Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Spring 2023

Homework 6 – Capture the Chair Game using Threads and Queues

Due: 30/05/2023, Tuesday, 21:00

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!

Introduction

In this homework, you are asked to write a **multithreaded** C++ program that simulates the universal game called *musical chairs* (https://en.wikipedia.org/wiki/Musical_chairs) (sandalye kapmaca in Turkish) using queue. The game starts after taking number of players as input. It will simulate each round of the game via several threads (one for each player) and print relevant information as can be seen in the sample runs. At each round, after the threads start, the threads will sleep until a certain time so that all players start the game at the same time. You will simulate capturing a chair by enqueueing the player ID to a queue shared by all threads. Please see "Details of the Program" section for more information.

And please first learn how to write *multithreaded* programs by checking out the lecture notes, sample codes and the videos, and then start the homework. Multithreaded programs are totally different than other type of programs that you wrote until now. You cannot learn it while doing the homework; first you have to learn and then write.

Using Threads

The number of threads is equal to the number of players. The number of players will be taken as input from the user. Your threads will access common resource(s), such as a queue. To avoid any synchronization conflicts while accessing them, you need to use mutex(es). At each round, your threads must be joined properly. Moreover, at the end, your program must terminate without any complications. In other words, you have to see exit code as 0.

Details of the Program

The game consists of rounds where each player aims to capture a chair to sit. The number of chairs is one less than the number of players. At each round, the player that fails to capture a chair will be eliminated. The number of chairs will also decrease by one after each round so that the players are eliminated one by one.

In the beginning, the user will input the number of players playing the game. This number is important for the size of the queue and number of threads (remember you have to have one thread for each player).

Each player will be given an ID consecutively starting from 0. It is up to you how to keep and manage these IDs.

In each round of the game, the program will first get the current time and display it. Then the player threads will start. Each player will hunt for a chair at the same time, which is 2 seconds after the current time. For the sake of fairness, you have to sleep all the player threads until this certain absolute time, so you have to use `sleep_until` (not `sleep_for`). Here be careful about overflows from seconds (at the end of the minutes) while calculating the absolute time which is 2 seconds after the current time. You may ignore the minute overflows at the end of the hours; we will not test your codes at the end of the hours. However, if you want to handle this, you are more than welcome.

At each round, each player will try and capture the chairs by trying to `enqueue` its ID to a common integer queue accessed by all threads. Since the queue size is one less than the number of players of the current round, one player will not be able to enqueue. Success or failure to capture a chair must be displayed by each player thread (see sample runs). Since the queue is accessed by all of the threads, in order to avoid any synchronization conflicts, you have to use a `mutex` properly.

As you see above, every player is given a signal after 2 seconds to capture a chair. The aim is to capture the chairs before the other players. This is where we bridge the analogy to multithreaded applications, the threads will act as the players, and the queue will act as the chair(s) to sit. The threads will be given a start time and then they try to lock the `mutex` before others.

The entry point of each of the threads will be the same function. That means, you will **NOT** write separate functions for the threads. In order to customize this function for each player, you may need to pass parameter(s) to it.

After each round is over, the threads need to be joined properly. Then, the main program should display the IDs players who captured a chair on the screen in the capturing order (in a single line; see the sample runs). After that, the next round should start by eliminating the player who could not capture a chair in the previous round. Do not forget that at each round, the number of chairs (the queue size) is decremented by one.

The game finishes when there is a single player left who is going to be declared as winner.

You may assume that input for the number of players is a positive integer greater than equal to 2, so no input checks are necessary.

Use of global variables and helper structures

You may use global variables in this homework. However, we kindly request you not to exaggerate the global usage since after a certain point you may lose control over your program (as the famous Turkish proverb says "azı karar, coğu zarar").

As explained above, you have to use an integer queue which is accessed by all threads. Other than this, you can use extra vectors or other structures in main, but the game logic (capturing chairs) in the thread function must be implemented via queue.

`IntQueueHW6` class

We provide header and implementation files of IntQueueHW6 class within the homework pack. This is a static integer queue class very similar to the one that we have seen in class. You will use this class for the queue in the game. You are not allowed to change it. During grading, we will use it verbatim. Please examine this class before starting the homework.

SUBMISSION GUIDELINES (CHANGED, READ IT)

There is no CodeRunner in this homework (see the reasons in the sample runs section below). Thus you have to submit your main cpp file as file upload.

There is a naming convention (SUNetUsername_LastName_OtherNames_HW6.cpp). For example, malikuskonmaz_kuskonmaz_mehmetali_HW6.cpp. Wrong naming costs 5 points (no kidding, I mean it).

Do not submit the IntQueueHW6 files.

Do not use any Turkish characters in the file name and anywhere in the program.

If you submit wrong file, you cannot change it after the deadline. Especially, realizing this after the grading does not help. We grade what you submit to SUCourse with no exceptions.

After uploading your cpp file, do not forget to complete the submission process. If you cannot successfully complete the submission process, we cannot see your homework and the grade will be zero with no exceptions.

Please see the previous homework specifications for the other important rules

Sample Runs

Some sample runs are given below. Due to the scheduling of threads, same input may yield different outputs for your code. However, the order of the events must be consistent with the homework requirements and the given input. Especially, the order chair captures should not have any pattern. The player who could not capture a chair should be displayed after the ones who captured. The duration between current time and the chair captures must always be 2 seconds in each round. At each round, the player who couldn't sit should be eliminated.

Due to the reasons explained above (different outputs for the same inputs) we cannot use CodeRunner in this homework. Thus, you have to test manually.

The inputs from the keyboard are written in ***`boldface and italic`***.

Sample Run 1:

```
Welcome to Musical Chairs game!
Enter the number of players in the game:
2
Game Start!

Time is now 16:28:01
Player 0 captured a chair at 16:28:03.
Player 1 couldn't capture a chair.
Remaining players are as follows: 0

Game over!
Winner is Player 0!
```

Sample Run 2:

```
Welcome to Musical Chairs game!
Enter the number of players in the game:
6
Game Start!

Time is now 16:31:53
Player 4 captured a chair at 16:31:55.
Player 5 captured a chair at 16:31:55.
Player 2 captured a chair at 16:31:55.
Player 3 captured a chair at 16:31:55.
Player 0 captured a chair at 16:31:55.
Player 1 couldn't capture a chair.
Remaining players are as follows: 4 5 2 3 0

Time is now 16:31:55
Player 0 captured a chair at 16:31:57.
Player 5 captured a chair at 16:31:57.
Player 4 captured a chair at 16:31:57.
Player 3 captured a chair at 16:31:57.
Player 2 couldn't capture a chair.
Remaining players are as follows: 0 5 4 3

Time is now 16:31:57
Player 5 captured a chair at 16:31:59.
Player 4 captured a chair at 16:31:59.
Player 3 captured a chair at 16:31:59.
Player 0 couldn't capture a chair.
Remaining players are as follows: 5 4 3

Time is now 16:31:59
Player 3 captured a chair at 16:32:01.
Player 5 captured a chair at 16:32:01.
Player 4 couldn't capture a chair.
Remaining players are as follows: 3 5

Time is now 16:32:01
Player 3 captured a chair at 16:32:03.
Player 5 couldn't capture a chair.
Remaining players are as follows: 3

Game over!
Winner is Player 3!
```

Sample Run 3:

```
Welcome to Musical Chairs game!
Enter the number of players in the game:
6
Game Start!

Time is now 16:34:54
Player 4 captured a chair at 16:34:56.
Player 5 captured a chair at 16:34:56.
Player 0 captured a chair at 16:34:56.
Player 1 captured a chair at 16:34:56.
Player 2 captured a chair at 16:34:56.
Player 3 couldn't capture a chair.
Remaining players are as follows: 4 5 0 1 2

Time is now 16:34:56
Player 4 captured a chair at 16:34:58.
Player 1 captured a chair at 16:34:58.
Player 5 captured a chair at 16:34:58.
Player 0 captured a chair at 16:34:58.
Player 2 couldn't capture a chair.
Remaining players are as follows: 4 1 5 0

Time is now 16:34:58
Player 1 captured a chair at 16:35:00.
Player 4 captured a chair at 16:35:00.
Player 5 captured a chair at 16:35:00.
Player 0 couldn't capture a chair.
Remaining players are as follows: 1 4 5

Time is now 16:35:00
Player 4 captured a chair at 16:35:02.
Player 1 captured a chair at 16:35:02.
Player 5 couldn't capture a chair.
Remaining players are as follows: 4 1

Time is now 16:35:02
Player 4 captured a chair at 16:35:04.
Player 1 couldn't capture a chair.
Remaining players are as follows: 4

Game over!
Winner is Player 4!
```

Sample Run 4:

```
Welcome to Musical Chairs game!
Enter the number of players in the game:
15
Game Start!

Time is now 16:36:11
Player 2 captured a chair at 16:36:13.
Player 0 captured a chair at 16:36:13.
Player 1 captured a chair at 16:36:13.
Player 13 captured a chair at 16:36:13.
Player 4 captured a chair at 16:36:13.
Player 12 captured a chair at 16:36:13.
Player 5 captured a chair at 16:36:13.
Player 14 captured a chair at 16:36:13.
Player 9 captured a chair at 16:36:13.
Player 6 captured a chair at 16:36:13.
Player 8 captured a chair at 16:36:13.
Player 7 captured a chair at 16:36:13.
Player 11 captured a chair at 16:36:13.
Player 10 captured a chair at 16:36:13.
Player 3 couldn't capture a chair.
Remaining players are as follows: 2 0 1 13 4 12 5 14 9 6 8 7 11 10

Time is now 16:36:13
```

```
Player 1 captured a chair at 16:36:15.
Player 2 captured a chair at 16:36:15.
Player 4 captured a chair at 16:36:15.
Player 12 captured a chair at 16:36:15.
Player 5 captured a chair at 16:36:15.
Player 0 captured a chair at 16:36:15.
Player 9 captured a chair at 16:36:15.
Player 6 captured a chair at 16:36:15.
Player 8 captured a chair at 16:36:15.
Player 14 captured a chair at 16:36:15.
Player 13 captured a chair at 16:36:15.
Player 10 captured a chair at 16:36:15.
Player 11 captured a chair at 16:36:15.
Player 7 couldn't capture a chair.
Remaining players are as follows: 1 2 4 12 5 0 9 6 8 14 13 10 11

Time is now 16:36:15
Player 5 captured a chair at 16:36:17.
Player 11 captured a chair at 16:36:17.
Player 12 captured a chair at 16:36:17.
Player 8 captured a chair at 16:36:17.
Player 0 captured a chair at 16:36:17.
Player 14 captured a chair at 16:36:17.
Player 9 captured a chair at 16:36:17.
Player 4 captured a chair at 16:36:17.
Player 1 captured a chair at 16:36:17.
Player 2 captured a chair at 16:36:17.
Player 13 captured a chair at 16:36:17.
Player 10 captured a chair at 16:36:17.
Player 6 couldn't capture a chair.
Remaining players are as follows: 5 11 12 8 0 14 9 4 1 2 13 10

Time is now 16:36:17
Player 1 captured a chair at 16:36:19.
Player 2 captured a chair at 16:36:19.
Player 10 captured a chair at 16:36:19.
Player 14 captured a chair at 16:36:19.
Player 4 captured a chair at 16:36:19.
Player 11 captured a chair at 16:36:19.
Player 12 captured a chair at 16:36:19.
Player 8 captured a chair at 16:36:19.
Player 9 captured a chair at 16:36:19.
Player 13 captured a chair at 16:36:19.
Player 0 captured a chair at 16:36:19.
Player 5 couldn't capture a chair.
Remaining players are as follows: 1 2 10 14 4 11 12 8 9 13 0

Time is now 16:36:19
Player 9 captured a chair at 16:36:21.
Player 0 captured a chair at 16:36:21.
Player 1 captured a chair at 16:36:21.
Player 4 captured a chair at 16:36:21.
Player 11 captured a chair at 16:36:21.
Player 8 captured a chair at 16:36:21.
Player 13 captured a chair at 16:36:21.
Player 14 captured a chair at 16:36:21.
Player 10 captured a chair at 16:36:21.
Player 2 captured a chair at 16:36:21.
Player 12 couldn't capture a chair.
```

```
Remaining players are as follows: 9 0 1 4 11 8 13 14 10 2

Time is now 16:36:21
Player 11 captured a chair at 16:36:23.
Player 4 captured a chair at 16:36:23.
Player 10 captured a chair at 16:36:23.
Player 13 captured a chair at 16:36:23.
Player 2 captured a chair at 16:36:23.
Player 8 captured a chair at 16:36:23.
Player 14 captured a chair at 16:36:23.
Player 9 captured a chair at 16:36:23.
Player 1 captured a chair at 16:36:23.
Player 0 couldn't capture a chair.
Remaining players are as follows: 11 4 10 13 2 8 14 9 1

Time is now 16:36:23
Player 1 captured a chair at 16:36:25.
Player 2 captured a chair at 16:36:25.
Player 4 captured a chair at 16:36:25.
Player 11 captured a chair at 16:36:25.
Player 8 captured a chair at 16:36:25.
Player 9 captured a chair at 16:36:25.
Player 10 captured a chair at 16:36:25.
Player 13 captured a chair at 16:36:25.
Player 14 couldn't capture a chair.
Remaining players are as follows: 1 2 4 11 8 9 10 13

Time is now 16:36:25
Player 13 captured a chair at 16:36:27.
Player 2 captured a chair at 16:36:27.
Player 1 captured a chair at 16:36:27.
Player 11 captured a chair at 16:36:27.
Player 8 captured a chair at 16:36:27.
Player 9 captured a chair at 16:36:27.
Player 10 captured a chair at 16:36:27.
Player 4 couldn't capture a chair.
Remaining players are as follows: 13 2 1 11 8 9 10

Time is now 16:36:27
Player 1 captured a chair at 16:36:29.
Player 11 captured a chair at 16:36:29.
Player 10 captured a chair at 16:36:29.
Player 8 captured a chair at 16:36:29.
Player 13 captured a chair at 16:36:29.
Player 2 captured a chair at 16:36:29.
Player 9 couldn't capture a chair.
Remaining players are as follows: 1 11 10 8 13 2

Time is now 16:36:29
Player 8 captured a chair at 16:36:31.
Player 11 captured a chair at 16:36:31.
Player 13 captured a chair at 16:36:31.
Player 1 captured a chair at 16:36:31.
Player 2 captured a chair at 16:36:31.
Player 10 couldn't capture a chair.
Remaining players are as follows: 8 11 13 1 2

Time is now 16:36:31
Player 1 captured a chair at 16:36:33.
Player 11 captured a chair at 16:36:33.
Player 13 captured a chair at 16:36:33.
Player 8 captured a chair at 16:36:33.
Player 2 couldn't capture a chair.
Remaining players are as follows: 1 11 13 8

Time is now 16:36:33
Player 13 captured a chair at 16:36:35.
Player 8 captured a chair at 16:36:35.
Player 1 captured a chair at 16:36:35.
Player 11 couldn't capture a chair.
Remaining players are as follows: 13 8 1

Time is now 16:36:35
Player 1 captured a chair at 16:36:37.
Player 8 captured a chair at 16:36:37.
Player 13 couldn't capture a chair.
Remaining players are as follows: 1 8

Time is now 16:36:37
Player 8 captured a chair at 16:36:39.
Player 1 couldn't capture a chair.
Remaining players are as follows: 8

Game over!
Winner is Player 8!
```

Good Luck!
Albert Levi, Selim Kırbıyık