# Assignment #1

## Getting an information for the assignment

Your job is to write an answer for each question. This may require a thorough analysis of several pieces of source code in Pintos. The purpose of this assignment is to get familiar with Pintos and figure out how it works. You MUST answer all the questions. You also MUST write down your answer in at least 3 sentences for each sub-question. We strongly recommend to explain the source code line-by-line. Sometimes, each question requires thought, but it does not need much time to solve the question. Follow the below instructions and submit your assignment through the Blackboard system.

## Q1) timer_sleep() function

In Q1, students should investigate `"device/timer.c"`, `"device/timer.h"`, and `"thread/thread.c"` files to answer the following sub-questions. Answer the following questions. The timer_sleep() is implemented in `"device/timer.c"`.

- Q1-1) Describe what the "timer_tick()" does.
- Q1-2) Describe how often interrupts occur in Pintos and which functions are called due to the interrupt.
- Q1-3) Describe how the "ticks" variable changes when an interrupt occurs.
- Q1-4) Describe what the "timer_elapsed()" does.
- Q1-5) Describe what the "timer_sleep()" does **in detail**.

## Q2) thread_yield() function

In Q2, students should investigate `"thread/thread.c"`, `"thread/thread.c"`, `"lib/kernel/list.c"`, and `"lib/kernel/list.h"` files. Answer the following questions. The thread_yield() is implemented in `"thread/thread.c"`.

- Q2-1) Describe each member field in the "struct thread".
- Q2-2) Describe what the "running_thread()" does.
- Q2-3) Describe the four statuses that a thread can have during its life cycle.
- Q2-4) Describe what the "list_push_back()" does.
- Q2-5) Describe why the "struct list read_list" variable is needed for scheduling.
- Q2-6) Describe what the "thread_yield()" does **in detail**.

## Q3) schedule() function

In Q3, students should investigate `"thread/thread.c"`, `"thread/thread.h"`, and `"thread/switch.S"` files. Answer the following questions. The schedule() is implemented in `"thread/thread.c"`.

- Q3-1) Describe what the "next_thread_to_run()" and "list_entry()" does, respectively.
- Q3-2) Describe what the "switch_threads()" does.
- Q3-3) Describe what the "switch_schedule_tail()" does.
- Q3-4) Describe what the "schedule()" does **in detail**.

## Q4) thread_block() & thread_unblock() function

In Q4, students should investigate `"thread/thread.c"`, `"thread/thread.h"` files. Answer the following questions. thread_block() and thread_unblock() are implemented in `"thread/thread.c"`.

- Q4-1) Describe what the "thread_current()" does.
- Q4-2) Describe what the "is_thread()" does.
- Q4-3) Describe what the "thread_block()" and "thread_unblock()" does **in detail**.

## Q5) init_thread() function and getting/setting priority of a thread

In Q5, students should investigate `"thread/thread.c"`, `"thread/thread.h"` files. Answer the following questions. init_thread() is implemented in `"thread/thread.c"`.

- Q5-1) Describe a range of priority a thread can have and the value of "PRI_MIN", "PRI_MAX", and "PRI_DEFAULT".
- Q5-2) Describe what the "init_thtread()" does **in detail**.
- Q5-3) Describe what the "thread_get_priority()" and "thread_set_priority" does.

## How to submit my homework

- Submit your report file through an assignment submission system in our class blackboard.
- You MUST submit your report in PDF format.
- You MUST follow the file naming rule like StudentID_Name.pdf (e.g., 20234512_Honggilldong.pdf).
- **Any kind of plagiarism is strictly prohibited in this class.**

## Due date

- **Submission deadline: Mon April 8, 23:59 pm**
- **Late submission deadline: Thu April, 11 23:59 pm**
- **Late submission policy:** 10% penalty will be applied per day after the due date