

Trabalho Prático – Escalonamento do Disco

1. Objetivos

Permitir que os alunos exercitem seus conhecimentos sobre os conceitos e os algoritmos de escalonamento estudados na disciplina. Possibilitar que o aluno implemente mecanismos e políticas fundamentais para o funcionamento de sistemas operacionais, incluindo gerencia da fila, escalonamento do disco e cálculo de métricas de interesse.

2. O que deve ser feito

Os alunos devem implementar a simulação de um sistema com escalonador de disco. As políticas FCFS, SSTF, SCAN, C-SCAN e C-LOOK devem ser implementadas. Os alunos devem propor uma nova política de escalonamento (MY). A política MY não pode se valer de informações do futuro, não pode ser meramente aleatória, não pode requerer alterações no formato dos dados de entrada, não pode ser igual às políticas estudadas. Além disso, a política deve apresentar desempenho superior a pelo menos três das políticas vista em sala para um arquivo de entrada definido pelo aluno.

Toda a implementação deve ser parte de um mesmo projeto. O programa principal desse projeto deve ler um arquivo de entrada, executar as políticas de escalonamento de disco e, com base nos resultados obtidos, gerar um arquivo de saída. O formato do arquivo de entrada e de saída está exemplificado abaixo.

NumSectors=200 NumTracks=50 StartPosition=0 R1=0,10,4,3 R2=1,20,2,3 R3=2,18,2,3 R4=3,17,2,4 R5=4,30,2,5 R6=5,100,2,6 R7=6,150,2,7 ;	FCFS -AccessTime= -WaitingTime= SSTF -AccessTime= -WaitingTime= SCAN -AccessTime= -WaitingTime= C-SCAN -AccessTime= -WaitingTime= C-LOOK -AccessTime= -WaitingTime= MY -AccessTime=??. -WaitingTime=??.
Exemplo do formato do arquivo de entrada (in.txt)	Exemplo do formato do arquivo de saída (out.txt)

Para avaliação das implementações, serão utilizados diversos arquivos de entrada diferentes. Algumas avaliações serão automatizadas. Portanto, o arquivo de saída gerado pela implementação deve seguir rigorosamente o formato exemplificado acima.

Uma requisição ao disco tem o formato “R1=A,B,C,D”, onde A é o tempo de chegada, B é o cilindro desejado, C é a trilha desejada e D é quantidade de dados. Como simplificação, considere seek time=B e transfer time=D. Latency time é dado pela quantidade de cilindros que precisarão ser superados até chegar ao cilindro desejado. Note que a quantidade e as características das requisições podem mudar dependendo dos dados de entrada. Também é importante frisar que os dados de seek time, latency time, transfer time são uma aproximação. Isso é uma simplificação para viabilizar a simulação, pois esses tempos também variam de uma máquina para outra, dependendo da velocidade do disco, etc. Conforme o exemplo, no arquivo de saída deve constar a média do tempo de acesso (accessTime) e do tempo de espera (waitingTime) para cada política de escalonamento.

Um membro da equipe deve apresentar (em 5 minutos) a implementação feita e as ideias da política MY implementada pela sua equipe. Tal membro será escolhido pelo professor no momento da apresentação. Como parte da apresentação, o professor escolherá um trecho de código e pedirá ao aluno para explicar.

3. Aquecimento

Como tarefa de aquecimento, os alunos devem: 1) elaborar um arquivo de entrada e um de saída (diferentes dos exemplos apresentados em sala) e que possam ser usados para teste da implementação; e 2) implementar e testar o programa com as políticas FCFS e SCAN. Naturalmente, para ter êxito nessa fase de aquecimento, os alunos devem estruturar todo o projeto e implementar as rotinas de entrada e saída de dados.

4. O que deve ser entregue

- Todo o projeto da implementação.
- Um relatório de 3 páginas descrevendo as ideias utilizadas nas implementações, as bibliotecas pesquisadas, os métodos dessas bibliotecas que foram utilizados, as dificuldades experimentadas durante a implementação, e as análises realizadas para saber se o código implementado está correto.

5. O que será avaliado

Será avaliado se o código está correto ou não. Um código será considerado correto se ele atende ao que foi especificado no item 2. Qualidade do relatório e da apresentação de acordo com o que foi especificado no item 3. Se ocorrer qualquer evidência de cópia de trabalhos, receberão nota zero todos os membros das equipes em que os trabalhos estiverem envolvidos em cópia.

É esperado que o aluno empregue boas práticas de engenharia de software incluindo a legibilidade do

código em termos de comentários, variáveis com nomes significativos, métodos com nomes significativos. É fortemente recomendado o versionamento de código no GitHub, mantendo o repositório privado para a equipe durante o desenvolvimento. Dica: o uso de padrões de projeto de *software*, como *factory*, pode simplificar muito o trabalho de desenvolvimento e reduzir drasticamente a quantidade de código que precisa ser escrita.

5. Equipe, avaliação e prazos

- Tamanho da equipe: 2 membros
- Valor do aquecimento: 5 pontos
- Valor do trabalho final: 10 pontos
- Data de entrega da tarefa de “Aquecimento”: 29/10/2018 (pelo SGA)
- Data da entrega final do código e do relatório: 22/11/2018 (pelo SGA)
- Data das apresentações dos trabalhos: 26/11/2018

Serão atribuídos 1 ponto extra a cada membro da equipe que implementar a política MY que gerar o menor tempo médio de espera e o menor tempo médio de acesso em um conjunto de entradas que será definido pelo professor.