

Trabalho Prático de Sistemas Operativos

Licenciatura em Segurança Informática em Redes de Computadores

Janeiro de 2022

Docentes:

Prof. Altino Sampaio

Prof. Hélio Sousa

Grupo 29

Gonçalo Pedro Gil, 8200335

Luís Miguel Pires Da Costa, 8200737

1 Índice

• Introdução	
– Contextualização	2
– Objetivos do trabalho	2
– Breve descrição do trabalho	2
• Manual	
– Compilação	3
– Configuração	3
– Utilização	4
• Descrição das funcionalidades implementadas	
– Teclado	5
– Validação de valores	5
– Múltiplos clientes	5
– Botão de emergência	5
– Funcionalidades/módulos solicitados no enunciado	5
• Mecanismos de sincronização e comunicação entre módulos	5
– Threads	5
– Semáforos	5
– Classe Util	5
• Conclusão	6
• Bibliografia	6

2 Introdução

2.1 Contextualização

O presente trabalho foi realizado para a disciplina de Sistemas Operativos na licenciatura de Segurança Informática em Redes e Computadores.

2.2 Objetivos do trabalho

A fim de estudar as técnicas de multi-processamento, comunicação e sincronização entre threads (usando a linguagem Java) aprendidas nas aulas foi atribuído ao grupo o tema "Máquina de Lavagem de Carros".

2.3 Breve descrição do trabalho

Primeiramente, foram caracterizadas as classes "Admin", "Aspersores", "Main", "Moedeiro", "Rolos", "Tapete" e a "Util".

Numa segunda instância, utilizaram-se mecanismos de semáforos com threads, possibilitando, assim, a comunicação e sincronização entre as mesmas .

3 Manual

3.1 Compilação

Para compilar o presente projeto basta executar a linha de comandos abaixo apresentada.

```
$ javac src/main/java/com/mycompany/so_grupo29/Main.java
```

Listing 1: Compilar

3.2 Configuração

Para o presente projeto, foi criado um documento json denominado de "config.json" que possibilita a alteração do custo da lavagem e a duração dos tapetes, rolos, aspersores e da secagem. Este documento será lido pela class Util no início do programa de forma a importar todos os valores mencionados anteriormente.

3.3 Utilização

3.4 Executar o programa

```
$ cd path\SO_Grupo29; "JAVA_HOME=C:\\Program Files\\JAVA\\jdk-13" cmd /c "\"C:\\Program Files\\NetBeans-12.1\\netbeans\\java\\maven\\bin\\mvn.cmd\" \"-Dexec.args=\"-classpath %classpath com.mycompany.so_grupo29.Main\" -Dexec.executable=\"C:\\Program Files\\JAVA\\jdk-13\\bin\\java.exe\" -Dmaven.ext.class.path=\"C:\\Program Files\\NetBeans-12.1\\netbeans\\java\\maven-nb\\lib\\netbeans-events.jar\" -Dfile.encoding=UTF-8 process-classes org.codehaus.mojo:exec-maven-plugin:1.5.0:exec\""
```

Listing 2: Executar o programa em Windows

```
$ cd /path/SO_Grupo29; \
  JAVA_HOME=/usr/lib/jvm/jdk-17 /path/netbeans-12.5/netbeans/java/maven/bin/mvn \
  "-Dexec.args=-classpath %classpath com.mycompany.so_grupo29.Main" \
  -Dexec.executable=/usr/lib/jvm/jdk-17/bin/java -Dexec.vmArgs= -Dexec.appArgs= \
  org.codehaus.mojo:exec-maven-plugin:1.5.0:exec
```

Listing 3: Executar o programa em linux

```
$ cd /path/SO_Grupo29; \
  JAVA_HOME=/usr/lib/jvm/jdk-17 /path/netbeans-12.5/netbeans/java/maven/bin/mvn \
  "-Dexec.args=-classpath %classpath com.mycompany.so_grupo29.Main" \
  -Dexec.executable=/usr/lib/jvm/jdk-17/bin/java -Dexec.vmArgs= -Dexec.appArgs= \
  org.codehaus.mojo:exec-maven-plugin:1.5.0:exec
Running NetBeans Compile On Save execution. Phase execution is skipped and output
directories of dependency projects (with Compile on Save turned on) will be used
instead of their jar artifacts.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany:SO_Grupo29 >-----
[INFO] Building SO_Grupo29 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.5.0:exec (default-cli) @ SO_Grupo29 ---
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Sistema Aberto
Troco a receber: 3.45 euros
Contador de clientes: 1
Tapete - O Carro chegou ao ponto de lavagem
Aspersores Ativados
Aspersores Desativados
Secador Ativado
Secador Desativado
Rolos Ativados
Rolos Desativados
Tapete - O Carro chegou ao Fim
[INFO] -----
[INFO] BUILD SUCCESS
```

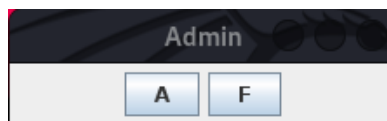
```
[INFO] -----  
[INFO] Total time: 50.066 s  
[INFO] Finished at: 2021-12-28T15:39:09Z  
[INFO] -----
```

Listing 4: Output

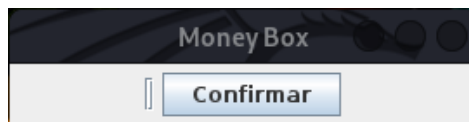
3.5 Utilização do programa

Ao executar o programa será aberta uma janela com dois botões:

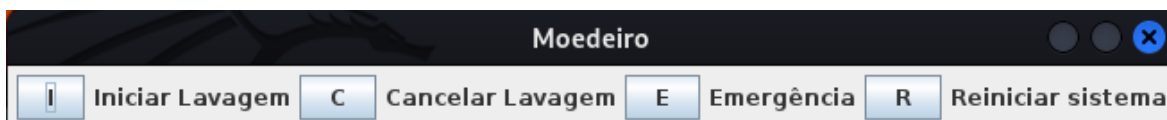
- "A": Aberto
 - Ao clicar no "A" o estado do sistema de lavagem será alterado para "aberto", possibilitando a inserção de novos clientes e a inicialização de novas lavagens.
- "F": Fechado
 - Ao clicar no "F" o estado do sistema de lavagem será alterado para "fechado", não sendo possível adicionar clientes e começar novas lavagens.



Após clicar nos botões, serão abertas duas outras janelas, uma com o objetivo de receber o dinheiro da lavagem ("MoneyBox") e outra de apresentar ao cliente todas as operações que o mesmo pode realizar ("Moedeiro").



- "I": Iniciar lavagem
 - Dá o troco ao utilizador e inicia a lavagem ou, caso já se encontre uma em execução, adiciona o veículo à lista de espera.
- "C": Cancelar lavagem
 - Cancela a lavagem e devolve o dinheiro ao utilizador.
- "E": Emergência
 - Para a lavagem a qualquer momento, interrompendo todas as tarefas em execução.
- "R": Reiniciar sistema
 - Reinicia de imediato o sistema, independentemente do estado das tarefas em execução.



4 Descrição das funcionalidades implementadas

4.1 Teclado

De forma a possibilitar uma melhor experiência de utilização do programa, foi implementada a capacidade de inicialização e de adição de novas lavagens no sistema, independentemente do seu estado de ocupação. Isto verificou-se possível através da disponibilização de um conjunto de botões (teclado) durante toda a utilização do programa. Para tal, foi utilizada a biblioteca "awt" e threads.

4.2 Validação de valores

No presente projeto, teve-se especial atenção à validação de todos os dados inseridos por parte do utilizador no sistema e de todos os valores dos botões ativados pelo cliente. Para concretizar esses objetivos, formara-se funções de validação presentes nas respetivas classes.

4.3 Múltiplos clientes

Enquanto o sistema executa uma lavagem, é possível adicionar clientes à fila de espera (estes serão atendidos pela ordem de entrada na fila - sistema FIFO). Com este fim em vista, foi criada uma lista onde todos os clientes vão sendo adicionados no decorrer do programa. Isto é possível devido à interface gráfica (teclado - anteriormente explicada).

4.4 Botão de emergência

Criou-se a funcionalidade de parar qualquer lavagem a qualquer momento, clicando no botão de "Emergência" do teclado.

4.5 Funcionalidades/módulos solicitados no enunciado

Para além de todas as funcionalidades acima mencionadas, desenvolveram-se todos os módulos solicitados no enunciado do projeto.

5 Mecanismos de sincronização e comunicação entre módulos

5.1 Threads

Procurando a capacidade de ter várias tarefas em execução em simultâneo e satisfazer os objetivos do projeto, foi programado um conjunto de threads, cada uma responsável por uma tarefa em específico, controladas por um sistema de semáforos.

5.2 Semáforos

Recorreu-se, em todo o programa, a um sistema de semáforos para controlo das threads responsáveis pelas tarefas a executar. Estes são uma estrutura de dados que controlam o acesso de aplicações a recursos, baseando-se num número inteiro, que representa o número de acessos que podem ser realizados. No nosso caso há 5 acessos.

5.3 Classe Util

Podemos afirmar que a class "Util" trata-se de uma das mais importantes classes deste projeto. Esta tem como principais funções:

- Notificar as threads
- Escrever logs
- Ler o documento de configuração json
- Receber e definir os valores dos botões.

6 Conclusão

Este trabalho cumpriu com todos os objetivos iniciais, tanto a nível dos objetivos propostos, como a nível das competências desenvolvidas pela execução dos mesmos.

O grupo teve a oportunidade de estudar aprofundadamente um ambiente mais prático da área da programação de sistemas multi-thread em java (envolvendo pesquisa, pensamento crítico e formulação). Podemos certamente afirmar que todo o trabalho e dedicação empregues neste projeto foram altamente enriquecedores em inúmeras áreas, desde o trabalho em equipa, à escrita de um relatório técnico, até o estudo autónomo por parte dos elementos do grupo.

Por fim, é seguro afirmar que acabamos este projeto mais enriquecidos e preparados para o restante percurso académico e mundo profissional.

7 Bibliografia

- <https://www.geeksforgeeks.org/wait-method-in-java-with-examples/>
- <https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example>
- <https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>
- <https://www.javatpoint.com/synchronization-in-java>