

CS3523: OS-II - ASSIGNMENT 2

Aim: The aim of this assignment is to understand how classical IPC synchronization problems surface when developing concurrent programs.

Problem:

A kindergarten school wants to conduct musical chairs competition among “n” players who show interest in the competition. The game proceeds like this:

1. The school arranges n-1 chairs at random places in the ground.
2. n players will be standing in the ground at random places.
3. The game iterates n-1 times. In each of the ith lap,
 - a. The umpire starts the lap by ensuring that all players are ready.
 - b. Once all players are ready the umpire starts the game by starting music.
 - c. Each of the players keeps running around in the ground till music plays.
 - d. When music is stopped by the umpire, each of the players quickly chooses a chair to occupy it. But if the chair is already occupied, the player steps back and chooses another free chair, and so on.
 - e. One player will not get any chair as there is one less chair, so this player is out of game.
 - f. The umpire stops the lap by ensuring that one player is knocked out, all other players are up from the chairs and a chair is removed from the game to start the next lap.

Write a program to mimic this game by creating one thread for the umpire and one thread for each of n players. The umpire thread works with all the players threads in lockstep synchronization. Each iteration, one player thread exits. Main thread waits for umpire thread and the n players to join back, and declares who has won the game.

Input specification

Examples shown below has commands to control umpire and player's execution.

- Only the umpire thread reads these input commands.
- lap_start, lap_stop, music_start, music_stop are commands for letting player threads to synchronize with umpire and act on.
- “umpire_sleep <sleep>” which is an optional command specified between music_start and music_stop to make umpire run slower (just by the umpire sleep) between

music_start and music_stop for a specific lap. If umpire_sleep is not specified in any lap, by default umpire threads does not sleep between music_start and music_stop.

- “player_sleep <playerid> <sleep>” is another optional command specified between lap_start and music_start to make these player threads run slower (just by making them sleep) between music_start and music_stop for a specific lap. If player_sleep is not specified in any lap, by default player threads do not sleep between music_start and music_stop.

If player_sleep is specified in a lap, but the player thread already exited, then the sleep time is ignored.

Syntax:

```
musicalchairs --nplayers <n>
```

Eg 1 (No sleep for players and umpire)-

```
$ cat input4randfast.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
```

```
$ ./a.out --np 4 < input4randfast.txt
```

```
Musical Chairs: 4 player game with 3 laps.
===== lap# 1 =====
2 could not get chair
*****
===== lap# 2 =====
0 could not get chair
*****
===== lap# 3 =====
3 could not get chair
*****
Winner is 1
Time taken for the game: 1728 us
```

Eg 2 (Sleeping umpire)-

```
$ 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
```

```
$ ./musicalchairs --np 4 < input4rand.txt
```

```
Musical Chairs: 4 player game with 3 laps.
===== lap# 1 =====
2 could not get chair
*****
===== lap# 2 =====
1 could not get chair
*****
===== lap# 3 =====
3 could not get chair
*****
Winner is 0
Time taken for the game: 1014484 us
```

Eg 3 - (Sleeping players and sleepy umpire)

```
$ cat input4deterministic.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
```

```
$ ./a.out --np 4 < input4deterministic.txt
```

```
Musical Chairs: 4 player game with 3 laps.
```

```
===== lap# 1 =====
```

```
3 could not get chair
```

```
*****
```

```
===== lap# 2 =====
```

```
2 could not get chair
```

```
*****
```

```
===== lap# 3 =====
```

```
1 could not get chair
```

```
*****
```

```
Winner is 0
```

```
Time taken for the game: 1012762 us
```