

Teste de 17 de Maio de 2021

Alef Keuffer

June 7, 2022

Parte I

Parte I Q1

“A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource, resulting in both programs ceasing to function.”

Parte I Q2

Temos que lidar com spurious wakeups e fazer `while(!condition) wait(condition)`. Isso torna a implementacao de uma barreira reutilizavel menos direta (por que?).

Parte II

```
import java.util.*;

public class AMatchMaker /*implements MatchMaker*/ {

    public static void main(String argv[]) {
        System.out.println("Compiles");
    }

    static class Cell {
        BoundedBuffer b;
        boolean ready = false;

        synchronized boolean notReady() {return !ready;}

        synchronized BoundedBuffer get() {
            if (b == null)
                b = new BoundedBuffer();
            ready = true;
            return b;
        }
    }

    Queue<Cell> producers = new ArrayDeque<>();
    Queue<Cell> consumers = new ArrayDeque<>();

    BoundedBuffer waitForConsumer() throws InterruptedException {
        Cell c = consumers.peek();
        if (c == null) {
            c = new Cell();
            producers.add(c);
            while (c.notReady())
                c.wait();
        }
        else {
            consumers.remove();
            c.notify();
        }

        return c.get();
    }

    BoundedBuffer waitForProducer() throws InterruptedException {
        Cell c = producers.peek();
        if (c == null) {
            c = new Cell();
            consumers.add(c);
            while (c.notReady())
                c.wait();
        }
        else {
            producers.remove();
            c.notify();
        }

        return c.get();
    }

}

class BoundedBuffer {}
```

Parte III

```
-module(matchMaker).
-export([start/0, waitForConsumer/0,waitForProducer/0]).

% funcoes de interface
start() ->
    register(?MODULE,spawn(fun() -> matchMaker([],[]) end)).

waitForConsumer() ->
    ?MODULE ! {waitForConsumer,self()}.
waitForProducer() ->
    ?MODULE ! {waitForProducer,self()}.

% funcoes privadas
matchMaker([C|Cs],[P|Ps]) ->
    C ! P ! boundedBuffer(),
    io:fwrite("Received bounded buffer!~n", []), % just to get feedback on shell
    matchMaker(Cs,Ps);

matchMaker(Cs,Ps) ->
    receive
        {waitForConsumer,P} ->
            matchMaker(Cs,[P|Ps]);
        {waitForProducer,C} ->
            matchMaker([C|Cs],Ps)
    end.

% Just to make it compilable
boundedBuffer() -> boundedBuffer.
```