

Roteiro de Estudos 3

Processos - Escalonamento

- LEITURA

- MAZIERO, C. Sistemas Operacionais: Conceitos e Mecanismos. Editora UFPR, 2019. 456 p. ISBN 978-85-7335-340-2
 - [Cap. 6, p. 70-82 \(Até a seção 6.4 inclusa\)](#)
 - [Cap. 7, p. 87-90](#)
 - ATENÇÃO!! Algumas obs. sobre este livro:
 - O termo “Tarefa” é utilizado no lugar de “Processo”
 - “orientadas a processamento” == *CPU bound*
 - Pag 71/72: “**Suspenso/Suspended**” é utilizado no lugar “**Bloqueado/Blocked**”
 - Pag 72: NÃO precisam ler o último parágrafo da seção 6.3. Isso será explicado no próximo roteiro.
 - Pag. 77 (seção 6.4.5): “algoritmo por prioridade **cooperativo**” é na verdade o algoritmo por prioridade NÃO-preemptivo, e NÃO utilizaremos no curso!

- VÍDEOS

- Neso Academy
 - [Introduction to CPU Scheduling](#) (10'13")
 - [CPU and I/O Burst Cycles](#) (8'15")
 - [Preemptive and Non-Preemptive Scheduling](#) (18'56")
 - [Scheduling Criteria](#) (13'34")
 - Scheduling Algorithms
 - [First Come First Served Scheduling \(Solved Problem 1\)](#) (18'49")
 - [Shortest Job First Scheduling \(Solved Problem 1\)](#) (11'58")
 - [Priority Scheduling \(Solved Problem 1\)](#) (15'57")
 - [Round Robin Scheduling - Solved Problem \(Part 1\)](#) (19'53")
 - [Round Robin Scheduling - Solved Problem \(Part 2\)](#) (6'50")
 - [Multilevel Queue Scheduling Algorithm](#) (14'35")
 - [Multilevel Feedback-Queue Scheduling Algorithm](#) (13'30")

- RESUMÃO

- [Slides com uma compilação do conteúdo](#)

- **EXERCÍCIOS (valendo turings!!)**
 - **Responda o formulário fornecido juntamente com este roteiro**

=====

Lista de Exercícios de Consolidação

O objetivo da lista é ajudar no estudo individual dos alunos. Soluções de questões específicas poderão ser discutidas em sala de aula, conforme interesse dos alunos.

=====

1. Explique as funções dos escalonadores de curto, médio e longo prazo.
2. Qual a relação entre Tempo de Espera e Tempo de Resposta?
3. O que significa um processo sofrer preempção?
4. A maioria dos escalonadores Round Robin usa um quantum de tamanho fixo. Dê um argumento em favor de um quantum pequeno. Agora pense em um argumento que justifique um quantum grande.
5. Considere o seguinte algoritmo de alocação de CPU (escalonamento) por prioridade, preemptivo, baseado em prioridades que mudam dinamicamente. Números de prioridades maiores indicam prioridades mais altas. Quando um processo está esperando para entrar em execução (na fila de prontos), sua prioridade muda segundo uma **taxa α** ; quando está em execução, sua prioridade muda segundo uma **taxa β** . Todos os processos têm a mesma prioridade quando são criados. Valores diferentes para os parâmetros α e β podem determinar muitos algoritmos de alocação diferentes.
 - a) Qual algoritmo é obtido com $\alpha > \beta > 0$?
 - b) Qual algoritmo é obtido com $\beta > \alpha > 0$?
6. Suponha um S.O. com escalonador de filas multiníveis na qual há cinco níveis (prioridade entre os níveis). O quantum do primeiro nível é 0,5 segundo. Cada nível mais baixo tem um quantum de tamanho duas vezes maior que o quantum do nível anterior. Um processo não pode sofrer preempção até o seu quantum terminar. Todos os processos iniciam na fila mais prioritária. Se um processo é preemptado, ele deve ser colocado na fila imediatamente inferior. O sistema executa processos em lote (cpu bound) e interativos (I/O bound).
 - a. Por que esse sistema é deficiente?
 - b. Quais mudanças mínimas você proporia para tornar o esquema mais aceitável para o mix de processos que pretende?
7. Qual dos algoritmos de escalonamento discutidos em sala de aula poderia ser modificado para acomodar alguns processos de tempo real (processos que devem ter uma resposta dentro de um certo período de tempo) misturados com os outros tipos de processo? Explique como você faria isso.

8. Cinco processos, de A até E, chegam ao computador ao mesmo tempo. Eles têm seus tempos de processamento estimados em 10, 6, 2, 4 e 8 minutos respectivamente. Suas prioridades (atribuídas externamente) são 3, 5, 2, 1 e 4, respectivamente, sendo 5 o representante da prioridade mais alta. Nenhum dos processos faz I/O. Para cada um dos algoritmos de escalonamento abaixo, determine o tempo médio de turnaround dos processos. Ignore o overhead causado pela troca de contexto.
- (a) Round Robin (fila começa em A, indo em ordem até E ; quantum = 4)
 - (b) Escalonamento com prioridade
 - (c) FIFO (ordem de execução: A, B, C, D, E)
 - (d) SJF
9. Considere um sistema operacional cuja máquina de estados inclui os estados Ready e Ready-Suspended (em disco). Suponha que seja hora do S.O. despachar (escalonar) um processo e que existam nesse momento processos tanto no estado Ready como no estado Ready-Suspended, e que pelo menos um processo no estado Ready-Suspended possui prioridade maior do que qualquer processo no estado Ready. Duas políticas extremas seriam: (a) sempre despachar um processo no estado Ready, de forma a minimizar swapping; e (b) sempre dar preferência ao processo de mais alta prioridade, mesmo que isso possa significar a ocorrência de swapping quando este não é necessário. Sugira uma política intermediária (explique e crie um algoritmo) que tente balancear prioridade e desempenho.
10. Considere um sistema que possui duas filas de escalonamento, com prioridades 0 e 1, sendo que somente pode ser escalonado um processo da fila de prioridade 1 não existindo processos na fila de prioridade 0. Sabendo que o algoritmo utilizado nas duas filas é o Round-Robin, escreva o pseudo-código dos procedimentos `insere(p)`, em que `p` é o índice da tabela de descritores (i.e. PCBs) de processos e que possui um campo (entre outros campos) que contém a prioridade dos processos e `r=seleciona()`, que retorna o índice da tabela de descritores que descreve o processo selecionado. Cite duas situações em que cada procedimento é chamado.