

Especificação do Projeto 1



Sistemas Operacionais

Prof. Lucas Figueiredo

Objetivo do Projeto

O primeiro projeto da disciplina será uma implementação de um **simulador multi-thread** que deve gerenciar múltiplas tarefas concorrente. **A cada semana de laboratório, os grupos deverão incrementar o projeto com novos conceitos estudados em cada aula**, começando com a criação de threads. O simulador deve garantir uma execução sincronizada das threads, evitando problemas de concorrência.

Cada grupo terá liberdade para personalizar o tema da simulação (ex.: pedidos de cliente, processamento de dados, simulação de combate em jogos, etc), respeitando os requisitos técnicos definidos a cada semana.

Requisitos do Projeto

Os requisitos detalhados para cada etapa do projeto serão descritos na respectiva semana. A seguir, estão as descrições gerais de cada semana:

1. Uso de Threads (Semana 1)

Os grupos deverão **criar threads** para simular tarefas concorrentes, usando a biblioteca **pthread**. A criação, execução e finalização correta das threads são essenciais.

2. Sincronização Básica (Semana 2)

Implementação de **sincronização** entre threads utilizando **mutexes** para evitar condições de corrida. As threads devem ser coordenadas para evitar o uso simultâneo de variáveis ou recursos compartilhados.

Exemplo: Controlar o acesso a um recurso compartilhado, como uma variável global.

3. Comunicação entre Threads (Semana 3)

As threads devem se **comunicar** para coordenar suas tarefas usando mecanismos de **comunicação entre processos/threads (IPC)**. Será necessário coordenar a execução de threads.

Exemplo: Deve haver pelo menos uma thread que espera por outra para continuar sua execução, como em uma linha de produção.

4. Sincronização Avançada e Controle de Concorrência (Semana 4).

O sistema deve gerenciar o acesso de threads a **recursos limitados** (ex.: número limitado de slots em um buffer ou recursos físicos). Use **semáforos** para garantir que o acesso a esses recursos seja controlado de forma eficiente.

Exemplo: Controle de acesso a recursos limitados, como um número fixo de médicos em um hospital ou de pistas de pouso em um simulador de tráfego aéreo.

Apresentação Final

Na primeira semana de outubro, na aula de laboratório, cada grupo deverá apresentar seu simulador, explicando o funcionamento de cada componente e como os problemas de concorrência foram resolvidos. A apresentação deve incluir:

- Explicação sobre as threads implementadas e suas funções.
- Descrição dos mecanismos de sincronização utilizados (mutexes, semáforos, variáveis de condição).
- Logs que mostrem o comportamento do sistema em termos de criação e finalização de threads, sincronização e controle de recursos.

Critérios de Avaliação

O projeto será avaliado com base nos seguintes critérios, com uma pontuação total de **10 pontos**:

PROJETO ENTREGUE (7 PONTOS)

1. Funcionalidade e Correção (3 pontos)

- O simulador funciona conforme os requisitos propostos em cada semana.
- As threads são corretamente criadas, executadas e finalizadas.
- O código implementa corretamente os conceitos de threads, mutexes, variáveis de condição e semáforos.
- O projeto evita problemas de concorrência, como condições de corrida, deadlocks ou starvation.

2. Adequação da Implementação ao Contexto (1 ponto)

- O simulador é bem adaptado ao tema escolhido pelo grupo, seja ele pedidos de clientes, controle de tráfego, ou outro tema.
- A lógica de funcionamento e a escolha de recursos e threads são coerentes com o cenário proposto.
- O projeto demonstra que as threads e os recursos foram implementados de forma apropriada ao problema que estão simulando.

3. Criação e Organização dos Logs (1 ponto)

- O sistema gera logs que acompanham o ciclo de vida das threads (criação, execução, sincronização e finalização).
- Os logs são bem estruturados e fornecem informações claras e úteis para acompanhar o comportamento do sistema.
- A organização dos logs permite identificar facilmente eventos críticos como troca de contexto, sincronização de threads e uso de recursos limitados.

4. Documentação (2 pontos)

- O código é bem documentado com comentários explicativos, especialmente em áreas que envolvem lógica de concorrência ou sincronização.
- O grupo entrega um documento descrevendo a estrutura do simulador, a função de cada parte do código e as decisões de design.
- A documentação cobre tanto a implementação quanto o comportamento do sistema durante a execução.

APRESENTAÇÃO (3 PONTOS)

1. Clareza e Organização da Apresentação (1.5 pontos)

- A apresentação é clara, organizada e cobre todos os aspectos do projeto.
- O grupo explica de forma coerente como cada parte do simulador foi implementada, destacando a criação, sincronização e comunicação das threads.
- O grupo justifica as escolhas feitas ao longo do desenvolvimento, como o uso de semáforos ou mutexes em determinadas situações.

2. Demonstração do Funcionamento do Sistema (1 ponto)

- O grupo demonstra o simulador funcionando ao vivo ou apresenta resultados que comprovam o correto funcionamento.

- A demonstração inclui a criação e finalização das threads, além da geração de logs que ilustram o comportamento do sistema em diferentes situações.
- A demonstração foca em mostrar como o sistema gerencia recursos limitados e resolve problemas de concorrência.

3. Capacidade de Responder Perguntas (0.5 pontos)

- O grupo responde adequadamente às perguntas feitas sobre a implementação, a lógica das threads e a resolução de problemas de concorrência.
- Todos os membros do grupo demonstram conhecimento sobre o funcionamento do projeto e a lógica de suas implementações.

Entrega

Os grupos deverão entregar o código-fonte e um documento explicativo, descrevendo a estrutura do simulador, os passos necessários para sua execução, as decisões tomadas durante o desenvolvimento e os resultados obtidos. Os logs gerados também devem ser apresentados como parte da entrega.