need a different kernel stack for every thread? What is the problem if we have just one kernel stack? Give an example.
- Turn in: Can we operate Pintos with only one kernel stack? What else is needed if we want to keep only one kernel stack?