Sistema de Reserva de Assento

Acadêmico: Roberto Haruo Honda Junior
Professor: Marcos Paulo Moro

[1](UFGD) UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
Dourados – MS - -Brasil
(FACET) Faculdade de Ciências Exatas e Tecnologia

E-mail: robertohonda96@gmail.com

Resumo. *Este relatório descreve a implementação do Sistema de Reserva de Assentos e o Programa Auxiliar. No Sistema de Reserva de Assentos, threads representam passageiros, uma thread é responsável por gerar um Arquivo de Log e a thread principal (main) é responsável por criar todas as threads. O Programa Auxiliar é responsável por executar as operações do sistema de forma sequencial e verificar o Arquivo de Log comparando com os resultados das operações contidas no mesmo.*

1. Introdução

O Sistema de Reserva de Assento simula solicitações de passageiros (thread passageiro) a um sistema (thread Arquivo de Log) com as operações:

1: visualizar assentos.

2: alocar assento livre.

3: alocar assento dado.

4: liberar assento dado.

Através do Programa auxiliar é feita a verificação do Arquivo de Log gerado pelo sistema.

2. Sistema de Reserva de Assento com threads

2.1. Estruturas e variáveis principais

Foram utilizadas 4 estruturas de dados.

t_Assento é a estrutura que representa o assento, contendo os campos: um inteiro para representa o identificador do passageiro, um booleano para representar o estado do assento(ocupado ou disponível) e um inteiro como identificador do assento.

t_Assentos é a estrutura que representa um conjunto de assentos (vetor de assentos), contendo os campos: um vetor de assentos e inteiro representando a quantidade de assentos.

Operacao é a estrutura utilizadas para identificar a operação, o identificador do passageiro e o identificador do assento, necessário para escrever no Arquivo de Log.

Passageiro é a estrutura que representa o passageiro, contendo os campos: um inteiro para identificar o passageiro e um booleano representando o estado do passageiro

(sentado ou levantado).

As variáveis comuns a todas as threads são:

operacaoMutex: seção crítica para realizar as operações.

escrever e pronto: eventos utilizados entre a thread passageiro e a thread de Arquivo de Log,

assentos: estrutura t_Assentos.

threadLog: thread responsável por gerar Arquivo de Log.

filaOperacoes: uma fila de Operações para serem escritas no arquivo de Log.

flag: variável de controle para terminar a thread de Log.

nomeArquivo: nome do arquivo de Log.

2.2. Funções e Procedimentos

Foram utilizadas 6 procedimentos e 5 funções.

Procedimento inicializar: procedimento para iniciar as variáveis.

Procedimento threadRegistrarLog: procedimento da thread responsável por gerar o Arquivo de Log, os eventos escrever e pronto são utilizados para que as threads passageiro possam solicitar a escrita no arquivo de Log, onde é colocado os parâmetros para realizar a escrita na fila de operações.

Procedimento passageiro: procedimento das threads passageiro onde é gerado, aleatoriamente, uma quantidade de operações a ser realizadas e as operações, é utilizada a seção crítica para todas operações, pois, mesmo a operação 1 (visualizarAssentos) não pode ser executada, concorrentemente, com as outras, devido ao conjunto de assentos comum a todos passageiros e as características das operações.

Procedimento filaLog: procedimento executado para solicitar a escrita no Log à thread de Arquivo de Log.

Procedimento visualizarAssentos: procedimento executado para realizar a operação 1.

Procedimento finalizar: procedimento para finalizar as variáveis do sistema.

Função verificarArgumentos: função para verificar os argumentos passados para o programa, retorna uma string vazia se os argumentos estão certos ou uma string de erro se não.

Função escreverLog: função que manipula o arquivo para escrita, retorna -1 se ocorrer erro e 0 se não ocorrer.

Função alocarAssentoLivre: função executada para realizar a operação 2 escolhendo, aleatoriamente, um assento disponível, retorna 1 se a operação foi realizada com sucesso e 0 caso contrário (passageiro já sentado ou não há assentos vazios).

Função alocarAssentoDado: função executada para realizar a operação 3, tenta alocar um assento dado, retorna 1 se a operação foi realizada com sucesso e 0 se não (passageiro já sentado ou assento ocupado por outro passageiro).

Função liberarAssentoDado: função executada para realizar a operação 4, tenta liberar um assento dado, retorna 1 se a operação foi realizada com sucesso e 0 se não (não é o passageiro que está no assento ou o passageiro não está sentado).

2.3. Fluxo do Sistema

A thread principal (main) cria todas a thread de Arquivo de Log, as threads passageiro e espera todas as threads terminarem. Cada thread passageiro irá realizar uma quantidade de operações aleatórias, solicitando para cada operação uma escrita no Arquivo de Log á thread de Arquivo de Log, que guarda a solicitação em uma fila e libera a thread passageiro. Como nenhuma das 4 operações pode ser realizada concorrentemente, as operações são realizadas em uma seção crítica, quando todas as threads passageiro terminarem e a thread de Arquivo de Log terminar todas escritas, então a thread principal desperta e finaliza o sistema.

3. Programa Auxiliar

3.1. Estruturas

Foram utilizadas 3 estruturas de dados.

t_Assento é a estrutura que representa o assento, contendo os campos: um inteiro para representa o identificador do passageiro, um booleano para representar o estado do assento(ocupado ou disponível) e um inteiro como identificador do assento.

t_Assentos é a estrutura que representa um conjunto de assentos (vetor de assentos), contendo os campos: um vetor de assentos e inteiro representando a quantidade de assentos.

Passageiro é a estrutura que representa o passageiro, contendo os campos: um inteiro para identificar o passageiro e um booleano representando o estado do passageiro (sentado ou levantado).

3.2. Funções e Procedimentos

Foram utilizadas 2 procedimentos e 5 funções.

Procedimento inicializar: procedimento para iniciar as variáveis.

Procedimento visualizarAssentos: procedimento executado para realizar a operação 1.

Função verificarArgumentos: função para verificar os argumentos passados para o programa, retorna uma string vazia se os argumentos estão certos ou uma string de erro se não.

Função revisar: função responsável por verificar se as informações gravadas no Arquivo de Log estão corretas, retorna 0 se não houver erros, retorna -1 se houver erro em abrir o arquivo ou retorna linha do erro caso houver erro.

Função montarString: função utilizada na verificação para montar string.

Função alocarAssentoDado: função executada para realizar a operação 2 e 3, pois na verificação possuem o mesmo comportamento. Tenta alocar um assento dado, retorna

1 se a operação foi realizada com sucesso e 0 se não (passageiro já sentado ou assento ocupado por outro passageiro).

Função liberarAssentoDado: função executada para realizar a operação 4, tenta liberar um assento dado, retorna 1 se a operação foi realizada com sucesso e 0 se não (não é o passageiro que está no assento ou o passageiro não está sentado).

3.3. Fluxo do Programa

O programa auxiliar verifica os parâmetros passados, inicializa suas variáveis e executa a função verificar e, por fim, imprime uma saída relatando se o Arquivo de Log está correto ou incorreto e a linha de erro se houver.

4. Testes e Resultados

Foram realizados diversos testes com a quantidade de até 500 assentos, considerando que a quantidade de assentos e passageiros é a mesma. Os resultados do Sistema de Reserva de Assento foram todos bem-sucedidos.

5. Conclusão

A implementação do Sistema de Reserva de Assento possibilitou maior compreensão do comportamento das threads e das bibliotecas de threads, contribuindo para o conhecimento da disciplina de Sistemas Operacionais.

6. Materiais de estudo utilizados

http://www.cplusplus.com/

https://msdn.microsoft.com/en-us/library/windows/desktop/ms684847