

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA POLITÉCNICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
SISTEMAS EM TEMPO REAL**

EXERCÍCIOS PARA ENTREGA – M1

por

Fabio Ivo Pereira de Oliveira Junior

Itajaí (SC), Agosto de 2024

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA POLITÉCNICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
SISTEMAS EM TEMPO REAL**

EXERCÍCIOS PARA ENTREGA – M1

por

Fabio Ivo Pereira de Oliveira Junior

Relatório apresentado como requisito parcial da disciplina Sistemas em Tempo Real do Curso de Engenharia de Computação para análise e aprovação.

Professor Responsável: Felipe Viel

Itajaí (SC), Agosto de 2024

Link para o repositório GitHub com os códigos desenvolvidos para a solução dos problemas propostos:

<https://github.com/FIPOJ/STR-EntregaM1.1>

- **Pergunta 1: Para o código de exemplo, compile o mesmo e o execute para que o laço de repetição tenha apenas uma execução. Qual o tempo de execução informado ao fim da execução?**

R:

Tempo de execução (segundos): 0.007000

- **Pergunta 2: Para o mesmo código, implemente um laço de repetição com 100 interações para as 5 tarefas. Quanto é o tempo de execução de total para o software?**

R:

Tempo de execução total para 100 iterações (segundos): 0.308000

- **Pergunta 3: Repita os dois exercícios anteriores para os diferentes níveis de otimização dos códigos (O0, O1, O2, O3).**

R:

O0: Tempo de execução (única iteração): 0.006 segundos, Tempo de execução total (100 iterações): 0.357 segundos

O1: Tempo de execução (única iteração): 0.006 segundos, Tempo de execução total (100 iterações): 0.373 segundos

O2: Tempo de execução (única iteração): 0.006 segundos, Tempo de execução total (100 iterações): 0.322 segundos

O3: Tempo de execução (única iteração): 0.008 segundos, Tempo de execução total (100 iterações): 0.336 segundos

- **Pergunta 4: Repita a pergunta 2 agora obtendo o tempo para executar cada tarefa. Utilize as bibliotecas `time.h` e `intrin.h` para obter os valores de tempo de execução.**

R:

Func1: Tempo de execução (segundos): 0.000000

Func2: Tempo de execução (segundos): 0.000000

Func3: Tempo de execução (segundos): 0.000000

Func4: Tempo de execução (segundos): 0.126000

Func5: Tempo de execução (segundos): 0.206000

- **Pergunta 5: Escolha uma das bibliotecas citadas anteriormente para avaliar o conceito de tempo de execução pela média. Obtenha 100 amostras da execução e calcule a média e exiba. Qual a função mais lenta e qual a função mais rápida.**

R:

Func1: Média: 0.000000, Melhor: 0.000000, Pior: 0.000000

Func2: Média: 0.000000, Melhor: 0.000000, Pior: 0.000000

Func3: Média: 0.000000, Melhor: 0.000000, Pior: 0.000000

Func4: Média: 0.001710, Melhor: 0.000000, Pior: 0.008000

Func5: Média: 0.002150, Melhor: 0.001000, Pior: 0.007000

Função mais rápida: Func1

Função mais lenta: Func5

- **Pergunta 6: Implemente uma solução para um problema usando semáforos, onde cada três tarefas 'calcula' permitem que exatamente uma tarefa 'mostra' execute.**

R:

O código foi desenvolvido usando semáforos para controlar a execução das tarefas 'calcula' e 'mostra'. Resultado da execução: As tarefas 'mostra' são executadas apenas após três execuções da tarefa 'calcula'.

Resultado no Prompt:

```
C:\Users\fipoc\OneDrive - UNIVALI\Documentos\Faculdade\2024.2\Sistemas em Tempo Real\M1\Exercícios em  
regar 15.08>6.exe  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'mostra' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'calcula' executando...  
Tarefa 'mostra' executando...  
Tarefa 'mostra' executando...
```

- Pergunta 7: Um simulador de tráfego urbano foi implementado. Implemente o mutex utilizando as funções da biblioteca de pthreads para gerenciar a passagem dos carros.**

R:

O código foi desenvolvido para simular o tráfego urbano usando mutexes para evitar deadlock e garantir que os carros passem pelo cruzamento de forma correta. Resultado da execução: Carros passaram corretamente nos sentidos 1 e -1 conforme esperado.

Resultado no prompt:

```
C:\Users\fipoc\OneDrive - UNIVALI\Documentos\Faculdade\2024.2\Sistemas em Tempo Real\M1\Exercicios em  
regar 15.08>7.exe  
Carro passando no sentido 1  
Carro passando no sentido 1  
Carro passando no sentido 1  
Carro passando no sentido 1  
Carro passando no sentido 1  
Carro passando no sentido -1  
Carro passando no sentido -1  
Carro passando no sentido -1  
Carro passando no sentido -1  
Carro passando no sentido -1
```

REFERÊNCIAS

KNS, Yui. *intrin: Collection of intrinsics for various CPUs*. Disponível em: <https://github.com/yuikns/intrin/tree/master>. Acesso em: 14 ago. 2024.

VIEL, Filipe. *VielF/SO-Codes: Codes for operating systems classes*. Disponível em: <https://github.com/VielF/SO-Codes>, Nas pastas de processadores. Acesso em: 14 ago. 2024.

IBM. *pthread.h file (Threads Programming Interfaces)*. Disponível em: <https://www.ibm.com/docs/pt-br/aix/7.3?topic=files-pthreadh-file>. Acesso em: 14 ago. 2024.

KERNIGHAN, Brian W.; RITCHIE, Dennis M. *The C Programming Language*. 2nd ed. Upper Saddle River: Prentice Hall, 1988. Disponível em: <https://a.co/d/7K2DH85>. Acesso em: 14 ago. 2024.

TUTORIALSPPOINT. *C Standard Library - time.h*. Disponível em: https://www.tutorialspoint.com/c_standard_library/time_h.htm. Acesso em: 14 ago. 2024.