

Cap. 2 – Processos Parte 3

Prof. Marcelo Moreno

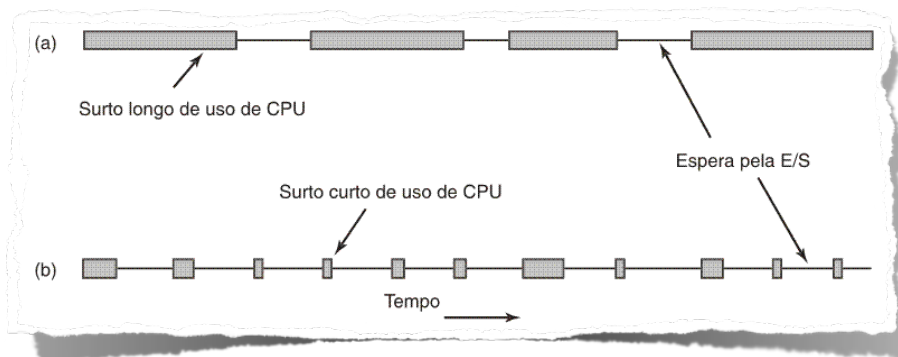
moreno@ice.ufjf.br

<http://sites.google.com/a/ice.ufjf.br/dcc062>

Escalonamento de Processos

- **Multiprogramação**
 - Diversos processos competem pela CPU
 - Cabe ao sistema operacional decidir o momento em que cada processo obterá a CPU
- **Escalonador de processos**
 - Subsistema do S.O. responsável por tal decisão
 - Utiliza um (ou até vários) algoritmos de escalonamento que estabelecem a lógica de tal decisão
- **Escalonamento vs. trocas de contexto**
 - Tempo gasto com chaveamento do modo usuário para o modo núcleo

Comportamento de Processos



(a) Processo CPU-bound ou orientado a CPU

(b) Processo I/O-bound ou orientado a E/S

Quando escalonar?

- Quando se faz necessária a escolha do próximo processo a obter a CPU?
 - Criação de um processo
 - Término de um processo
 - Processo é bloqueado
 - Após uma Interrupção
 - Periodicamente, a cada k inter. de relógio
- Escalonamento Não-Preemptivo
- Escalonamento Preemptivo
- Escalonamento não-preemptivo. Escalonamento apenas em situações que praticamente obrigam que uma decisão seja tomada.
 - Escalonamento preemptivo. Escolher um processo e lhe conceder a CPU durante um certo tempo. Findado o tempo, a CPU é concedida a outro processo.

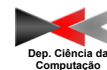
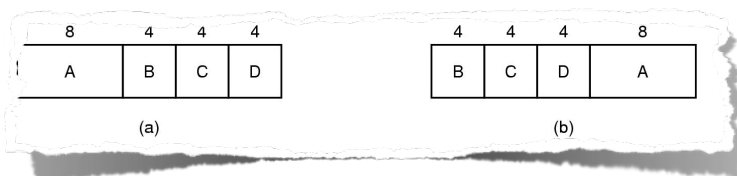
Categorias de algoritmos de escalonamento

- Escalonamento de processos pode envolver diferentes tipos de requisitos, seguindo assim diferentes parâmetros e diferentes lógicas
- Classificação segundo o tipo de sistema, tipo de aplicação...
- Tanenbaum:
 - Sistemas em Lote
 - Sistemas Interativos
 - Sistemas de Tempo Real



Escalonamento em Sistemas em Lote

- Job mais curto primeiro
 - Shortest Job First (SJF)
 - Exige conhecimento do tempo de execução de um job
 - Entre a fila de jobs aguardando escalonamento, escolhe aquele que possui o menor tempo de execução
 - Melhora o tempo médio de retorno
 - Desvantagem
 - Jobs chegando entre a execução de outros podem ter tempos curtos e caso escolhidos primeiro reduziriam o tempo médio de retorno



Escalonamento em Sistemas em Lote

- Primeiro a chegar, primeiro a ser servido
 - First come, first served (FCFS)
 - CPU é atribuída a processos na ordem em que a requisitaram
 - Uma única fila de processos prontos
 - Novos jobs entram no fim da fila
 - Simples e justo
 - Desvantagem
 - CPU-bound vs. I/O-bound
 - Preempção faz falta

Escalonamento em Sistemas em Lote

- Próximo de menor tempo restante
 - Shortest remaining time first (SRTF)
 - Novamente, tempo de execução de cada job deve ser conhecido
 - Na chegada de novo job, seu tempo total é comparado com o tempo restante dos jobs atualmente enfileirados
 - Será enfileirado antes daquele que possui tempo restante imediatamente superior a ele
 - Podendo inclusive tomar a CPU logo na chegada
 - Quase preempção



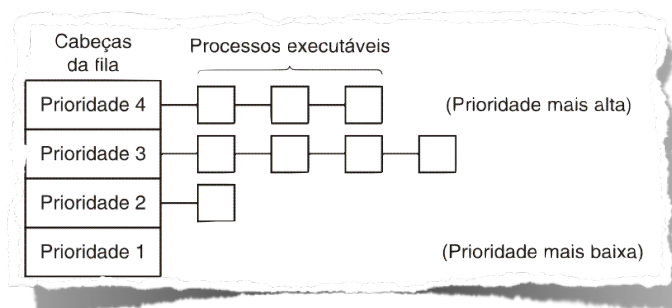
Escalonamento em Sistemas Interativos

- **Escalonamento por alternância circular**
 - **Round-robin**
 - Cada processo ganha um intervalo de tempo para uso contínuo da CPU (quantum)
 - Se ao final do quantum, processo ainda está processando, há preempção e outro processo será escolhido
 - Se houve bloqueio ou o processo terminou antes do fim do quantum, outro processo será escolhido
 - **Dimensionamento do quantum é sensível**
 - Overhead das trocas de contexto
 - Tempo de resposta



Escalonamento em Sistemas Interativos

- **Prioridades + Round-robin**
 - **Definem-se classes de prioridade**
 - **Normalmente promove justiça apenas intra-classe**



Escalonamento em Sistemas Interativos

- **Escalonamento por prioridades**
 - Cada processo possui uma prioridade
 - O processo pronto com maior prioridade ganha a CPU
 - Processo de mais alta prioridade deixaria a CPU somente quando quisesse
 - Pode-se baixar a prioridade do processo executando, a cada tick de relógio
 - Ou estabelecer um quantum máximo
 - **Atribuição pode ser estática ou dinâmica**
 - **Comumente usado em conjunto com round-robin**

Escalonamento em Sistemas Interativos

- **Escalonamento Garantido**
 - **Fazer promessas reais aos usuários**
 - **Exemplo: Havendo n usuários conectados garantir a fração $1/n$ de CPU**
 - **Cálculo fica baseado no tempo de uso da CPU por cada processo desde sua criação**
 - **Processos com menor tempo de uso ganham CPU para se aproximar da fração garantida**
 - **Pode levar a grandes variações no tempo de resposta**

Escalonamento em Sistemas Interativos

- **Escalonamento por Loteria**
 - Prover garantias é interessante, mas com implementação mais complexa
 - Pode-se obter algo próximo a garantias mas com implementação mais simples
 - Processos ganham bilhetes de loteria. O prêmio é ganhar a CPU
 - A decisão de escalonamento consiste simplesmente da escolha aleatória de um bilhete
 - O quantum pode variar, conforme número de sorteios por segundo



Escalonamento em Sistemas de Tempo Real

- **Sistemas com propósito específico**
- **Resposta certa, porém tardia, é tão ruim quanto não ter resposta**
- **Tempo real crítico**
 - Prazos absolutos a serem cumpridos
- **Tempo real não-crítico**
 - Descumprimento ocasional de prazos é tolerável
- **Normalmente, processos de tempo real têm comportamento bem conhecido**
 - Curto processamento
- **Escalonador deve trabalhar para que todos os prazos sejam respeitados**



Escalonamento em Sistemas Interativos

- **Escalonamento por Loteria**
 - Como oferecer diferenciação de processos?
 - Dar mais bilhetes de loteria aos processos mais importantes, aumentando suas chances de ganhar
 - Processos recém-criados participam da loteria prontamente
 - Processos cooperativos podem trocar bilhetes entre si
 - Cliente/Servidor
 - Bilhetes de loteria acabam representando frações da CPU
 - Se um processo possui um fração f dos bilhetes, possui praticamente uma fração f da CPU
 - Muito útil como forma de oferecer garantias diferentes a processos com demandas diferentes



Escalonamento em Sistemas de Tempo Real

- **Sistema de tempo-real escalonável**
 - É possível descobrir se um sistema de tempo real é realmente escalonável com base na periodicidade dos eventos (tratados pelos processos)
 - m eventos periódicos
 - evento i ocorre em períodos de P_i unidades de tempo e requer C_i unidades de tempo para execução
 - Então a carga poderá ser tratada somente se

$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1$$

