

Trabalho: Bilheteria de ônibus

1. Descrição do problema

Sistema de bilheteria eletrônica precisam lidar constantemente com múltiplos passageiros tentando comprar passagens nos mesmos horários e poltronas. Tais sistemas precisam implantar salvaguardas a fim de evitar que dois passageiros comprem passagens para as mesmas cadeira.

Seu trabalho será implementar uma simulação de um sistema de comprar de bilhetes online.

2. Especificação

O seu trabalho deverá funcionar da seguinte maneira:

2.1. O programa deverá ser executável pelo terminal Linux

As instruções de compilação estão na Seção 4.

2.2. O programa deverá receber como parâmetro um número inteiro indicando a quantidade de passageiros a serem gerados.

2.3. Deverá haver uma thread chamada “GeradorPassageiros”

Essa thread deverá gerar continuamente passageiros, os quais deverão escolher um **horário** de ônibus e uma **poltrona**

A poltrona deverá ser escolhida aleatoriamente dentre as poltronas disponíveis para um ônibus em um dado horário.

2.4. Para verificar quais poltronas estão disponíveis, deverá ser chamado o método *verificar_poltronas_disponíveis(horário)*.

O método/procedimento deverá retornar as poltronas disponíveis no ônibus do dado horário.

Ministério da Educação

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

UNED Nova Friburgo

Bacharelado em Sistemas da Informação

Disciplina de Programação Paralela e Concorrente

Professor Bruno Policarpo Toledo Freitas

- 2.5. O passageiro deverá chamar um procedimento/método chamado *reservar_passagem(horário, poltrona)* para tentar reservar uma passagem.

O procedimento deverá retornar um booleano indicando se a reserva foi feita com sucesso ou não.

O resto da implementação do procedimento/método é de livre escolha do aluno, incluindo colocar mais parâmetros ao procedimento, por exemplo.

- 2.6. Cada passageiro deverá ter associado um número inteiro de 0 até n - o total de passageiros a serem gerados pela simulação.

- 2.7. Existem ônibus de hora em hora a partir das 7 da manhã até 21h (ou seja, 13 horários disponíveis). Cada ônibus possui 40 lugares.

- 2.8. Quando um passageiro conseguir reservar uma passagem, deverá ser impresso na tela . “PASSAGEIRO n RESERVOU A POLTRONA m DO ÔNIBUS PARTINDO AS x HORAS”

- 2.9. A simulação deverá parar automaticamente após os n passageiros reservarem as passagens

- 2.10. Os passageiros somente tentarão comprar passagem 1 vez

Ministério da Educação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
UNED Nova Friburgo
Bacharelado em Sistemas da Informação
Disciplina de Programação Paralela e Concorrente
Professor Bruno Policarpo Toledo Freitas

3. **Entrega**

Deverá ser entregue apenas um arquivo com o código fonte de sua solução. Poderão ser utilizadas as linguagens C, C++, ou Java, ou outras linguagens, desde que previamente informadas a fim de determinar como será feita a compilação e execução do programa.

O nome do arquivo deverá obrigatoriamente ser *bilheteria* seguido da extensão da linguagem utilizada. Por exemplo: *bilheteria.c* ou *bilheteria.java*

Os códigos-fonte serão compilados e testados pelo terminal do Linux da seguinte maneira:

- C
gcc bilheteria.c -o bilheteria -pthread
chmod +x bilheteria
./bilheteria 1000 (para uma simulação com 1000 passageiros)
- C++
g++ bilheteria.cpp -o bilheteria -pthread
chmod +x bilheteria
./bilheteria 1000 (para uma simulação com 1000 passageiros)
- Java
javac bilheteria.java
java bilheteria.class 1000 (para uma simulação com 1000 passageiros)

NOTA: Se você deseja usar outra linguagem e seus respectivos mecanismos de threads e sincronização, entre em contato comigo para definir como será feita a compilação e execução do trabalho

Ministério da Educação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
UNED Nova Friburgo
Bacharelado em Sistemas da Informação
Disciplina de Programação Paralela e Concorrente
Professor Bruno Policarpo Toledo Freitas

4. **Avaliação**

A avaliação do trabalho será conforme a Tabela 1. Cada critério poderá ser avaliado como:

- Excelente: 100% da nota do critério
 - Sem erros
- Bom: 75% da nota do critério
 - 1 erro
- Razoável: 50% da nota do critério
 - 2 erros
- Ruim: 25% da nota do critério
 - 3 ou mais erros
- Não fez: 0%

Tabela 1: Critérios de avaliação

Critério	Valor
Compila conforme especificação	10 %
Programa consegue terminar normalmente (não necessariamente corretamente)	10 %
Threads auxiliares	10 %
Procedimento “verificar_poltronas_disponiveis”	30 %
Procedimento “reservar_poltrona”	30 %
Programa sem Condições de Corrida ou Deadlocks	10 %