# Sistema simplificado de controle de elevador

Data de entrega: até 14 de março de 2025.

Loca: Embedded, sala 105.

Deverá ser feita uma defesa oral.

# Descrição geral

Neste projeto, vamos desenvolver um simulador de elevador concorrente utilizando linguagem C e semáforos no Linux. O sistema contará com um único elevador que atende múltiplos usuários concorrentes em um prédio de 5 andares. O objetivo é gerenciar chamadas de forma organizada, garantindo que o elevador atenda os usuários corretamente sem conflitos entre threads.

#### **Objetivos**

- Implementar threads concorrentes no Linux.
- Utilizar **semáforos** para controle de acesso e sincronização.
- Gerenciar uma fila de chamadas de maneira eficiente.
- Simular movimentação do elevador baseada em requisições.

## Funcionamento do sistema

## Principais componentes

- Usuários (threads) fazem chamadas para o elevador de andares aleatórios.
   Não deve haver limite na quantidade de usuários do sistema, embora você possa gerar novos usuários a uma determinado fluxo de usuários.
- 2. Cada chamada é armazenada em uma fila compartilhada controlada por semáforos. Você deve garantir que essa fila seja grande o suficiente para atender ao fluxo de usuários.
- O elevador (outra thread) verifica a fila e atende as chamadas na ordem de chegada (FIFO).
- 4. Quando o elevador chega ao andar solicitado, ele **aguarda alguns segundos** para simular embarque/desembarque.
- 5. A thread continua até que todas as chamadas tenham sido atendidas.

#### Regras do elevador

- O elevador começa no andar 0 e só se move quando há chamadas pendentes.
- As chamadas são atendidas na ordem de chegada, independentemente da direção.
- Durante o deslocamento, **novas chamadas podem ser adicionadas à fila**, exigindo que o elevador as processe assim que possível.
- Um **semáforo protege o acesso à fila**, evitando que múltiplos usuários modifiquem os dados ao mesmo tempo.

# Exemplo de execução

Um possível caso de uso é dado a seguir:

- Usuário 1 chama o elevador do andar 2 para o andar 4.
- Usuário 2 chama do andar 0 para o andar 3.
- O elevador atende primeiro a chamada do andar 2 para o 4 e depois desce ao andar 0 para levar o usuário ao 3.

O sistema deve continuar executando indefinidamente e sempre atender novas chamadas de usuários que chegam.

## Exemplo de saída

A seguir temos um exemplo da saída esperada do seu sistema.

```
[Elevador] Inicializado no andar 0. Aguardando chamadas...
[Usuário 1] Chamando o elevador do andar 2 para o andar 4.
[Usuário 2] Chamando o elevador do andar 0 para o andar 3.
[Elevador] Nova chamada recebida de Usuário 1: Andar 2 -> 4.
[Elevador] Movendo-se para o andar 2...
[Elevador] Chegou ao andar 2. Usuário 1 embarcou.
[Elevador] Movendo-se para o andar 3...
[Elevador] Movendo-se para o andar 4...
[Elevador] Chegou ao andar 4. Usuário 1 desembarcou.
[Elevador] Nova chamada recebida de Usuário 2: Andar 0 -> 3.
[Elevador] Movendo-se para o andar 3...
[Elevador] Chegou ao andar 3...
[Elevador] Chegou ao andar 3. Usuário 2 desembarcou.
[Elevador] Nenhuma chamada pendente. Retornando ao modo de espera.
```

# Entregáveis

Você deverá entregar 3 artefatos:

- 1. Código fonte do projeto;
- 2. Documentação;
- 3. Vídeo no YouTube com demonstração e explicação do código.

Seu projeto deverá idealmente ser desenvolvido em C. Mas também é permitido fazer em C++, Java ou Python.

Você deverá enviar um link para o repositório no GitHub com seu código fonte. Adicione o usuário kyllercg@gmail.com com permissão de leitura ao repositório. Você só precisa enviar esse link. Todos os demais itens devem estar contidos no repositório. Envie 'link para kyller@dee.ufcg.edu.br.

A documentação do projeto deverá estar no GitHub (README). Nela, você deve explicar o que foi feito, como o sistema funciona e como eu faço para compilar seu projeto.

Lembre-se que eu irei compilar o código em meu computador. Portanto, você deverá colocar na documentação toda instrução necessária (incluindo o passo a passo para instalação de bibliotecas adicionais, se for necessário), para que eu possa repetir o procedimento.

O vídeo de demonstração não deverá exceder mais do que 10 minutos.

### Outras coisas

1. Pode ser feito em grupos de até 3 pessoas.