

Relatorio Trabalho Sistema Operacionais

IFMG - Campus Formiga

Guilherme Cardoso Silva, Renata Caroline Cunha

¹ Sistemas Operacionais – Instituto Federal de Minas Gerais (IFMG)
Caixa Postal 35570-000 – Formiga – MG – Brasil

godgcs@gmail.com.br, renatinhacunha17@hotmail.com

1. Estrutura Geral

Os funcionarios foram implementados para executarem as funções na seguinte ordem:

- Primeiramente os funcionários devem esperar para que possam estacionar. Se nenhum funcionário estiver esperando, ele poderá estacionar. Caso contrário, ele deve esperar ser sinalizado para estacionar.
- Quando um funcionário é sinalizado para estacionar ele ocupa a vaga por 1 segundo.
- Em seguida o funcionario libera a vaga, ao liberar ele verifica entre todos os funcionarios que estão esperando qual possui maior prioridade e o sinaliza. Caso ao passar a vaga seja detectado uma espera circular entre os funcionarios, uma variavel inteira recebera valor 1 para sinalizar que ocorreu um deadklock. Quando isto ocorrer o diretor poderá sortear um funcionario para estacionar.
- Se o contador de estacionamento do funcionario tenha chegado a 0 ele será liberado, caso contrario ele deverá esperar de 3 a 5 segundos para voltar a competir pela vaga.

O Diretor atua da seguinte forma:

- De 5 em 5 segundos é feita uma verificação se a variavel de deadlock foi sinalizada.
- Caso tenha ocorrido um deadklock o Diretor irá sortear um dos 3 grupos e sinaliza-lo, aquele funcionario do grupo sorteado que chegou primeiro poderá estacionar.
- Depois de ter sorteado um funcionario em caso de deadklock o Diretor irá dormir por 5 segundos.
- Se não foi identificado um deadlock o Diretor também irá dormir por 5 segundos.
- O Diretor irá executar sua função até que todos os funcionarios finalizem.

2. Decisões de Projeto

Foi decidido que cada grupo(funcionario e técnico administrativo) teria uma variável de condição. Como são três grupos, existem três variaveis de condição, onde ambos parceiros podem dar wait para aguardar sua vez. Como a fila de espera do semaforo é implementada como uma FIFO, o funcionario que solicitar a vaga primeiro, será o primeiro a ser atendido.

Quando um funcionario solicitar a vaga, e ele for o unico que quer o recurso, ele poderá estacionar sem esperar, caso contrario será barrado até que sua variavel de condição for sinalizada.

A sinalização para quem será o próximo a estacionar ocorre no fim do estacionamento, onde na função *PassaVaga* serão verificados entre todos os grupos de funcionários que querem a vaga, qual tem maior prioridade, este terá sua variável de condição sinalizada e o primeiro do grupo a solicitar poderá estacionar.

Caso todos os grupos demonstrem o interesse em estacionar, significa que aconteceu um deadlock e não será possível passar a vaga, então através de uma variável chamada *DeadLock* que tem seu valor alterado para 1, o diretor sorteia um dos 3 grupos para estacionar.

É possível verificar quais os funcionários que estão esperando para estacionar através de um vetor de inteiros chamado *InfoThreads* que armazena qual o seu estado, podendo ser:

- *EmCasa*: Significa que não demonstrou interesse na vaga.
- *Esperando*: Significa que está esperando pela vaga.
- *Terminou*: Significa que já estacionou a quantidade de vezes passadas por parâmetro.

Para verificação é armazenado em uma variável chamada *UltimoEstacionar* qual é o funcionário que acabou de estacionar e está passando a vaga. Na função *PassaVaga*, caso não seja detectado um deadlock, é verificado se o funcionário com maior prioridade não é o parceiro do último a estacionar e existe mais alguém esperando para estacionar, se esta verificação for verdadeira a vaga é passada para o próximo com maior prioridade para evitar a troca da vaga entre os parceiros.

Foram criadas três mutex:

- *mutexGeral*: é utilizada para proteger a sessão crítica de todo o processo de estacionamento
- *mutexDeadLock*: é utilizada para proteger a variável *DeadkLock* que pode ser acessada tanto pelo Diretor que verifica se ocorreu um deadlock ou pelos funcionários que ao tentar passar a vaga detecta um *DeadkLock* e ao serem liberados pelo Diretor mostram uma mensagem de liberação que é verificada pela variável *DeadLockFuncionario* que também é protegida por esta mesma mutex.
- *mutexEstado*: é utilizada para proteger o vetor de estados de cada funcionário, para que ninguém acesse a variável de um funcionário enquanto ele esteja alterando seu Estado, por exemplo o Diretor verifica se todos os funcionários já terminaram para que possa terminar também, enquanto isto algum funcionário pode estar alterando seu estado e pode gerar troca de contexto.