

Ingressos para a “Bróduei”

A casa de espetáculos “Bróduei” encomendou um programa para a venda de ingressos através de bilheteria eletrônica. Os lugares de cada espetáculo da “Bróduei” são numerados sequencialmente, de 1 à MAX. O sistema permite que múltiplos espetáculos possam ter seus ingressos vendidos em simultâneo na bilheteria eletrônica.

O programa deve funcionar da seguinte forma. Cada cliente deseja comprar um ingresso de um espetáculo em específico. O cliente solicita um lugar naquele espetáculo (dentro os disponíveis naquele espetáculo) e efetua o pagamento. O pagamento é feito por cartão de crédito, e pode ser aprovado pela operadora ou não (a aprovação ou não da compra consiste em uma escolha arbitrária do programa). Caso o pagamento não seja aprovado, o cliente desiste da compra, perde o direito ao lugar escolhido, e este lugar volta a ficar disponível. Tanto o procedimento de solicitação de ingresso como o de aprovação de pagamento implicam em esperas (*sleeps* com valores aleatórios). Considere que a autorização de pagamento está sujeita a falhas de conexão e longos tempos de espera. Assim, um cliente que aguarda a autorização de pagamento para um ingresso de um determinado espetáculo não pode prejudicar a venda de outros ingressos para o espetáculo em questão, nem para os demais.

O programa não pode vender mais de uma vez o mesmo lugar de um espetáculo, nem pode marcar como ocupados os lugares escolhidos por clientes que não conseguiram efetuar o pagamento. Em um mesmo espetáculo, apenas um ingresso pode ser comprado por vez. Entre espetáculos diferentes, ingressos podem ser vendidos na mesma compra. Caso os ingressos de um determinado espetáculo esgotem, os clientes que ficarem sem ingressos podem aceitar a sugestão do sistema em comprar um ingresso para um outro espetáculo que tenha lugares disponíveis. Essa possibilidade consiste em uma escolha arbitrária do programa, dentre os espetáculos disponíveis, para cada cliente que ficar sem ingresso.

Para evitar conflitos, deadlocks e condições de disputa, **o programa deve ser implementado usando semáforos**, ou seja, primitivas `sem_init`, `sem_wait`, `sem_post`,... da biblioteca Pthread - linguagem C, ambiente Linux. **Não devem ser utilizadas as funções `pthread_lock` e `pthread_unlock`!!! Trabalhos que não seguirem esta determinação serão desconsiderados.**

Quando inicializado, o programa deve ler os seguintes valores inteiros de um arquivo de entrada:

- Nome do espetáculo *i*;
- MAX ingressos do espetáculo *i*;
- Valor do ingresso (R\$) do espetáculo *i*;
- Número de clientes a serem gerados para a compra de ingressos no espetáculo *i*.

O formato do arquivo de entrada (.txt) consiste em N linhas (1 linha por espetáculo), cada linha contendo os quatro valores acima, separados por um pipe (|). Um exemplo de arquivo de entrada segue abaixo:

```
$ cat input.txt
```

```
RockInRio|5|100|10  
SuperBowl|10|300|10  
Tomorrowland|15|150|20
```

O andamento do programa, diferenciando espetáculo e clientes por um identificador único, deve ser **exibido na tela (trace de execução) e salvo em um arquivo texto** para posterior análise (uma mensagem por linha). Ao final, o programa deve exibir, **por espetáculo**:

- o número de ingressos vendidos dentre os disponíveis daquele espetáculo;
- o valor total arrecadado com a venda dos ingressos daquele espetáculo;
- o número de ingressos que não foram vendidos (se aplicável);
- o número de clientes que ficaram sem ingresso dentre o número de clientes gerados para aquele espetáculo em questão, discriminados pelo motivo, sendo eles:
 - falha de pagamento;
 - recusa de comprar ingresso de outro espetáculo após os ingressos de seu espetáculo terem esgotado;
 - indisponibilidade total de ingressos, ou seja, todos os lugares da Bróduei em todos os espetáculo já foram vendidos.

Seguem exemplos de possíveis mensagens a serem exibidas no trace de execução (no mínimo):

```
$ cat output.txt
```

```
> cliente 1 está escolhendo lugar no espetáculo 'SuperBowl'  
> cliente 2 está escolhendo lugar no espetáculo 'RockInRio'  
> cliente 2 escolheu o lugar 3 no espetáculo 'RockInRio' e está aguardando  
autorização de pagamento  
> cliente 3 está escolhendo lugar no espetáculo 'Tomorrowland'  
> cliente 4 está escolhendo lugar no espetáculo 'RockInRio'  
> cliente 1 escolheu o lugar 7 no espetáculo 'SuperBowl' e está aguardando  
autorização de pagamento  
> cliente 3 escolheu o lugar 1 no espetáculo 'Tomorrowland' e está aguardando  
autorização de pagamento  
< cliente 1 teve seu pagamento aprovado e fez a compra do lugar 7 do espetáculo  
'SuperBowl'  
> cliente 5 está escolhendo lugar no espetáculo 'Tomorrowland'
```

```
> cliente 4 escolheu o lugar 4 no espetáculo 'RockInRio' e está aguardando
autorização de pagamento
< pagamento do cliente 3 no espetáculo 'Tomorrowland' não foi autorizado
> cliente 6 está escolhendo lugar no espetáculo 'SuperBowl'
< cliente 2 teve seu pagamento aprovado e fez a compra do lugar 3 do espetáculo
'RockInRio'
...
> cliente 20 está escolhendo lugar no espetáculo 'RockInRio'
> cliente 21 está escolhendo lugar no espetáculo 'RockInRio'
...
> 'RockInRio' não tem lugares disponíveis. Cliente 21 recusou a proposta de compra
de ingresso em outro espetáculo
> 'RockInRio' não tem lugares disponíveis. Cliente 20 aceitou a proposta de compra
do ingresso no espetáculo 'Tomorrowland' e está aguardando autorização de
pagamento do lugar 9
...

***** Espetáculo "RockInRio"; 5/5 ingressos vendidos (R$ 500); 3/10 clientes sem
ingresso: 2 por falha de pagamento; 1 por recusa de compra de ingresso em outro
espetáculo
***** Espetáculo "SuperBowl"; 10/10 ingressos vendidos (R$ 3000); 0/10 clientes sem
ingresso
***** Espetáculo "Tomorrowland"; 15/15 ingressos vendidos (R$ 2250); 7/20 clientes
sem ingresso: 4 por falha de pagamento, 3 por indisponibilidade de ingressos
```

Informações Importantes:

Além da especificação anterior, que descreve o programa a ser implementado, deve-se considerar:

- **A realização do trabalho é individual.**
- Implementação:
 - O programa deve ser implementado de forma modular, ou seja, com o uso de funções.
 - **Não** devem ser inseridos comentários no código.
 - O código deve estar devidamente indentado.
 - **O programa deve explorar a concorrência entre as threads, de forma a não permitir a sequencialidade na venda dos ingressos.**
 - O programa deve ser implementado usando **semáforos**.

- Apresentação:
 - Todos os trabalhos devem ser apresentados.
 - As apresentações serão agendadas com a professora. Os alunos receberão o link (**Google Meet**) para o horário combinado e devem manter a **câmera e microfone abertos no decorrer da apresentação.**
 - **As apresentações de trabalho serão gravadas.**
 - Os alunos que tiverem dificuldades em atender essas orientações devem entrar em contato com a professora **previamente**.

- Avaliação:
 - Durante a apresentação, o aluno será questionado sobre a implementação do trabalho e deve estar ciente de todos os seus detalhes.
 - Ao longo da apresentação, poderão ser solicitadas pequenas modificações no código do aluno, as quais deverão ser realizadas naquele momento.
 - **Trabalhos não apresentados serão desconsiderados e o aluno obterá nota zero.**
 - Serão utilizadas ferramentas de análise de similaridade de código-fonte (por exemplo MOSS, JPlag). **Em caso de cópia entre trabalhos, será atribuída nota zero.**
 - Aconselha-se que os alunos não troquem informações entre si, de modo a evitar coincidências entre os códigos, as quais serão acusadas pelas ferramentas de análise. **Quaisquer similaridades terão impacto na avaliação do trabalho do(s) aluno(s).**
 - A utilização de código de outros trabalhos que não sejam de autoria do aluno (deste e de outros semestres) é considerada cópia/plágio.

- Entrega:
 - Data: **02/08/2020.**
 - Formato: arquivo.c, o qual deve ser postado no Moodle.