Exercício 1 - Tranca

Lucas Augusto Tansini - 265038
Lucas Valandro - 243675
Prof. Claudio Fernando Resin Geyer
Disciplina - Programação Distribuída e Paralela

Tranca é um mecanismo de sincronização que permite que uma, ou mais threads mães, aguarde que outra thread termine sua execução.

Seu funcionamento acontece por meio da Classe ***CountDownLatch*** inicializa a processo através do seu construtor, recebendo um ***int*** que corresponde ao número de threads que se deseja aguardar, como por exemplo no código:

- CountDownLatch latch = new CountDownLatch(3);

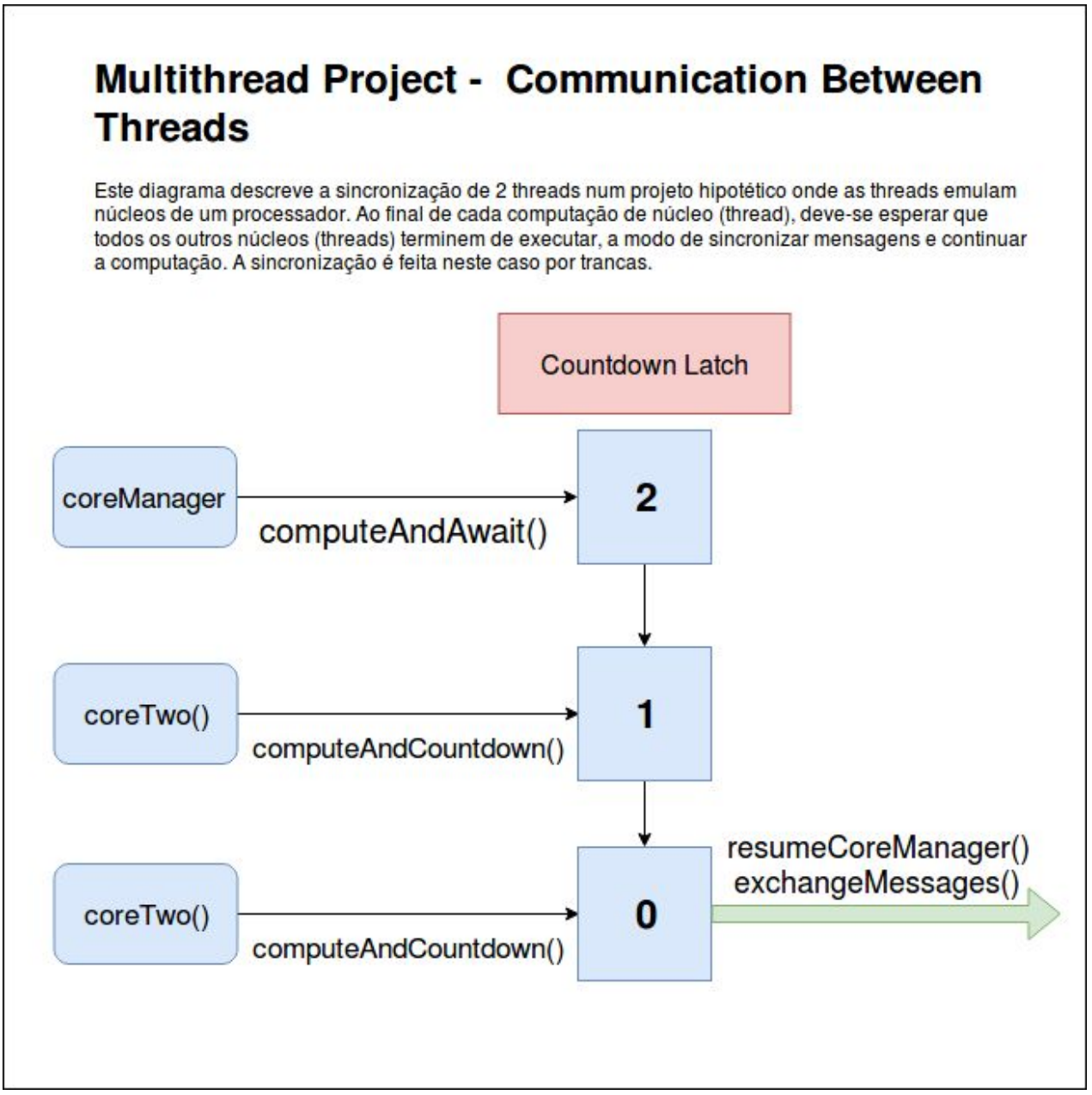
Após a criação da tranca, deve-se executar as ***N threads*** que se deseja aguardar, e **imediatamente** após a chamada das threads, deve-se chamar:

- latch.await();

A execução do código irá parar no latch.await() e irá aguardar até que todas as threads filhas executem e notifiquem a thread mãe com:

- latch.countDown();

Essa função decrementa o contador e quando ele atinge **zero**, o fluxo de execução volta para a thread mãe. O diagrama abaixo explica o funcionamento descrito acima (note que o método await() foi modificado para o método computeAndAwait() para ficar mais didático). O código em anexo também descreve um exemplo de código para trancas.



Application.java

```
import java.util.concurrent.CountDownLatch;

public class Application {
	public static void main(String[] args) {

	CountDownLatch latch = new CountDownLatch(3);


	new Thread(new ProcessThread("Thread 1",latch, 2000L)).start();
```

```
        new Thread(new ProcessThread("Thread 2",latch, 6000L)).start();
        new Thread(new ProcessThread("Thread 3",latch, 4000L)).start();

        System.out.println("Espera Threads filhas executarem e notificarem....");

        try {
        latch.await();
        } catch (InterruptedException e) {
        e.printStackTrace();
        }

        System.out.println("Todas Threads executaram....");

        System.out.println("Voltando para a Thread mãe....");

        }
}
```

ProcessThread.java

```
import java.util.concurrent.CountDownLatch;

public class ProcessThread implements Runnable {
        private CountDownLatch latch;
        private Long workDuration;
        private String name;

        ProcessThread(String name, CountDownLatch latch, Long duration) {
        this.name = name;
        this.latch = latch;
        this.workDuration = duration;
        }

        @Override
        public void run() {
        try {
        System.out.println(name +" Processando "+ workDuration/1000 + " segundos...");
        Thread.sleep(workDuration);
        } catch (InterruptedException e) {
        e.printStackTrace();
        }
        System.out.println(name+ " Execução terminada!");

        latch.countDown();
        }
```

```
}
```