Done by:

Yuguang Huang

Shreyas Bhivandkar

Do Nhat Anh

Sadman Sakib Hasan

EECS 3221 – ASSIGNMENT 2

**Table of contents**

1. Introduction -------------------------------------- pg. 1
2. Description --------------------------------------- pg. 1
3. Difficulties ----------------------------------------- pg. 2-3
4. Solution --------------------------------------------
5. Source code ---------------------------------------
6. Test output ----------------------------------------
7. Makefile --------------------------------------------

**My\_Alarm.c**

# INTRODUCTION

My\_Alarm.c is a C Program which can be used to set multiple alarms for different timings. The user has to set the alarm with a reminder message which is displayed when the alarm goes off. The code is made for POSIX-based systems. The input is provided by the user on the Terminal.

# DESCRIPTION

The Program uses PThreads for handling multiple alarms which run parallel to each other. These threads are programmed to print Epoch seconds when they run displaying the difference between the time when the alarm is set and when it goes off. The alarm displays the time in epoch seconds when the alarm was received followed by when the alarm was retrieved and also when it exits. These epoch seconds keep printing continuously until the countdown for the alarm is finally over.

# DIFFICULTY

1. Since the threads were running in parallel, the epoch seconds were getting printed twice. The synchronization caused incorrect outputs for the alarms as well. For example a 3 seconds alarm will go off at the same time as 5 second alarm. Or a 5 Second alarm will not go off after 5 second. This issue is due using one thread for all alarms at the beginning.
2. We faced a major issue while synchronizing the threads.

The issue with printing countdown seconds is that the sleep lock for the thread gets activated after the printing is finished. For example, a 5 seconds alarm will print the countdown to 5 seconds (epoch seconds) and then wait for another 5 seconds before the alarm goes off making it a total wait time of 10 seconds.

# SOLUTION

1. The solution to the 1st