REPORT

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that
I have acknowledged all material and sources used in its preparation, whether they be books, articles,
reports, lecture notes, and any other kind of document, electronic or personal communication. I also
certify that this assignment/report has not previously been submitted for assessment in any other course,
except where specific permission has been granted from all course instructors involved, or at any other
time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other
students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In
addition, I understand my responsibility to report honour violations by other students if I become aware
of it.

Name: A. Venkata Sai Mahesh
Date:25/02/2020
Signature: Venkata Sai Mahesh Abburi

Name: J. Deepak Reddy
Date:25/02/2020
Signature: Deepak Reddy Jillela

DESCRIPTION

This program comprises 3 computational functions.
1.umpire_main
2.player_main
3.musical_chairs

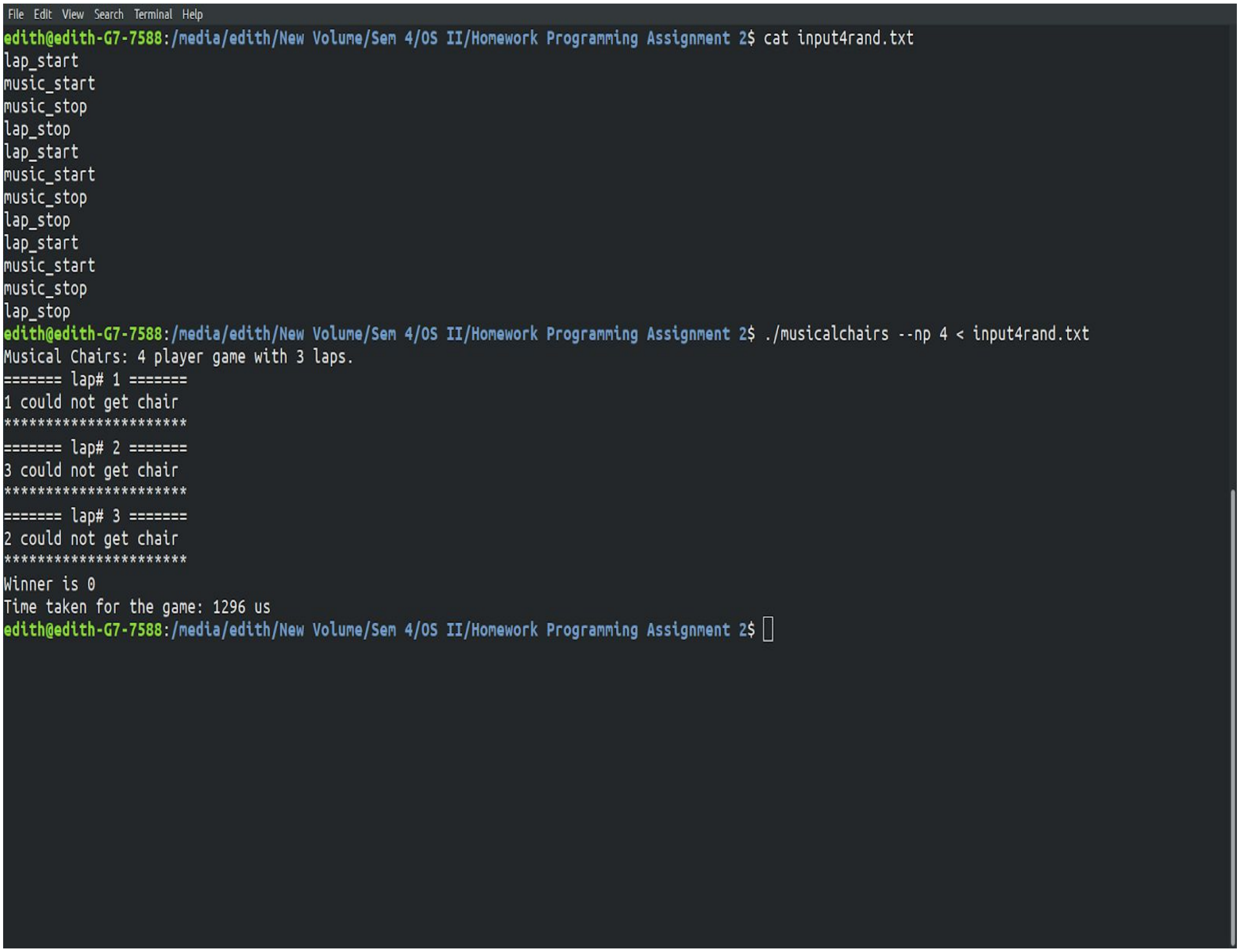
musical_chairs:
In this function umpire and player, threads are initialized to perform umpire_main and player_main functions respectively. Before and after spawning of threads clock() time is captured in order to find the time taken for the game. The main thread waits for umpire and player threads to complete their execution their respective functions.

umpire_main and player_main:
The umpire thread keeps track of the number of laps elapsed and proceeds accordingly. For every lap, the umpire thread reads the command from the file. The player threads wait for the umpire thread to start the lap, this happens when the umpire thread reads the command lap_start. Now the player thread waits until the umpire starts the music. Once the umpire thread reads the command music_start then the player threads start playing the game speaking classically(actually run in a loop until the music is stopped). If the umpire thread reads the command umpire_sleep then umpire thread goes to sleep for the specified period of time while player threads awaiting for the music halt and if the umpire thread reads the command player sleep then the desired time for sleep is added to the corresponding player index in the array which is used to store the sleeping times of players. Meanwhile

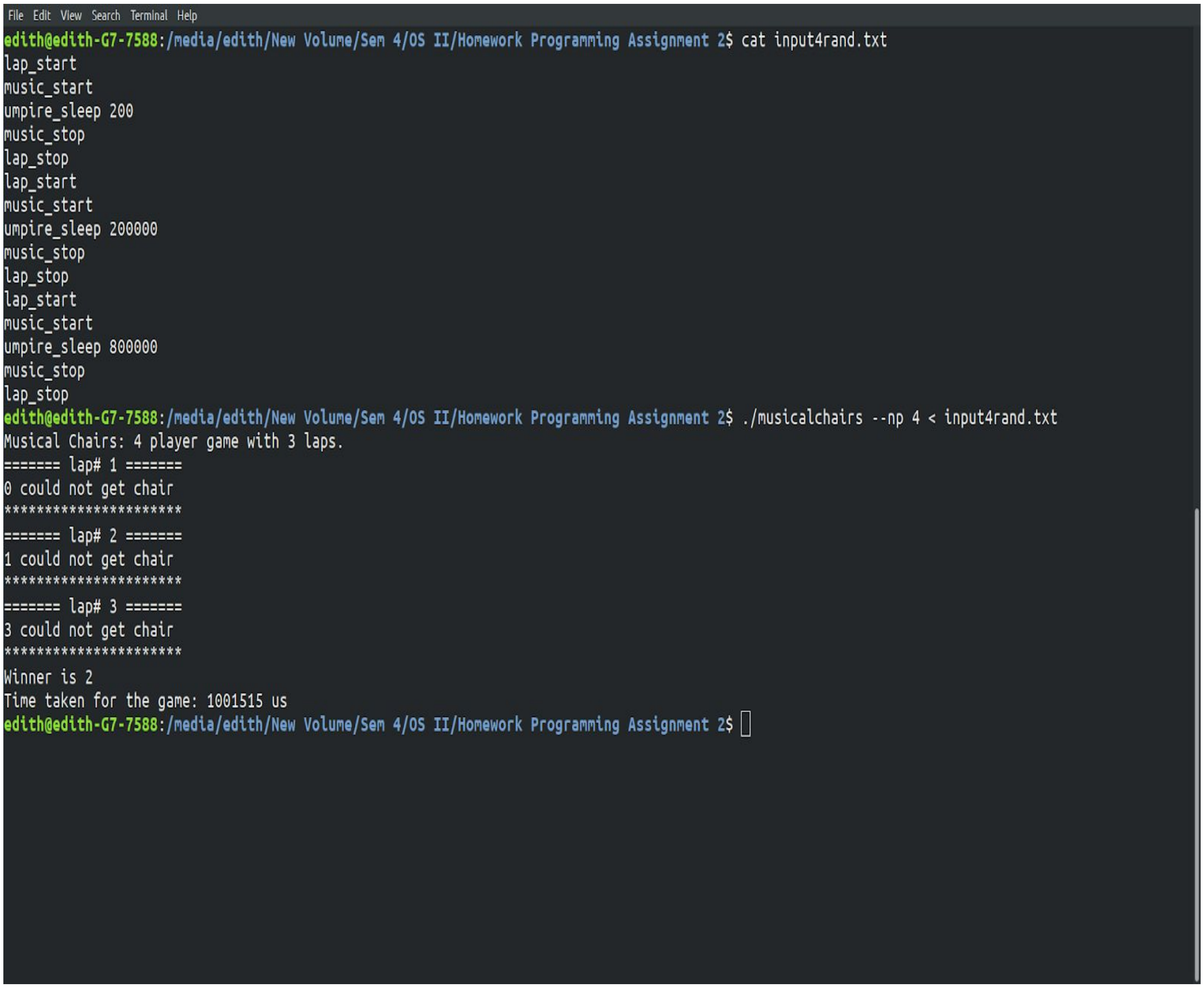
waiting for the music halt the player threads may sleep for a certain time as specified and later waits for the music to be stopped. When the umpire reads the music_stop command then the player threads start occupying chairs(speaking classically)[in practical a count variable is used to keep track of number chairs occupied] and gives up CPU until the last player(loser of the round) notifies. If the count reaches one less than the number of currently playing players, then the last player notifies umpire that the round is ended declares himself that he couldn’t get a chair and the thread terminates. This notification of the last player to the umpire(synchronization) is achieved with the use of the conditional variable. The remaining player threads which acquired the chair waits until the umpire thread notifies that lap is completed. This synchronization is also achieved with the help of a condition variable. To synchronize the status of the number of chairs available among the different player threads mutex lock [ensures the consistency of a variable which is used to keep track of the number of chairs available]. Each round ends when all currently available chairs are occupied. Finally, the game ends when only one player is left.

OBSERVATIONS

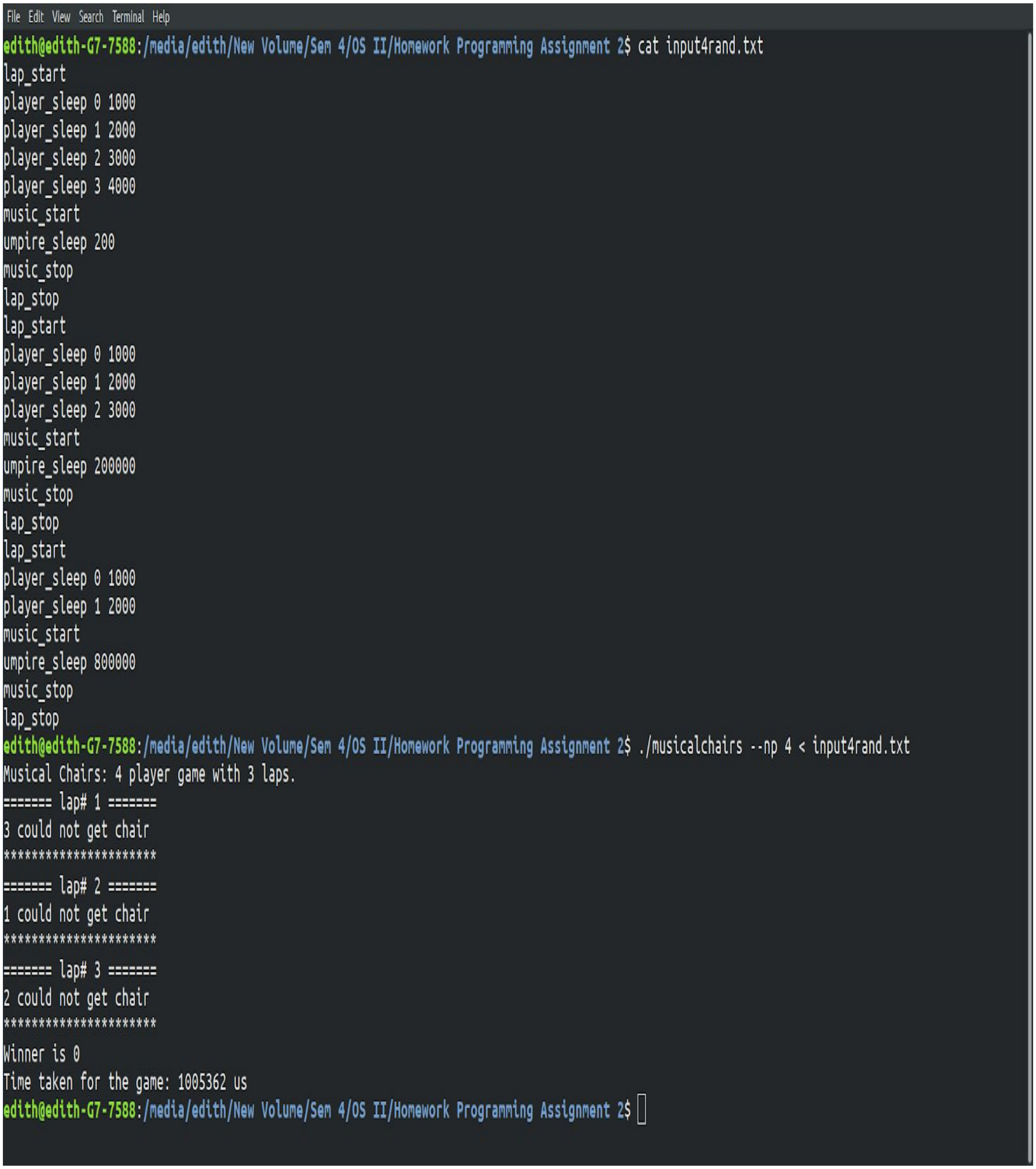
1. Order of execution of threads cannot be determined.
2. Use of mutex locks instead of condition variables may waste CPU resources which in turn increases the time of execution.
3. Using busy waiting instead of condition variables consumes much time and had occurred TLE in the autograder and may also lead to deadlocks.
4. Synchronization must be performed to ensure the smooth execution of musical chairs game.



```
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ cat input4rand.txt
lap_start
music_start
music_stop
lap_stop
lap_start
music_start
music_stop
lap_stop
lap_start
music_start
music_stop
lap_stop
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ ./musicalchairs --np 4 < input4rand.txt
Musical Chairs: 4 player game with 3 laps.
======= lap# 1 =======
1 could not get chair
**********************
======= lap# 2 =======
3 could not get chair
**********************
======= lap# 3 =======
2 could not get chair
**********************
Winner is 0
Time taken for the game: 1296 us
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ 
```



```
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ cat input4rand.txt
lap_start
music_start
umpire_sleep 200
music_stop
lap_stop
lap_start
music_start
umpire_sleep 200000
music_stop
lap_stop
lap_start
music_start
umpire_sleep 800000
music_stop
lap_stop
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ ./musicalchairs --np 4 < input4rand.txt
Musical Chairs: 4 player game with 3 laps.
======= lap# 1 =======
0 could not get chair
**********************
======= lap# 2 =======
1 could not get chair
**********************
======= lap# 3 =======
3 could not get chair
**********************
Winner is 2
Time taken for the game: 1001515 us
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ 
```



```
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ cat input4rand.txt
lap_start
player_sleep 0 1000
player_sleep 1 2000
player_sleep 2 3000
player_sleep 3 4000
music_start
umpire_sleep 200
music_stop
lap_stop
lap_start
player_sleep 0 1000
player_sleep 1 2000
player_sleep 2 3000
music_start
umpire_sleep 200000
music_stop
lap_stop
lap_start
player_sleep 0 1000
player_sleep 1 2000
music_start
umpire_sleep 800000
music_stop
lap_stop
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ ./musicalchairs --np 4 < input4rand.txt
Musical Chairs: 4 player game with 3 laps.
======= lap# 1 =======
3 could not get chair
**********************
======= lap# 2 =======
1 could not get chair
**********************
======= lap# 3 =======
2 could not get chair
**********************
Winner is 0
Time taken for the game: 1005362 us
edith@edith-G7-7588:/media/edith/New Volume/Sem 4/OS II/Homework Programming Assignment 2$ 
```