

Simulador de espera bancário

Integrantes

- Carlos André Feldmann Júnior (17200107)

Problema

Criar um sistema de espera de atendimento bancário utilizando threads e mecanismos de sincronização. Existe um número fixo de cadeiras onde os clientes podem sentar para esperar o atendimento do gerente. Ao ser atendido o cliente vai para sala do gerente liberando a cadeira e gastando X tempo do gerente. Caso algum cliente chegue e não haja cadeiras disponíveis o cliente vai embora!

Modelagem

Uso de threads

- O Gerente tem uma thread própria constante responsável por atender os clientes(sleep) e chamar o próximo da fila.
- Quando um cliente senta em uma cadeira, é criada uma nova thread para monitorar se é a vez dele ser atendido, porém após ele se levantar da cadeira a thread é destruída e é necessário criá-la novamente quando o próximo cliente sentar nela.

Structs

Foi criado uma struct para:

- Cliente, contendo o nome e o tempo que vai ser necessário para atendê-lo.
- Cadeira
 - Número da cadeira
 - Cliente sentado
 - Mutex lock - Lock usado para evitar escrita paralela por duas threads.
 - Semáforo - O gerente chama o cliente através desse semáforo.
 - Thread atual
- Gerente
 - Cliente sendo atendido
 - Mutex lock - Lock usado para evitar escrita paralela por duas threads.
 - Semáforo - usado para avisar o gerente que chegou um cliente novo.
 - Flag dizendo que acabou o turno (`stop_working_flag`) setada após todos os clientes chegarem.
- State
 - Fila de clientes a serem atendidos
 - Array de cadeiras do banco
 - Gerente do banco
 - Mutex lock - Lock usado para evitar escrita paralela por duas threads.

Lógica

- Quando um cliente chega, ele procura a primeira cadeira que não tem ninguém sentado. (`client_arrives()`) e senta nela se existir. Criando uma thread para monitoramento (`chair_handler()`).
- Gerente é notificado através do semáforo `client_notification` que chegou um novo cliente, se estiver no meio do atendimento, será lido somente depois.
- O gerente ficou livre e avisa o próximo cliente da fila pelo semáforo `manager_call` que deve ir até a sala dele, liberando a cadeira e finalizando a thread dela.
- Após o atendimento, o cliente vai procurar o próximo cliente da fila e continua o ciclo.
- Ao acabar a fila, e ter finalizado o turno do gerente(`stop_working_flag`), o gerente vai pra casa finalizando o programa.

O loop do gerente é localizado na função `manager_handler()`

Execução

Para executar o programa basta rodar o comando `make run`, está hardcoded a lista de clientes que vão chegar e quando.

Exemplo da execução:

Output

```
./target/main
Thu Nov 4 02:16:35 2021 There is a total of 5 chairs!
Thu Nov 4 02:16:35 2021 🪑 Kleber chegou e sentou na cadeira 0!
Thu Nov 4 02:16:35 2021 🙋 Gerente atendendo o client Kleber !
Thu Nov 4 02:16:35 2021 ✨ Cadeira 0 agora está livre!
Thu Nov 4 02:16:35 2021 🪑 Carlos chegou e sentou na cadeira 1!
Thu Nov 4 02:16:35 2021 🪑 João chegou e sentou na cadeira 0!
Thu Nov 4 02:16:35 2021 🪑 Ednaldo chegou e sentou na cadeira 2!
Thu Nov 4 02:16:35 2021 🪑 Leonardo chegou e sentou na cadeira 3!
Thu Nov 4 02:16:35 2021 🪑 Valdir chegou e sentou na cadeira 4!
Thu Nov 4 02:16:35 2021 😡 Osmar não achou cadeira vazia e foi embora!
Thu Nov 4 02:16:35 2021 😡 Ligma não achou cadeira vazia e foi embora!
Thu Nov 4 02:16:37 2021 😡 Eduarda não achou cadeira vazia e foi embora!
Thu Nov 4 02:16:47 2021 🧑 Finalizado atendimento do cliente Kleber!
Thu Nov 4 02:16:47 2021 🙋 Gerente atendendo o client Carlos !
Thu Nov 4 02:16:47 2021 ✨ Cadeira 1 agora está livre!
Thu Nov 4 02:16:53 2021 🧑 Finalizado atendimento do cliente Carlos!
Thu Nov 4 02:16:53 2021 🙋 Gerente atendendo o client João !
Thu Nov 4 02:16:53 2021 ✨ Cadeira 0 agora está livre!
Thu Nov 4 02:16:56 2021 🧑 Finalizado atendimento do cliente João!
Thu Nov 4 02:16:56 2021 🙋 Gerente atendendo o client Ednaldo !
Thu Nov 4 02:16:56 2021 ✨ Cadeira 2 agora está livre!
Thu Nov 4 02:16:57 2021 🪑 Leticia chegou e sentou na cadeira 0!
Thu Nov 4 02:17:02 2021 🪑 Gerson chegou e sentou na cadeira 1!
Thu Nov 4 02:17:02 2021 🧑 Finalizado atendimento do cliente Ednaldo!
Thu Nov 4 02:17:02 2021 🙋 Gerente atendendo o client Leonardo !
```

```
Thu Nov 4 02:17:02 2021 ✨ Cadeira 3 agora está livre!
Thu Nov 4 02:17:04 2021 🪑 Simone chegou e sentou na cadeira 2!
Thu Nov 4 02:17:10 2021 🚶 Finalizado atendimento do cliente Leonardo!
Thu Nov 4 02:17:10 2021 🙋 Gerente atendendo o client Valdir !
Thu Nov 4 02:17:10 2021 ✨ Cadeira 4 agora está livre!
Thu Nov 4 02:17:20 2021 🚶 Finalizado atendimento do cliente Valdir!
Thu Nov 4 02:17:20 2021 🙋 Gerente atendendo o client Leticia !
Thu Nov 4 02:17:20 2021 ✨ Cadeira 0 agora está livre!
Thu Nov 4 02:17:24 2021 🪑 Lucas chegou e sentou na cadeira 0!
Thu Nov 4 02:17:25 2021 🚶 Finalizado atendimento do cliente Leticia!
Thu Nov 4 02:17:25 2021 🙋 Gerente atendendo o client Gerson !
Thu Nov 4 02:17:25 2021 ✨ Cadeira 1 agora está livre!
Thu Nov 4 02:17:26 2021 🚶 Finalizado atendimento do cliente Gerson!
Thu Nov 4 02:17:26 2021 🙋 Gerente atendendo o client Simone !
Thu Nov 4 02:17:26 2021 ✨ Cadeira 2 agora está livre!
Thu Nov 4 02:17:41 2021 🚶 Finalizado atendimento do cliente Simone!
Thu Nov 4 02:17:41 2021 🙋 Gerente atendendo o client Lucas !
Thu Nov 4 02:17:41 2021 ✨ Cadeira 0 agora está livre!
Thu Nov 4 02:17:56 2021 🪑 Cuca chegou e sentou na cadeira 0!
Thu Nov 4 02:18:04 2021 🚶 Finalizado atendimento do cliente Lucas!
Thu Nov 4 02:18:04 2021 🙋 Gerente atendendo o client Cuca !
Thu Nov 4 02:18:04 2021 ✨ Cadeira 0 agora está livre!
Thu Nov 4 02:18:46 2021 🚶 Finalizado atendimento do cliente Cuca!
Thu Nov 4 02:18:46 2021 🍷 Não tem mais nenhum cliente para atender e
acabou o turno do gerente!
```

Dificuldades

Acabei tendo um pouco de dificuldade inicialmente pois não me lembrava nada de C, mas aos pouquinhos foi ficando mais fácil.

Após isso, foi difícil pensar nos sistema de comunicação do gerente com os clientes na cadeira, e acabei optando por um semáforo.

Precisei ficar bem atento para não cair em Deadlocks, teve situações que uma função que trancava o cliente chamava outra função que tentava adquirir o lock, gerando um Deadlock, demorei bastante para perceber o que estava ocorrendo.

Conclusão

Desenvolver o trabalho sozinho foi relativamente fácil, não tive grandes dificuldades com o conteúdo em si, e sim com a linguagem.

SOSIM

Vídeo com a apresentação do software: <https://www.youtube.com/watch?v=b5HBceJQigY>