

## Entrega do trabalho dia 10/03

Grupos de 1 até 2 participantes: Entregar Tema 1 (Implementação com threads) e Tema 2 (SOSim)

Grupos de 1 até 3 participantes: Entregar Tema 1, Tema 2 e Tema Extra.

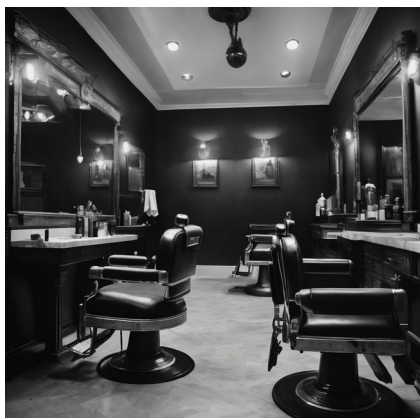
O que entregar?

- Etapa 1 – Código e apresentação/vídeo
  - Conteúdo do Vídeo:
    - Nome dos participantes.
    - Não é necessário apresentar linha por linha, apenas mostrar onde estão sendo lançados os threads e qual a lógica aplicada. Apresente como implementou os barbeiros, cadeiras, tesouras e pentes abordando a lógica aplicada em cada um.
    - Apresente as dificuldades (problemas e soluções) encontradas.
    - Relacione o programa desenvolvido com conceitos como: deadlock, preempção, starvation e inanição. Discuta sobre esses assuntos em relação ao seu sistema.
    - Sugestão de trabalhos para os próximos semestres [opcional].
- Etapa 2- Artigo (formato SBC, no mínimo 4 páginas e no máximo 8 páginas) ou vídeo.
- Etapa Extra – Artigo (formato SBC, no mínimo 4 páginas e no máximo 8 páginas).

### OBSERVAÇÕES IMPORTANTES:

- Trabalho em grupo deve ser realizado por todos participantes.
- É importante ter em mente que ao participar de um grupo, está assinando por todo o trabalho. Logo, poderá ser questionado por qualquer atividade relacionada (Não será aceita a resposta: "eu não fiz essa parte, foi o fulano").
- Posso sortear algum grupo (ou mais de um) para uma apresentação extra. Caso isso ocorra, os participantes podem receber notas diferentes.
- É um trabalho em grupo e todos devem participar. Em casos de problemas no grupo, me enviem um e-mail e esse grupo será um forte candidato a ser convocado para apresentação.
- Problemas com cópia/compartilhamento de trabalho (mesmo parcial), a nota será zerada.

## TEMA 1 - Simulação da barbearia “Pronto Socorro”



O problema consiste em simular o funcionamento de uma barbearia com as seguintes características. A barbearia tem uma sala de espera com N cadeiras de espera e M cadeiras de barbear (N representa uma quantidade não fixa nesta descrição, experimente variar e explique na apresentação o impacto disto em relação a quantidade de clientes atendidos; e M igual a quantidade de barbeiros). Se não tem clientes à espera, os barbeiros sentam em cadeiras e dormem. Quando chega um cliente, ele acorda um dos barbeiros. Se chega outro cliente enquanto todos os barbeiros estão trabalhando, ele ocupa uma cadeira e espera (se tem alguma cadeira disponível) ou vai embora (se todas as cadeiras estão ocupadas).

Um barbeiro, para poder cortar o cabelo de alguém, precisa ter acesso a uma tesoura e um pente, porém, nessa barbearia não há uma quantidade de tesouras e pentes que permita que todos os barbeiros trabalhem ao mesmo tempo.

Vamos direcionar nossa simulação apenas em uma barbearia hipotética direcionada ao atendimento de clientes. Para isso, é necessário ter um número  $N$  de barbeiros, um número  $N/2$  pentes e um número  $N/2$  tesouras.

Com esse contexto, simule o seguinte cenário usando Pthreads.

Cada barbeiro é um thread e cada um dos clientes também.

Uma função principal deve ficar lançando uma quantidade indefinida de threads, que representam clientes, com tempos variados entre eles (use um valor aleatório a cada iteração), até que alguma condição de parada faça ela encerrar.

Um cliente (thread lançado) deve ir para a execução buscando uma cadeira, enquanto houver espaço. Caso as cadeiras estejam ocupadas, o cliente vai embora. Obs. (Dica): cuidado com o handle do join.

Cada cliente deve ter um tempo diferente para cada corte, isso porque supõe-se que cada cliente escolheu um corte de cabelo diferente. Os tempos devem variar de 3 a 6 segundos.

Ao final da execução, deve-se informar a quantidade de clientes que foram atendidos e quantos desistiram devido a sala de espera estar cheia.

Enquanto a condição de parada não for atingida, a main deve ficar criando threads e enviando para a execução. A condição de parada pode ser um tempo pré-definido passado como parâmetro de execução, ou um tempo aleatório, ou a leitura de um getchar() na main.

## **TEMA 2 – Simulador SOSim**

Com o objetivo de explorar os conceitos estudados de Sistemas Operacionais, utilize o simulador SOSim<sup>1</sup> (funciona no Linux com o uso do Wine) e crie um documento, no formato de artigo (use o modelo SBC). Apresente o simulador, e descreva sobre a gerência de processos e memória do sistema [no mínimo 4 páginas e no máximo 8 páginas].

Ou vídeo não listado no Youtube sobre o SOSim, abordando:

a) Visão geral do simulador

- Apresente as principais janelas e funcionalidades

b) Sobre processos

- Utilize no mínimo dois tipos de processos: *CPU-bound* e *I/O-bound*.

- No simulador, quais são os possíveis estados para os dois tipos de processos acima?

- Explique sobre fatia de tempo, clock e qual impacto desses conceitos no simulador?

---

<sup>1</sup> <http://www.training.com.br/sosim/>

- Demonstre o uso do “escalonamento circular” (dica: é um parâmetro do sistema).

- Faça o seguinte experimento:

Utilize escalonamento circular com Prioridade estática e crie:

- 2 processos com prioridade 3 para I/O;
- 2 processos com prioridade 2 para misto;
- 2 processos com prioridade 1 para CPU.

E se as prioridades forem definidas ao contrário?

#### c) Gerência de Memória

- Mostre experimentos trocando a política de busca (paginação antecipada e paginação por demanda). Sugestão: Também, analise a quantidade de “*page fault*” no arquivo de log por um determinado período de tempo.

- Mostre as janelas de gerência de memória, paginação e log.

- Apresente e analise a Tabela de Páginas (PCB → ver tab. de páginas)

#### d) Conclusão

- Faça uma análise crítica do simulador SOSim.

- Compare os conceitos vistos em aula com o SOSim.

- Existe outro simulador disponível? Descreva brevemente.

## TEMA EXTRA

O que entregar?

- Artigo com no mínimo 4 páginas e no máximo 8 páginas.

Tema:

- Faça um artigo com um dos dois últimos conteúdos do curso: Sistemas de Arquivos ou Sistemas de I/O de Sistemas Operacionais

### **Exemplos de Propostas**

- Comparação da implementação do Sistema de Arquivos no Windows e Linux.
- Comparação da implementação do Sistema de I/O no Windows e Linux.
- Análise de um sistema de arquivos distribuídos, por exemplo: NFS (Network File System).
- **Se tiver dúvida se a sua proposta atende o solicitado, envie um e-mail com uma breve descrição.**