

Docente: Prof. Sergio T. Carvalho

Discentes: Ariel Marte Araújo Silva 201900264

Marco Feitosa Araújo 201905542

Experimento 01: Criação e Execução das Threads

Entrada de Dados: Ao todo 5 Strings contendo “Usuario 0X”, onde X corresponde a uma referencia a String

Saída Esperada: 5 impressões de cada String (ou de cada usuário) totalizando 25 impressões, podendo a ser ordem aleatória

Saída de Dados: Foram constadas ao todo 25 impressões sendo distribuídas em: 11 impressões “Usuario 05”; 06 impressões “Usuario 03”; E 07 impressões “Usuario 01”

Entrada de Dados:	Saída Esperada:	Saída de Dados:
Usuario 01	Usuario 01	Usuario 05
Usuario 02	Usuario 02	Usuario 05
Usuario 03	Usuario 03	Usuario 05
Usuario 04	Usuario 04	Usuario 05
Usuario 05	Usuario 05	Usuario 05
	Usuario 01	Usuario 05
	Usuario 01	Usuario 05
	Usuario 01	Usuario 05
	Usuario 02	Usuario 05
	Usuario 02	Usuario 05
	Usuario 02	Usuario 03
	Usuario 04	Usuario 01
	Usuario 04	Usuario 01
	Usuario 04	Usuario 01
	Usuario 03	Usuario 01
	Usuario 03	Usuario 05
	Usuario 03	Usuario 01
	Usuario 05	Usuario 01
	Usuario 05	Usuario 01
	Usuario 05	Usuario 01
	Usuario 01	Usuario 03
	Usuario 02	Usuario 03
	Usuario 03	Usuario 03
	Usuario 04	Usuario 03
	Usuario 05	Usuario 03

Análise:

Por não haver controle de exclusão mútua, enquanto uma thread está “dormindo” após setar o nome do recurso, outra thread altera este nome. Assim, quando a primeira acorda e mostra o nome registrado no recurso, este nome não será mais o esperado. Com isso, dificilmente haverá corretamente o número esperado de vezes que cada usuário tem seu nome mostrado em tela.

```
public void run() {  
    for (int i = 0; i < 5; i++) {  
        recurso.setTexto(nomeThread); // Seta recurso compartilhado  
        try {  
            sleep(30);                // thread dorme  
        } catch (Exception e) {  
        }  
        recurso.mostraTexto(); // Usa recurso compartilhado  
    }  
}
```

Experimento 02: Criação e Execução das Threads

Entrada de Dados: Ao todo 5 Strings contendo “Usuario 0X”, onde X corresponde a uma referencia a String

Saída Esperada: 5 impressões de cada String (ou de cada usuário) totalizando 25 impressões, podendo a ser ordem aleatória.

Saída de Dados: Conforme a “Saída Esperada”

Entrada de Dados:	Saída Esperada:	Saída de Dados:
Usuario 01	Usuario 01	Usuario 02
Usuario 02	Usuario 02	Usuario 02
Usuario 03	Usuario 03	Usuario 02
Usuario 04	Usuario 04	Usuario 02
Usuario 05	Usuario 05	Usuario 02
	Usuario 01	Usuario 01
	Usuario 01	Usuario 01
	Usuario 01	Usuario 01
	Usuario 02	Usuario 01
	Usuario 02	Usuario 01
	Usuario 02	Usuario 05
	Usuario 04	Usuario 05
	Usuario 04	Usuario 05
	Usuario 04	Usuario 05
	Usuario 03	Usuario 05
	Usuario 03	Usuario 04
	Usuario 03	Usuario 04
	Usuario 05	Usuario 04
	Usuario 05	Usuario 04
	Usuario 05	Usuario 04
	Usuario 01	Usuario 03
	Usuario 02	Usuario 03
	Usuario 03	Usuario 03
	Usuario 04	Usuario 03
	Usuario 05	Usuario 03

Análise:

Havendo o controle de exclusão mútua utilizando monitor de Horas e algoritmo de request/release, quando uma thread acessa a região critica seta como ocupado o recurso utilizado, as demais threads solicitam a utilização do recurso porém como está sendo utilizado as demais threads aguardam. Assim quando a thread finaliza a sua execução (uso do recurso) notifica as demais threads, as quais acordam, e a primeira que acessar o recurso realiza o mesmo processo citado.

```
public void run() {  
    for (int i = 0; i < 5; i++) {  
        monitor.request(); // Solicita o monitor para usar o recurso  
        monitor.setRecurso(nomeThread);  
        try {  
            sleep(30);  
        } catch (Exception e) {  
        }  
        monitor.usaRecurso();  
        monitor.release(); // Libera o monitor para o uso do recurso  
    }  
}
```