Universidade de São Paulo
EACH - Escola de Artes, Ciências e Humanidades

Sistemas Operacionais - Profª Gisele S. Craveiro

Gustavo Palma Figueroa - 11208068
Lorena Braghini Miranda - 11208120

Relatório EP 4

Todo o exercício foi feito no sistema operacional Windows (versão 10), utilizando a linguagem java (versão 15) para sua resolução.

Pesquisamos sobre o tema por meio do livro presente na ementa do curso, vídeos e textos da internet, para que assim fosse possível haver um maior entendimento sobre o assunto. Todas as referências a esses materiais encontram-se no final deste relatório.

O arquivo fonte comentado do programa encontra-se disponível em: https://github.com/lorenabraghini/SO/tree/main/EP4

Fizemos dois códigos: “Jantar.java” e “JantarSemaforo.java”.

Jantar.java: Este script utiliza o príncipio de métodos sincronizados nativo da linguagem java. Este recurso permite que somente uma thread acesse um determinado recurso ao mesmo tempo. Isto é possível graças aos métodos wait() e notifyall(). Os quais estão explicados ao aparecem no código. Neste script, dois filósofos conseguem comer em um ciclo e as impressões são feitas de forma correta.

JantarSemaforo.java: Este script utiliza um semáforo geral para controlar quando algum filósofo pode pegar um hashi ou não. No método *pegarHashis*, o semáforo adquire uma permissão para o filósofo pegar os hashis, mantendo o mesmo ocupado até o filósofo largar os hashis no método *returningHashis* utilizando os métodos acquire() e release() do semáforo, que são equivalentes ao lock() e unlock() da linguagem C.. No entanto, as threads parecem estar executando fora da ordem ideal, fazendo com que as impressões não saiam do modo desejado. Não obstante, o algoritmo do Jantar dos Filósofos funciona corretamente (dois filósofos comendo por vez).

Execução

Para executar o programa, basta compilar e executar o arquivo *Jantar.java* na linha de comando. Abaixo está um exemplo do programa sendo executado:

```
C:\Users\gutof\Área de Trabalho>javac Jantar.java

C:\Users\gutof\Área de Trabalho>java Jantar
```

```
O Filosofo 0 sentou-se a mesa
O Filosofo 1 sentou-se a mesa
O Filosofo 2 sentou-se a mesa
O Filosofo 3 sentou-se a mesa
O Filosofo 4 sentou-se a mesa
-----------------------------------------------------------------------------------------------------------------------
O Filosofo 2 pegou o hashi 2
O Filosofo 2 pegou o hashi 3
-----------------------------------------------------------------------------------------------------------------------
Hashis = [HASHI 0 LIVRE | HASHI 1 LIVRE | HASHI 2 OCUPADO | HASHI 3 OCUPADO | HASHI 4 LIVRE]
-----------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 2 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 3 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 4 Tentou Pegar Hashi 1 Vez(es)
-----------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Esta PENSANDO
O Filosofo 1 Esta PENSANDO
O Filosofo 2 Esta COMENDO
O Filosofo 3 Esta PENSANDO
O Filosofo 4 Esta PENSANDO
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------NOVO CICLO-----------------------------------------------------------
O Filosofo 0 pegou o hashi 0
O Filosofo 0 pegou o hashi 1
-----------------------------------------------------------------------------------------------------------------------
Hashis = [HASHI 0 OCUPADO | HASHI 1 OCUPADO | HASHI 2 OCUPADO | HASHI 3 OCUPADO | HASHI 4 LIVRE]
-----------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 2 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 3 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 4 Tentou Pegar Hashi 1 Vez(es)
-----------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Esta COMENDO
O Filosofo 1 Esta PENSANDO
O Filosofo 2 Esta COMENDO
O Filosofo 3 Esta PENSANDO
O Filosofo 4 Esta PENSANDO
```

```
C:\Users\gutof\Área de Trabalho>Javac Jantar1.java

C:\Users\gutof\Área de Trabalho>Java Jantar1
```

```
O Filosofo 0 sentou-se a mesa
O Filosofo 1 sentou-se a mesa
O Filosofo 2 sentou-se a mesa
O Filosofo 3 sentou-se a mesa
O Filosofo 4 sentou-se a mesa
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 pegou o hashi 0
O Filosofo 0 pegou o hashi 1
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 2 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 3 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 4 Tentou Pegar Hashi 1 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Esta COMENDO
O Filosofo 1 Esta PENSANDO
O Filosofo 2 Esta PENSANDO
O Filosofo 3 Esta PENSANDO
O Filosofo 4 Esta PENSANDO
```

```
-----------------------------------------------NOVO CICLO-----------------------------------------------------
O Filosofo 1 largou o hashi 1
O Filosofo 1 largou o hashi 2
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 2 pegou o hashi 2
O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 2 pegou o hashi 3
O Filosofo 2 Tentou Pegar Hashi 3 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 3 Tentou Pegar Hashi 3 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 4 Tentou Pegar Hashi 2 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 0 Esta O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
PENSANDO
O Filosofo 2 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 1 Esta O Filosofo 3 Tentou Pegar Hashi 3 Vez(es)
O Filosofo 4 largou o hashi 4
PENSANDO
O Filosofo 4 largou o hashi 0
O Filosofo 4 Tentou Pegar Hashi 2 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 2 Esta O Filosofo 0 pegou o hashi 0
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Tentou Pegar Hashi 3 Vez(es)
O Filosofo 0 pegou o hashi 1
COMENDO
-------------------------------------------------------------------------------------------------------------------
O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
O Filosofo 0 Esta O Filosofo 2 Tentou Pegar Hashi 2 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 3 Esta O Filosofo 0 Tentou Pegar Hashi 2 Vez(es)
O Filosofo 3 Tentou Pegar Hashi 3 Vez(es)
COMENDO
O Filosofo 4 Tentou Pegar Hashi 1 Vez(es)
-------------------------------------------------------------------------------------------------------------------
O Filosofo 0 Esta O Filosofo 1 Tentou Pegar Hashi 1 Vez(es)
COM FOME
```

Referências Bibliográficas

Binary Semaphore Tutorial and Example. Disponível em: <https://howtodoinjava.com/java/multi-threading/binary-semaphore-tutorial-and-example.

Class Object. Disponível em: <https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html>.

Curso de Java: Exercícios Aula 74: Semáforo com Threads. Disponível em: <https://www.youtube.com/watch?v=N-vc74o3SlQ&ab_channel=LoianeGroner>.

Fundamentos de Sistemas Operacionais. Baer, P. Gagne, G. Silberschatz A.

Java Threads: Utilizando wait, notify e notifyAll. Disponível em: <https://www.devmedia.com.br/java-threads-utilizando-wait-notify-e-notifyall/29545>.

Semaphore (Java SE 14 & JDK 14). Disponível em: <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/concurrent/Semaphore.html>.

Sincronização e Deadlock. Disponível em: <https://www.ime.usp.br/~oberlan/DCE11720/Aulas/Aula17.pdf>.

Sistemas Operacionais – Aula 13 – Problemas com Comunicação de Processos. Disponível em:

<https://www.youtube.com/watch?v=Bh3erX2Lfzk&list=PLxI8Can9yAHeK7GUEGxMsqoPRmJKwI9Jw&index=14&ab_channel=UNIVESP>.

Synchronized Vs Semaphore. Disponível em: <https://stackoverflow.com/questions/16907992/synchronized-vs-semaphore#:~:text=Semaphore%20is%20used%20to%20restrict,go%20and%20blocks%20the%20others.>.

[Tutorial] [Java] Aplicação em Thread - Jantar dos Filósofos. Disponível em: <http://papeldiario.blogspot.com/2013/07/tutorial-java-aplicacao-em-thread.html>