

Trabalho de Sistemas Operativos

Trabalho Prático – época normal

**Trabalho elaborado por:**

**Grupo 19**

8220169 – César Ricardo Barbosa Castelo

8220337 – Hugo Ricardo Almeida Guimarães

8220307 – Pedro Marcelo Santos Pinho

2023-2024

Índice

[Índice de Figuras 2](#_Toc155001950)

[Chave de Siglas 3](#_Toc155001951)

[Introdução 4](#_Toc155001952)

[1. Manual de compilação 5](#_Toc155001953)

[2. Manual de Utilização 6](#_Toc155001954)

[3. Funcionalidades Implementadas 9](#_Toc155001955)

[ Middleware 9](#_Toc155001956)

[ Kernel 9](#_Toc155001957)

[ CPU 9](#_Toc155001958)

[ Memória (MEM) 9](#_Toc155001959)

[4. Mecanismos de Sincronização Utilizados 9](#_Toc155001960)

[5. Funcionalidades Não implementadas 9](#_Toc155001961)

[Conclusão 10](#_Toc155001962)

# Índice de Figuras

# Chave de Siglas

|  |  |
| --- | --- |
| API | Application Program Interface |
| CPU |  |
|  |  |
|  |  |

# Introdução

Este trabalho foi realizado para o âmbito da disciplina de *Sistemas Operativos*, que funcionará como integrador dos conhecimentos adquiridos no decorrer das aulas. Ele consiste na elaboração de um programa em Java, que faça uso das técnicas de multiprocessamento, comunicação e sincronização aprendidas nas aulas para a construção de uma simulação de um sistema operativo de um satélite que gere duas unidades de computação (o CPU e a MEM) em tempo-real, recorrendo a mecanismos de sinalização e comunicação lecionados durante as aulas práticas.

A simulação proposta desempenha um papel crucial na consolidação do conhecimento teórico adquirido, proporcionando uma compreensão prática e aprofundada dos princípios dos Sistemas Operativos. Ao modelar a gestão em tempo-real de duas unidades de computação, o CPU e a MEM, o projeto visa demonstrar a aplicação eficaz de mecanismos de sinalização e comunicação, destacando a relevância prática desses conceitos.

# Manual de compilação

Para compilar o projeto, primeiramente tem de se entrar dentro da pasta da aplicação a partir do terminal e inserir o comando presente na figura 1.

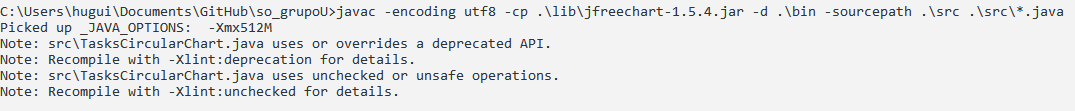


Figura - Comando para a compilação do programa

Este comando os seguintes parâmetros:

* **-encoding utf8** - Especifica a codificação de caracteres a ser usada durante a compilação.
* **-cp .\lib\jfreechart-1.5.4.jar** - Define o caminho para a biblioteca externa usada no programa o *JFreeChart*[[1]](#footnote-1).
* **-d .\bin** - Especifica a pasta de destino para os ficheiros compilados, que será na pasta *\bin*
* **-sourcepath .\src -** Especifica onde procurar os ficheiros com o código do programa, que será na pasta *\src*.
* **.\src\\*.java -** Lista de todos ficheiros com o código do programa a serem compilados, que serão todos os ficheiros com a extensão *.java*, na pasta *\src*

Quando todos os ficheiros são compilados, o terminal irá mostrar umas notas que dizem que alguns ficheiros usam versões desatualizadas de algumas API’s, mas é normal aparecer essas mensagens, pois o programa foi propositadamente construído em cima dessas versões.

Após todos os ficheiros estarem compilados, basta executar o comando presente na figura 2 para que o programa seja executado



Figura - Comando para a execução do programa

Este comando os seguintes parâmetros:

* **-cp .\lib\jfreechart-1.5.4.jar;.\bin\ -** Define para procurar as classes e as bibliotecas externas durante a execução. Neste caso, está incluído o ficheiro JAR da biblioteca *JFreeChart* e a pasta onde as classes compiladas estão localizadas.
* **.\src\application\App.java-** Especifica a classe Java a ser executada. Neste caso, vai executar a classe App localizada na pasta **\src\application\App.java**

# Manual de Utilização

Antes da inicialização do programa existe um ficheiro de configurações na dentro da pasta “*Files”*, nele constam todas as configurações que o programa necessita para poder executar como, por exemplo: os tempos de espera até à execução das próximas tarefas, ou algumas mensagens. Todos esses valores podem ser alterados pelo utilizador, visto que o ficheiro foi é guardado e lido com o *.txt*, logo, pode ser lido e reescrito facilmente.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura - Ficheiro de configuração

Quando o programa é executado irá aparecer janela presente na Figura 4. Ela contém um botão de ligar e desligar, que irá ligar e desligar o satélite, também tem uma secção de tarefas onde é possível gerar uma tarefa customizada, com: um nome; mensagem; memória que irá necessitar; tempo expectado para a tarefa e a prioridade da mesma. Para além disso, também tem um botão que efetua um teste de stress, que irá executar várias tarefas ao mesmo tempo, tarefas essas com valores aleatórios, que podem ser definidos no pelo ficheiro de configuração, este teste serve para testar como é que o CPU opera sobre muita pressão. Lembrando que estas funcionalidades funcionam apenas quando sistema operativo, está ligado.

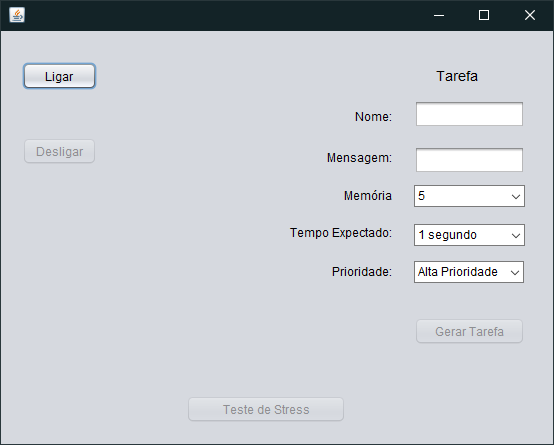


Figura - Janela inicial do programa

Quando o satélite é ligado, irão aparecer 3 novas janelas. Uma dessas janelas é uma consola que representa o monitor da Terra, onde irão ser exibidas as tarefas enviadas para o satélite, bem como as respostas recebidas. As outras janelas são dois gráficos que têm quase os mesmos valores, mas exibidos de maneiras diferentes. Um deles é um gráfico circular que tem a quantidade de tarefas que estão em espera, em execução e as que já finalizaram. O segundo, é um gráfico de barras que contém apenas as tarefas que estão em execução bem como as tarefas que estão em espera na CPU. Este segundo gráfico, mesmo contendo os mesmos valores que o primeiro, é extremamente útil, pois como não tem as tarefas finalizadas, e como é um gráfico de barras, temos uma maior noção da quantidade de tarefa que estão a ser executadas em função das que estão em espera.

Uma imagem com captura de ecrã, computador

Descrição gerada automaticamente

Figura - Consola com as tarefas recebidas

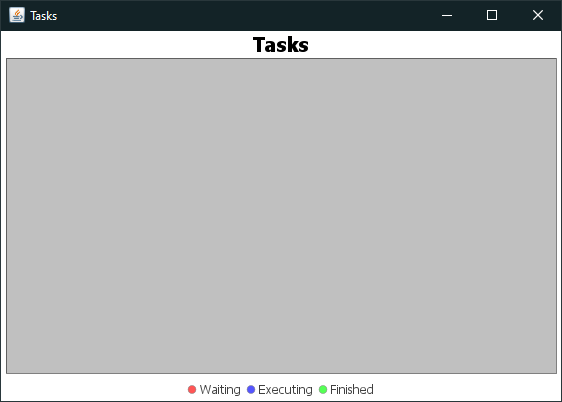


Figura - Gráfico circular das tarefas

Uma imagem com texto, captura de ecrã, ecrã, software

Descrição gerada automaticamente

Figura - Gráfico de barras das tarefas

Caso se queira saber o que aconteceu durante a execução do programa, com ele já encerrado, basta ir à pasta “*.\Files*”, lá contém dois ficheiros *“.txt”* que registam os eventos entre a Terra e o satélite. O ficheiro da Terra contém as tarefas que foram agendadas, e as mensagens que foram recebidas do satélite, enquanto o satélite tem as tarefas que lhe foram enviadas e as respostas que ele deu às mesmas, contém também a data e hora que ele ligou e encerrou.

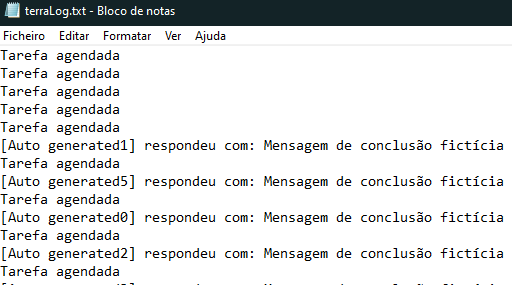


Figura – Ficheiro com os registos da terra

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Figura – Ficheiro com os registos do satélite

Se calhar é melhor que quando encerramos as threads em espera não são executadas, se calhar é melhor colocar aí os gráficos em execução

# Funcionalidades Implementadas

### Middleware

### Kernel

### CPU

### Memória (MEM)

# Mecanismos de Sincronização Utilizados

# Funcionalidades Não implementadas

# Conclusão

Achamos que a realização deste trabalho foi bem concebida tendo em conta o que nos foi proposto e também as nossas próprias exigências para o trabalho, entre elas, a constante busca por um programa funcional e coeso com aquilo que um sistema operativo é.

Ao longo da realização deste trabalho, confrontamo-nos com diversos desafios, sendo os principais: a identificação de situações de competição, que exigiram uma análise cuidadosa para compreender a natureza dos problemas e, subsequentemente, implementar soluções eficazes; a identificação do método mais apropriado de sinalização, considerando as complexidades inerentes ao problema em questão. Esta fase do projeto proporcionou uma oportunidade única de aplicar os conceitos teóricos aprendidos em sala de aula a um contexto prático e desafiador.

Em suma, este projeto desempenhou um papel crucial em nosso desenvolvimento acadêmico e profissional. A abordagem prática contribuiu para sedimentar nosso conhecimento. A resolução de desafios específicos, como a competição por recursos e a escolha de métodos de sinalização, fortaleceu nossa capacidade de análise e solução de problemas, competências essenciais na área de Sistemas Operativos.

1. uma biblioteca em Java que facilita a exibição de gráficos. [↑](#footnote-ref-1)