

Entrega do trabalho dia 11/11

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 vídeo (pode ser vídeo não listado no Youtube).
 - 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 pacientes, nebulizadores, enfermeiros e médicos, 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.
 - 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 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 de uma sala de Pronto Socorro



Uma sala de pronto-socorro, também conhecida como sala de emergência ou pronto atendimento, é um espaço médico em uma unidade de saúde destinado a fornecer atendimento imediato e urgente a pacientes que necessitam de assistência médica. Estas salas possuem uma capacidade limitada e são equipadas com equipamentos médicos avançados, como monitores cardíacos, aparelhos de respiração assistida, medicamentos de emergência e equipes médicas especializadas, incluindo médicos, enfermeiros e técnicos treinados para lidar com situações críticas. O objetivo principal de uma sala de pronto-socorro é estabilizar e tratar os pacientes de forma rápida e eficaz, encaminhando-os posteriormente para cuidados adicionais, se necessário.

Vamos direcionar nossa simulação apenas em uma ala hipotética direcionada ao atendimento de pacientes com insuficiência respiratória. Neste contexto, nossa simulação possui uma sala equipada com nebulizadores com a capacidade de atender 4 pacientes simultaneamente e permite que outros 12 pacientes aguardem por atendimento (totalizando até 16 pacientes). Neste ambiente, existem enfermeiros fazendo a triagem e monitorando os sinais vitais dos pacientes (parâmetros como a frequência cardíaca, a pressão arterial, a temperatura corporal e a frequência respiratória), enquanto esperam o atendimento do médico.

Devido a limitação de nebulizadores, os 16 pacientes que apresentam insuficiência respiratória estão disputando por 4 nebulizadores, eles usam por um tempo e voltam para os assentos.

De tempos em tempos, um médico vem até a sala e leva o paciente mais velho (maior idade) para tratamento e posterior liberação (ou internação em outra ala que não será abordada nesta simulação), este atendimento do médico libera uma vaga da sala, permitindo que uma nova pessoa entre na sala de espera.

Existem 16 assentos na sala, portanto pode ter até 16 pacientes esperando pelo tratamento, que ficam alternando acesso aos nebulizadores. Existem, então, 4 nebulizadores que aumentam os sinais vitais dos pacientes enquanto são nebulizados. E, por fim, existem 2 médicos que de tempos em tempos atendem um dos pacientes da sala, liberando espaço para mais pessoas.

Com esse contexto, simule o seguinte cenário usando Pthreads:

Cada paciente é um thread e cada um dos médicos também. Existe o chefe dos enfermeiros que fica variando o acesso dos pacientes aos nebulizadores.

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

Um paciente (thread lançado) deve ir para a execução na sala de espera, enquanto houver espaço. Caso a sala de espera esteja lotada (já possui 16 pacientes), o paciente será encaminhado para outro hospital (thread será ignorado e não irá disputar por recursos). Obs. (Dica): cuidado com o handle do join.

Características relevantes:

c1) O sinal vital varia 0 a 10, onde 0 é um sinal vital de risco e 10 um sinal vital bom.

c2) O paciente que não está utilizando o nebulizador possui seu sinal vital reduzido em 1 ou 2 níveis, periodicamente, (dica: use a função rand()). Por outro lado, enquanto estiver utilizando o nebulizador, seu sinal vital terá um aumento de 2 a 4 níveis, também, periodicamente.

c3) Quem controla quanto tempo uma pessoa fica nebulizando é o chefe dos enfermeiros (função análoga a um escalonador). Esse valor de tempo não deve ser maior do que o tempo que os médicos levam para atender.

c4) O tempo de atendimento dos médicos é aleatório (por paciente) e quando um paciente é atendido, ele é liberado com alta (Dica: cuidado com o handle do join).

c5) Deve-se permitir que o simulador tenha o seu tempo total de execução, ajustável.

c6) A função main() deve lançar threads até que o tempo de simulação se esgote, mas deve ser garantido aos threads, já lançados, o tempo necessário para que todos os restantes sejam atendidos.

Ao final da execução, a simulação deve indicar quantos pacientes foram atendidos, quantos foram até o hospital e quantos pacientes chegaram a um nível vital igual a zero.

TEMA 2 – Simulador SOSim

Com o objetivo de explorar os conceitos estudados de Sistemas Operacionais, utilize o simulador SOSim¹ (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?
- 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.

¹<http://www.training.com.br/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

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.**