Project 1

This project is worth 5% of your final grade.

Due Wednesday 23rd March (2 weeks)

Goal:

Reimplement timer_sleep(), defined in 'devices/timer.c'. Although a working implementation is provided, it "busy waits," that is, it spins in a loop checking the current time and calling thread_yield() until enough time has gone by. Reimplement it to avoid busy waiting.

1. void timer_sleep (int64 t ticks) [Function] Suspends execution of the calling thread until time has advanced by at least x timer ticks. Unless the system is otherwise idle, the thread need not wake up after exactly x ticks. Just put it on the ready queue after they have waited for the right amount of time. timer_sleep() is useful for threads that operate in real-time, e.g. for blinking the cursor once per second or for a RR scheduler.

2. The argument to timer_sleep() is expressed in timer ticks, not in milliseconds or any another unit. Do not change this value, because any change is likely to cause many of the tests to fail.

3. Separate functions timer_msleep(), timer_usleep(), and timer_nsleep() do exist for sleeping a specific number of milliseconds, microseconds, or nanoseconds, respectively, but these will call timer_sleep() automatically when necessary. You do not need to modify them.

The test cases you will need to successfully run for this part of the project are:

1. alarm-single
2. alarm-multiple
3. alarm-simultaneous
4. alarm-zero
5. alarm-negative

You will find the expected output in

pintos/src/tests/threads/foo.ck

where foo is the name of the test.

These are also run when you execute **make check**. You may need to **make clean** on occasion.

Hints

1. Read through the first tutorial.
2. Become familiar with pintos – this will take a while.
3. Look over:

a. Pintos/src/devices/timer.h

b. Pintos/src/devices/timer.c

c. Pintos/src/threads/thread.h

d. Pintos/src/lib/kernel/list.h

4. Find examples of using lists and interacting with threads throughout the source.

Hand in

As there is little scope for design in this project, and the main effort will be to become familiar with pintos, only working (COMMENTED) code need be submitted. Please indicate the code you have added with:

/* my code begins */

Your code here

/* my code ends */

For later projects you will be required to also submit design documents and a writeup.