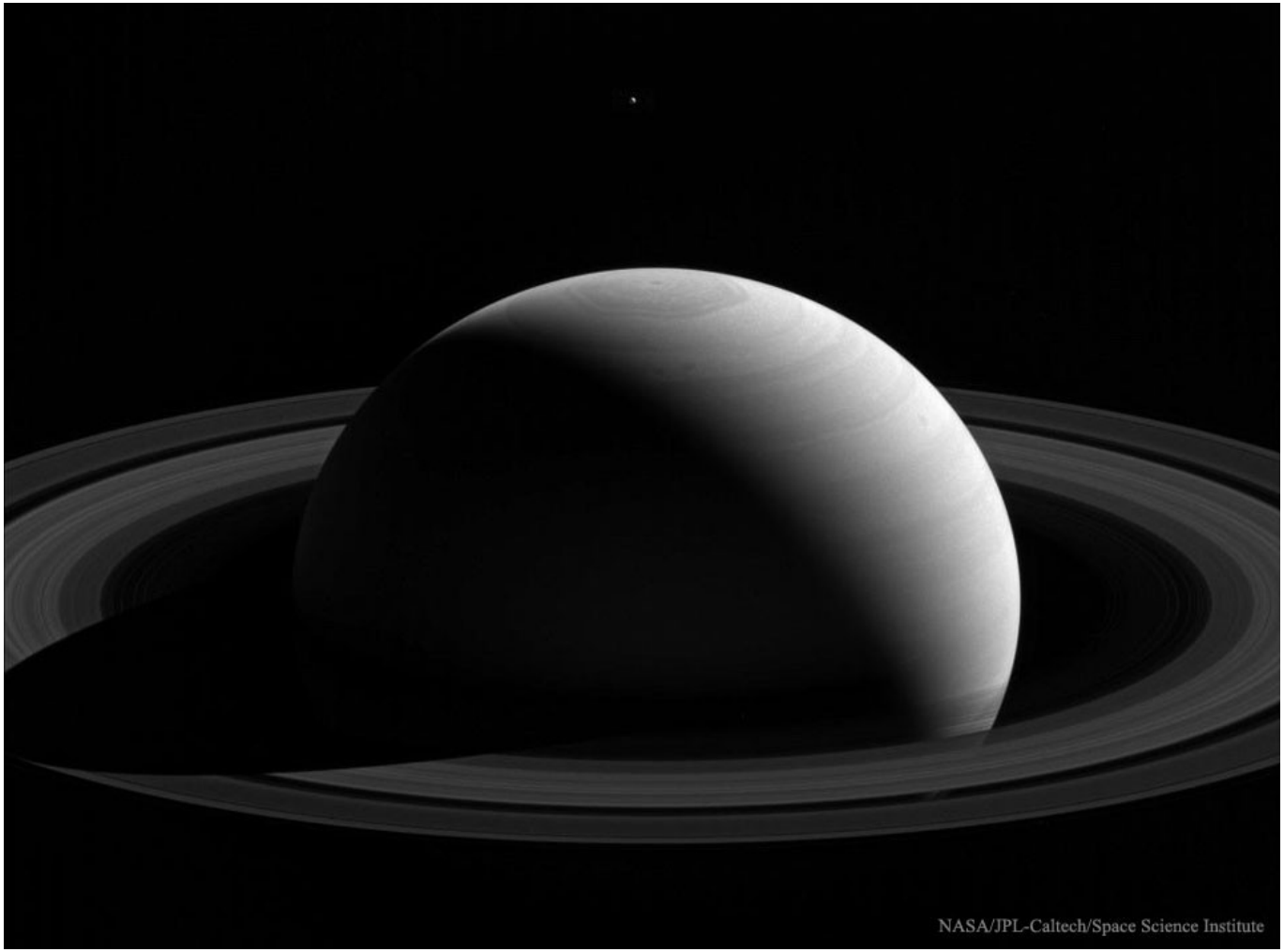
Sistemas Operacionais



Escalonamento

Prof. José Roberto Bezerra

Agenda

- Tipos de tarefas
- Preempção de tarefas
- Escalonamento de processos
- Algoritmos de escalonamento
- Troca de contexto
- Prioridades
- *Overhead*
- Aplicações STR

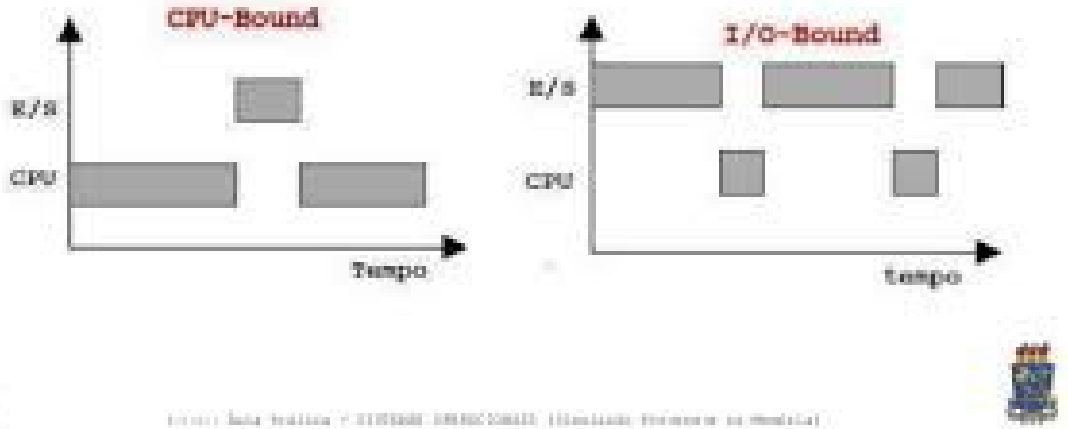
Tipos de Tarefas (temporais)

- Interativas
  - Ocorrem eventos externos por solicitação do usuário
- *Batch* (em lote)
  - São executadas em sequência, sem a intervenção do usuário
- Tarefas em Tempo Real
  - O tempo de execução pode ser previsto

Tipos de Tarefas (processamento)

- *CPU Bound*
  - Demandam a utilização da CPU na maior parte de seu tempo de execução
- I/O *Bound*
  - Durante a maior parte do tempo esperam dados oriundos dos dispositivos de E/S

CPU-Bound x I/O-Bound



Exemplos

- *CPU Bound*
  - Programas que realizam cálculos de previsão de tempo, determinação de trajetória de foguetes, modelos matemáticos complexos como aerodinâmica, termodinâmica e similares
- I/O *Bound*
  - Operações de rede em que o fluxo de dados é variável, coletores de dados, programas básicos como editores de texto, planilhas eletrônicas e similares

Preempção

Se diz que um algoritmo/sistema operacional é preemptivo quando um processo ganha controle da CPU e o mesmo pode perder o controle antes do término de sua execução

Escalonamento de Processos

- Quando um ou mais processos estão prontos para execução, o SO deve **decidir** qual dos processos deve entrar em execução primeiro
- A parte do SO responsável por esta decisão é chamada escalonador de processos e o algoritmo para esta finalidade é conhecido como algoritmo de escalonamento de processos

Sistemas Preemptivos

- Nos sistemas preemptivos, uma tarefa (processo) pode "perder" CPU caso:
  - O *quantum* de tempo termine
  - Processo executar uma chamada de sistema bloqueando o processo
  - Aconteça uma interrupção de E/S

Sistemas Não-Preemptivos

- Em sistemas não-preemptivos a execução de uma tarefa (processo) selecionada pelo SO permanece até seu final, exceto quando:
  - Processo executar uma chamada de sistema bloqueando o processo
  - Aconteça uma interrupção de E/S

Escalonamento de Processos

Como função do SO, o escalonamento de processos é uma tarefa essencial para a melhoria da eficiência do sistema como um todo, bem como evitar que processos “morram de fome” (starvation)

Critérios para escalonamento

- **Justiça** – Fazer com que todos os processos tenham acesso a CPU
- **Eficiência** – Manter a taxa de ocupação da CPU a maior possível (100%)
- **Tempo de Resposta** – Minimizar o tempo de resposta das tarefas interativas
- **Vazão** (throughput) – Maximizar a quantidade de processos por unidade de tempo

Dificuldade

Uma questão fundamental a se considerar na implementação de um escalonador de processos é que cada processo é uma entidade única e imprevisível

Clock

Para evitar que um processo permaneça tempo demais em execução, interrupções periódicas são geradas para realizar a troca de processos

Algoritmos de Escalonamento de Processos

FIFO

- *First In First Out*
- Também conhecido como FCFS (*First Come First Served*)
- Processos são escalonados de acordo com a ordem de chegada na fila de processos prontos
- Quando um processo bloqueia, o próximo da fila é escalonado e o processo bloqueado vai para o final da fila

Características

- Não preemptivo
- Algoritmo de fácil implementação
- Típico para sistemas em lote
- Não adequado quando processos orientados a CPU e orientados a E/S estão na fila

Shortest Job First

- Mais Curto Primeiro
- Tem como objetivo estabelecer uma sequência de execução de processos que levam menor tempo de execução para serem executados primeiro
- Também é necessário conhecer previamente o tempo de execução de cada tarefa

Características

- Utilizado para sistemas em lote
- Somente é viável quando todos os processos estão prontos para execução simultaneamente
-

Exemplo

- Considerando 4 processos A, B, C e D com tempos de execução 8, 4, 4 e 4 minutos, respectivamente, quanto seria o tempo de retorno de cada processo executados nessa ordem?

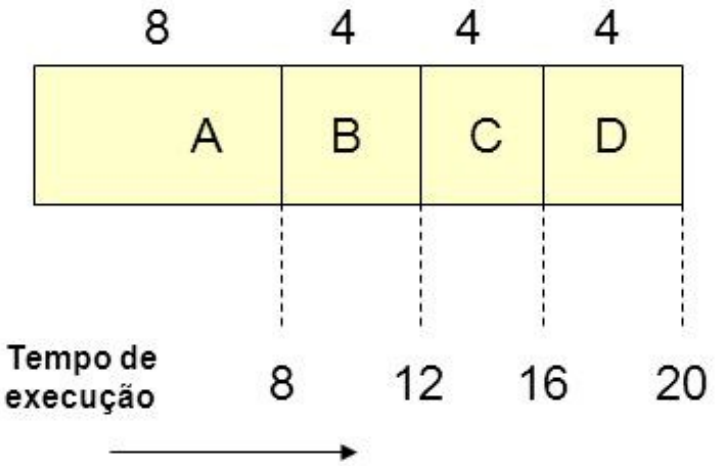
Exemplo

- E se a ordem de execução for alterada para B, C, D e A?

Escalonamento de Processos

O Menor Job Primeiro (Shortest-Job-First)

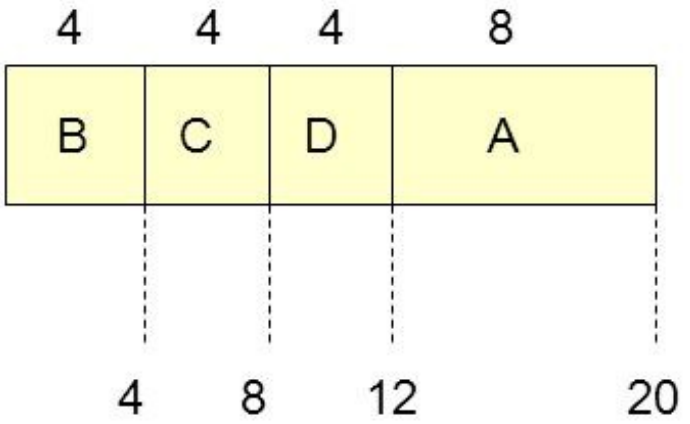
Agendamento FIFO



Média de retorno:

(8+12+16+20)/4 = 14 minutos

Agendamento Menor Job Primeiro



Média de retorno:

(4+8+12+20)/4 = 11 minutos

Shortest Remaining Time Next

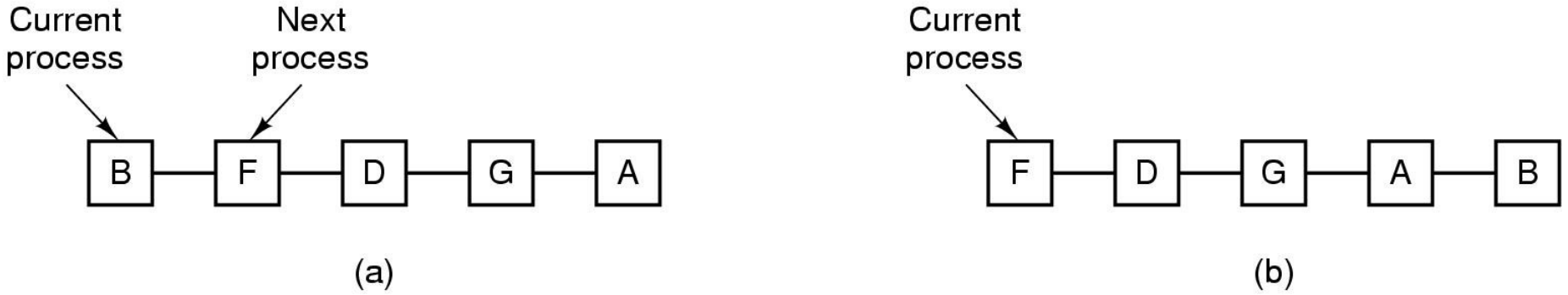
- Próximo de Menor Tempo Restante
- Escalona o processo que tenha o menor tempo de execução para ser concluído
- A cada novo processo que chega, seu tempo de execução para finalizar é comparado com o do processo em execução. Caso seja menor, o novo processo é escalonado

Características

- Versão preemptiva do Job Mais Curto Primeiro
- Bom desempenho para um conjunto de tarefas rápidas

Round Robin

- Um dos algoritmos mais antigos, simples, justos e ainda amplamente utilizado em sistemas interativos
- É atribuído um quantum de tempo a cada processo executar
- Ao final do quantum o processo sofre preempção pelo SO
- Outro processo pronto passa a ser executado



Características

- Implementação simples
  - Basta manter uma lista de processos prontos
- Utilizado em sistemas interativos
- O tamanho do *quantum* determina a eficiência do algoritmo
-

Troca de contexto

- Mudar de um processo para outro demanda do sistema um tempo extra para organização
  - Salvar/carregar registradores e mapas de memória
- A alternância entre processos presente nos sistemas preemptivos é chamada troca de contexto
- O tempo para a troca de contexto é ocasiona *overhead*

Overhead

- Supondo que uma troca de contexto típica dure 5ms (apesar de não ser possível estabelecer um valor preciso)
- Supondo ainda que o quantum de cada processo é de 20ms
- Após 20ms de trabalho a CPU gasta 5ms com operações administrativas, ou seja, 20% do tempo da CPU é gasto em overhead

Escalonamento por Prioridades

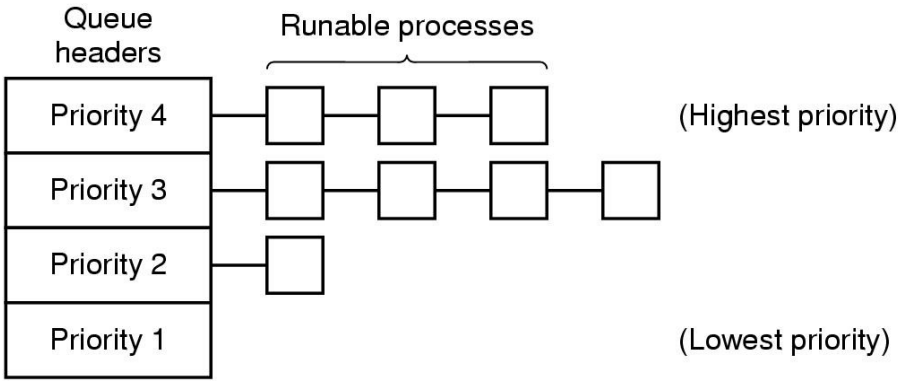
- Estabelece que diferentes processos não tem a mesma importância para o sistema
- Cada processo possui uma prioridade associada a seu nível de importância
- O processo pronto para executar que tenha maior prioridade ganha a CPU
- Caso não seja estabelecido um mecanismo de troca de prioridades periódico, os processos com menor prioridade podem sofrer *starvation*

Round Robin com Prioridades

- Escalonamento circular pressupõe que todos os processos tem a mesma importância
- Não há preferência por um processo ou grupo de processos
- Por exemplo, processo que envia emails em segundo plano deve possuir menor prioridade que o processo que exibe o vídeo na tela
- É conveniente agrupar processos por classes de prioridades

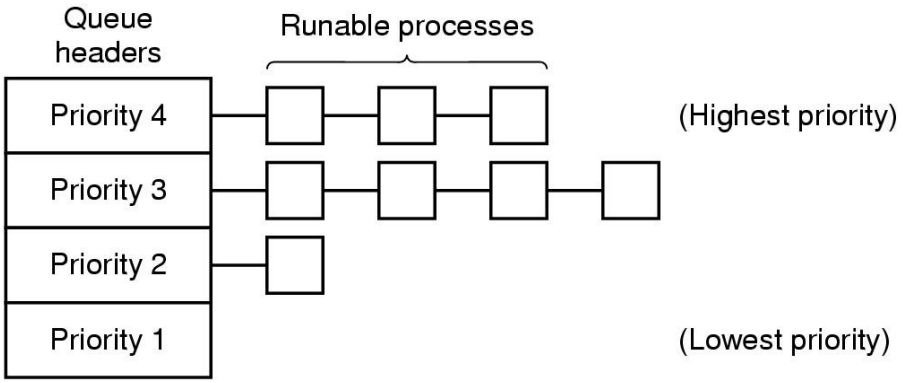
Round Robin com Prioridades

- Dentro de cada classe é adotado o escalonamento circular
- Processos de uma classe inferior serão escalonados quando a fila de classe superior estiver vazia



Round Robin com Prioridades

- Para evitar que processos com alta prioridade executem indefinidamente, o escalonador de processos pode decrementar a prioridade do processo em execução a cada ciclo de *clock*, forçando a troca de processos



Selfish Round Robin

- Alternância Circular Egoísta
- Antes que um processo possa ser escalonado, o mesmo passa um período envelhecendo (gradualmente aumentando seu nível de prioridade)
- Ao atingir um determinado grau de maturidade o processo entra na disputa do escalonamento por Round Robin
- Prioriza os processos mais antigos

Aplicações de Sistemas em Tempo Real

- Um sistema em tempo real é aquele em que o tempo é essencial para o resultado
- O sistema deve apresentar resultados dentro de um intervalo de tempo previsível
- Em geral, vários dispositivos físicos externos ao computador geram estímulos (sinais) e o computador deve reagir apropriadamente a eles dentro de um intervalo de tempo

Aplicações de Sistemas em Tempo Real

- Aplicações
  - Piloto automático de aeronaves
  - Controle de trajetória de foguetes
  - Controle de robôs em fábricas
  - Monitoramento do sistema elétrico

CD *Player*

- O processo responsável por um CD *player* obtém os bits oriundos do drive e precisa realizar a conversão Digital/Analógica e transformar os bits em som em um intervalo de tempo muito curto
- Se o processamento envolvido for longo a música será tocada de maneira diferente

Tipos de STR

- Tempo Real Crítico
  - Há prazos em tempo absoluto que devem ser cumpridos
- Tempo Real Não-Crítico
  - O descumprimento ocasional de um prazo não é desejável, mas é tolerável

Eventos

- Os eventos que um STR deve responder podem ocorrer em intervalos regulares (periódicos) ou de forma imprevisível (aperiódicos)
- Dependendo do tempo necessário para tramento dos eventos pode não ser possível tratar todos os eventos
- Apenas um STR escalonável pode tratar adequadamente a ocorrência dos eventos

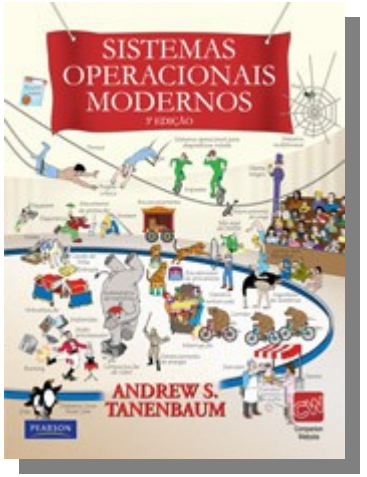
STR Escalonável

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

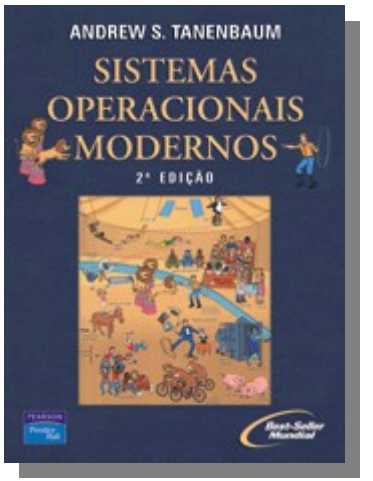
- m é a quantidade de eventos
- P_i é o período na qual o evento i ocorre
- C_i é tempo necessário para o sistema tratar este evento

Bibliografia

- Tanenbaum, Andrew S. Sistemas Operacionais Modernos. 3a. Ed. Pearson, 2010 (Seção 2.4)



- Tanenbaum, Andrew S. Sistemas Operacionais Modernos. 2a. Ed. Pearson, 2003 (Seção 2.1)



Dúvidas e Perguntas

FIM