1. Introduction

In this programming project, we will develop a solution for “boiler men and bathers problem”. The details of the problem and the project and its requirements will be presented using a PPT presentation.

2. Program specifications

Your final program must satisfy the following requirements:
(a) Three bathers and two boiler men should be concurrently running (multi-tasked).
(b) The first process becomes the first boiler man (B1) and should create other four processes (one is another boiler man, and three are bathers) using fork system call.
(c) Newly created processes must wait for all processes to be created and start running.
(d) When each process enters the critical section, each process outputs the following message:
   - “B# starts his boiler …”, if it is a boiler-man and # = 1 or 2 (boiler-man number)
   - “T# is entering the bathing area”, if it is a bather and # = 1, 2, or 3 (bather number)
(e) When a bather is leaving the bathing area, each bather outputs the following message:
   - “T# is leaving the bathing area”
(f) When a boiler-man is leaving the bathing area, the boiler-man outputs the following message:
   - “B# is leaving the bathing area”
(g) Each process waits for a random amount of time each time in the bathing area or outside of the area.
(h) Boiler men continue to repeat until each of them finishes repeating NUM_REPEAT times. The three bathers terminated after both boiler men repeat NUM_REPEAT times.
(i) The main process should delete semaphore(s) and the shared memory before it terminates itself.
(j) To control the activity timing of the bathers and the boiler-men, the following five labels should be declared at the top of your source code and be used in the critical section as specified below. Figure 1 shows the required structure of the critical section for each bather. Figure 2 shows the required structure of the critical section for each boiler-man.
   - NUM_REPEAT: the number of times each boiler-man does his work (heat up the water)
   - BATHER_TIME_01: the time interval a bather sleeps out of the pool.
   - BATHER_TIME_02: the time interval a bather stays in the pool.
   - BOLIERMAN_TIME_01: the time interval a boiler-man does his work.
   - BOLIERMAN_TIME_02: the time a needs to do his work (heating up the water).
3. Requirements

Your program must follow the following requirements:

(a) No starvation should occur.
(b) No violation of mutual exclusion should occur for boiler men.
(c) No deadlock should occur (all the processes should be always completed)
(d) Multiple bathers should be able to enter the bathing area (at least several times in a program execution).

4. Objectives

This programming is designed for the following objectives

(1) To understand the concept of process
(2) To understand the concept of inter process communication (IPC)
(3) To understand the concept of process synchronization
(4) To develop system programming skills to manage race condition and critical section
(5) To develop programming skills to avoid process starvation and deadlock
(6) To have UNIX system programming experience
(7) To be familiar with UNIX-based operating systems

5. Grading Criteria

1. Any compile-time error (-80%)
2. Minor run-time error: -5% (for each)
3. A program that satisfies (a) no race condition, (b) no violation for mutual condition, and (c) concurrent bathers: 85% of the credit
4. Design and implement “starvation-free” for the two boiler-men: 15% of the credit
5. Major run-time error (failing the meet any of requirement (a) through (d): depends.
6. Program style (program structure and in-line comments): 10%

6. Guidelines for acceptable activities

1. Using external references (web sites, programming reference books, and etc.): Allowed.
2. Exchanging ideas with your classmates: Not recommended but acceptable
3. Exchanging source codes with your classmates: Not acceptable (absolutely not allowed, no matter how many lines of the codes).
4. Requesting someone else (but yourself) to write any code for you for this assignment: Not acceptable (absolutely not allowed). Exams may ask you some questions regarding your code. If you fail to convince Dr. Fujinoki your work, your assignment credit can be cancelled).
5. Sample source code files posted to the course (CS314, Spring 2017) home by the course instructor (Fujinoki): Allowed (recommended).

8. Required submission

Program softcopy (should be e-mailed to the instructor) by the project due (11:00 AM on Match23rd).
9. Late Submission

- Penalty of -10% will be given for every 24 hours after the due (i.e., -10% for a submission within the first 24 hours after the due).
- Submission more than 48 hours after the due will not be accepted.

APPENDIX:

The five constant labels for defining time interval should be placed at the beginning of your source code file as shown below:

```c
/* ********************************************************** */
/* CS314 Project #2 solution */
/* Your last-three: */
/* Your course section #: */
/* Spring 2017 */
/* ********************************************************** */

#define NUM_REPEAT 50   // each boiler-man repeats
#define BATHER_TIME_01 300   // 300ms = 0.3 seconds
#define BATHER_TIME_02 800   // 800ms = 0.8 seconds
#define BOLIERMAN_TIME_01 1200   // 1200ms = 1.2 seconds
#define BOLIERMAN_TIME_02 1600   // 1600ms = 1.6 seconds

#include <........>
```