



Sistemas Operacionais

Prof. Jó Ueyama

Apresentação baseada nos slides da Profa. Dra. Kalinka Castelo Branco, do Prof. Dr. Antônio Carlos Sementille e da Profa. Dra. Luciana A. F. Martimiano e nas transparências fornecidas no site de compra do livro “Sistemas Operacionais Modernos”



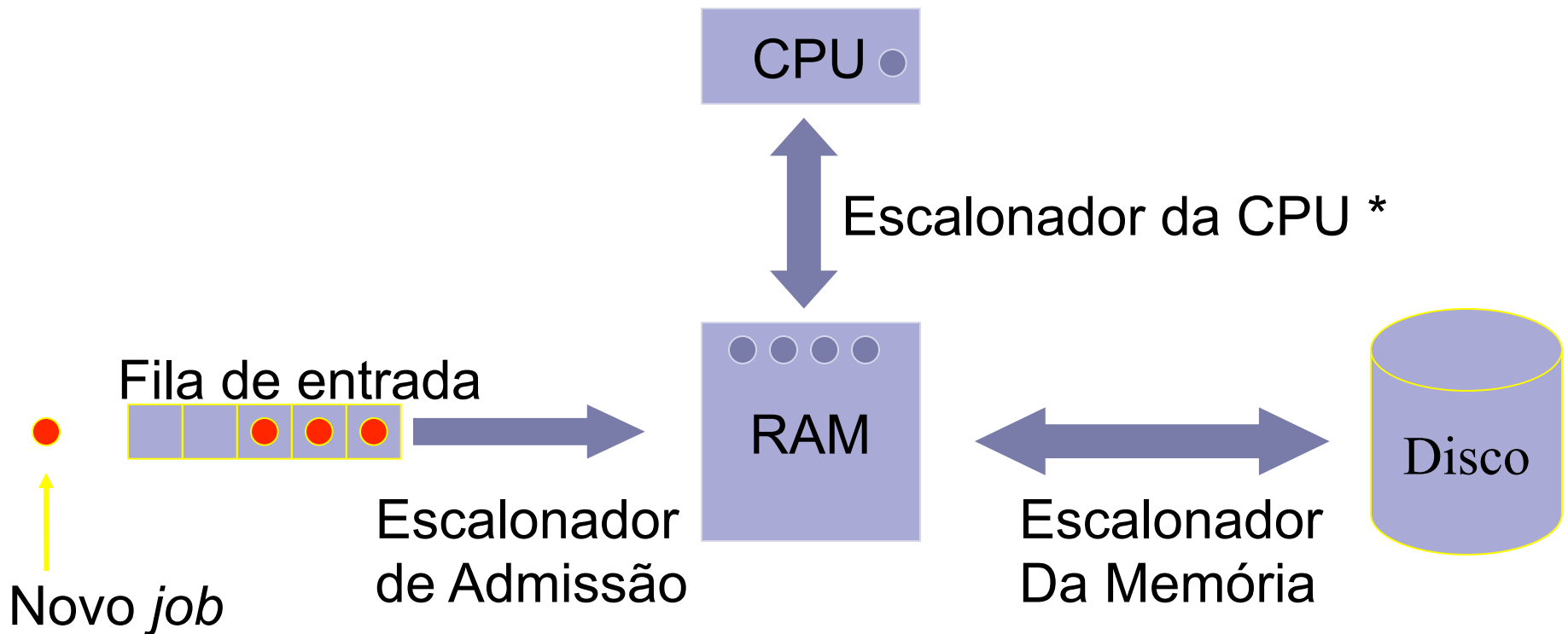
Aula de Hoje (conteúdo detalhado)

- 1. Escalonamento em Batch**
- 2. Algoritmos de Escalonamento em Sistemas Batch**
- 3. Algoritmos de Escalonamento em Sistemas Interativos**

Escalonamento de Processos

Sistemas em *Batch*

■ Escalonamento *Three-Level*



Escalonamento de Processos

Sistemas em *Batch*

■ Escalonamento *Three-Level*

- **Escalonador de admissão**: processos menores primeiro; processos com menor tempo de acesso à CPU e maior tempo de interação com dispositivos de E/S;
- **Escalonador da Memória**: decisões sobre quais processos vão para a MP:
 - A quanto tempo o processo está esperando?
 - Quanto tempo da CPU o processo já utilizou?
 - Qual o tamanho do processo?
 - Qual a importância do processo?
- **Escalonador da CPU**: seleciona qual o próximo processo a ser executado;



Aula de Hoje (conteúdo detalhado)

1. Escalonamento em Batch
- 2. Algoritmos de Escalonamento em Sistemas Batch**
- 3. Algoritmos de Escalonamento em Sistemas Interativos**

Escalonamento de Processos

Sistemas em *Batch*

- Algoritmos para Sistemas em *Batch*:
 - *First-Come First-Served (ou FIFO)*;
 - *Shortest Job First (SJF)*;
 - *Shortest Remaining Time Next (SRTN)*;

Escalonamento de Processos

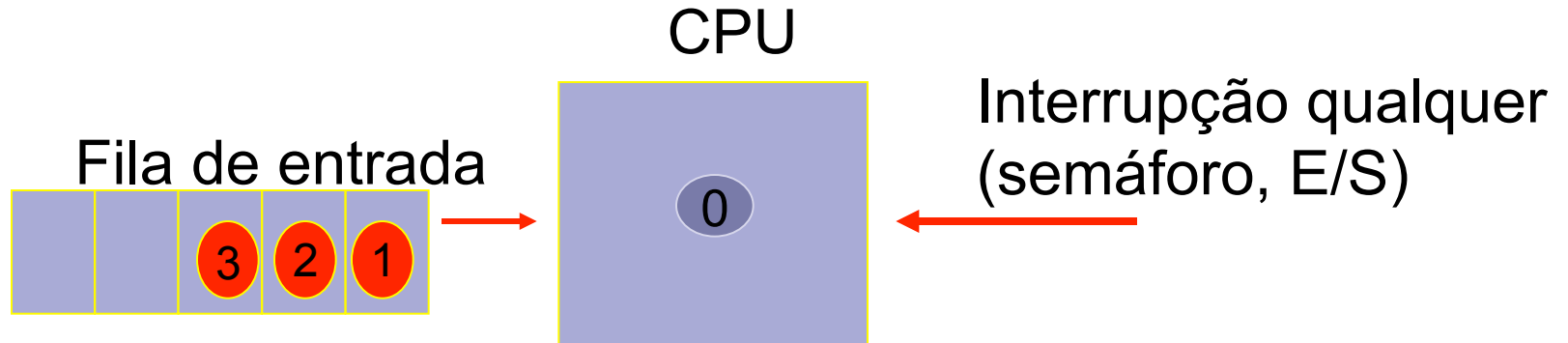
Sistemas em *Batch*

■ Algoritmo *First-Come First-Served*

- Não-preemptivo;
- Processos são executados na CPU seguindo a ordem de requisição;
- Fácil de entender e programar;
- Desvantagem:
 - Ineficiente quando se tem processos que demoram na sua execução;

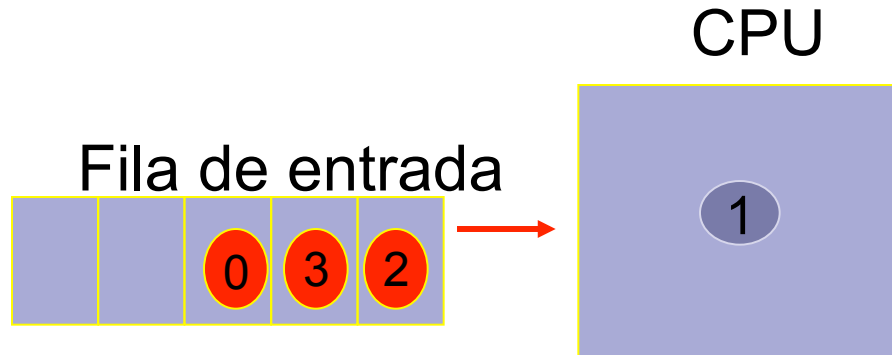
Escalonamento de Processos Sistemas em *Batch*

■ Algoritmo *First-Come First-Served*



Escalonamento de Processos Sistemas em *Batch*

- Algoritmo *First-Come First-Served*



CPU não controla o tempo dos processos!
(não-preemptivo)

Escalonamento de Processos

Sistemas em *Batch*

■ Algoritmo *Shortest Job First*

- Não-preemptivo;
- Possível prever o tempo de execução do processo;
- Menor processo é executado primeiro;
- Menor *turnaround* (tempo desde a submissão até a execução);
- Desvantagem:
 - Baixo aproveitamento quando se tem poucos processos prontos para serem executados;

Escalonamento de Processos

Sistemas em *Batch*

■ Algoritmo *Shortest Job First*

A → a

B → b+a

C → c+b+a

D → d+c+b+a

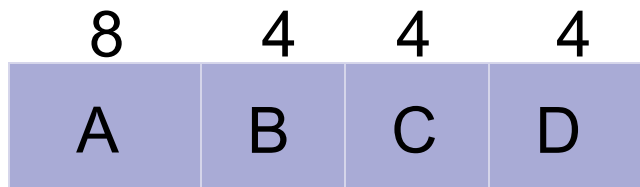
Tempo médio-*turnaround* $(4a+3b+2c+d)/4$

Contribuição → se $a < b < c < d$ tem-se o mínimo tempo médio;

Escalonamento de Processos

Sistemas em *Batch*

■ Algoritmo *Shortest Job First*



Em ordem:

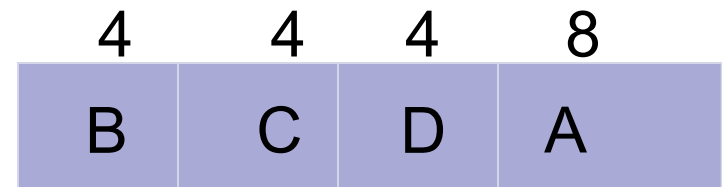
Turnaround A = 8

Turnaround B = 12

Turnaround C = 16

Turnaround D = 20

Média $\rightarrow 56/4 = 14$



Menor *job* primeiro:

Turnaround B = 4

Turnaround C = 8

Turnaround D = 12

Turnaround A = 20

Média $\rightarrow 44/4 = 11$

$$(4a+3b+2c+d)/4$$

Número de
Processos

Escalonamento de Processos

Sistemas em *Batch*

- Algoritmo *Shortest Remaining Time Next*
 - Preemptivo;
 - Processos com menor tempo de execução são executados primeiro;
 - Se um processo novo chega e seu tempo de execução é menor do que do processo corrente na CPU, a CPU suspende o processo corrente e executa o processo que acabou de chegar;
 - Desvantagem: processos que consomem mais tempo podem demorar muito para serem finalizados se muitos processos pequenos chegarem!



Aula de Hoje (conteúdo detalhado)

1. Escalonamento em Batch
2. Algoritmos de Escalonamento em Sistemas Batch
- 3. Algoritmos de Escalonamento em Sistemas Interativos**



Escalonamento de Processos

Sistemas Interativos

- Algoritmos para Sistemas Interativos:
 - *Round-Robin*;
 - Prioridade;
 - Múltiplas Filas;
 - *Shortest Process Next*;
 - Garantido;
 - *Lottery*;
 - *Fair-Share*;
- Utilizam escalonamento em dois níveis (escalonador da CPU e memória);

Escalonamento de Processos

Sistemas Interativos

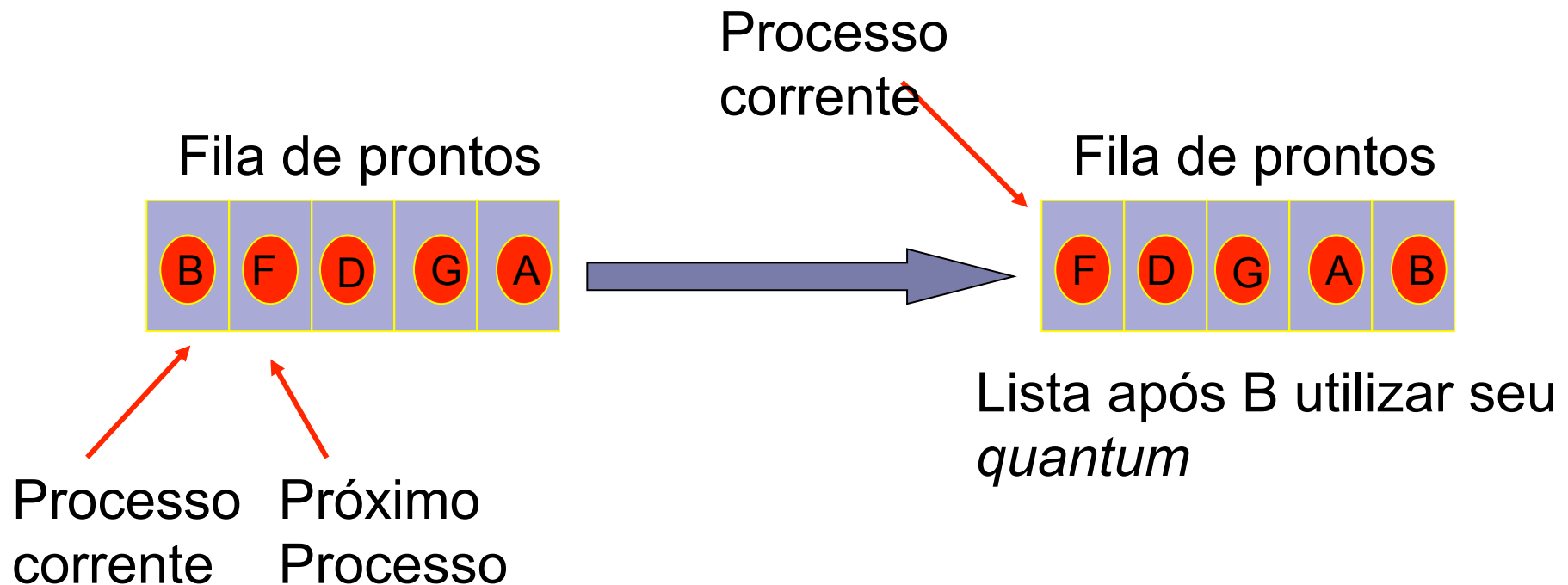
■ Algoritmo *Round-Robin*

- Antigo, mais simples e mais utilizado;
- Preemptivo (quantum, I/O, system call pelo processo);
- Cada processo recebe um tempo de execução chamado *quantum*; ao final desse tempo, o processo é suspenso e outro processo é colocado em execução;
- Escalonador mantém uma lista de processos prontos;

Escalonamento de Processos

Sistemas Interativos

- Algoritmo *Round-Robin*
- Diferença com o FIFO?



Escalonamento de Processos

Sistemas Interativos

■ Algoritmo *Round-Robin*

- Tempo de chaveamento de processos;
- *quantum*: se for muito pequeno, ocorrerem muitas trocas diminuindo, assim, a eficiência da CPU; se for muito longo o tempo de resposta é comprometido;

Escalonamento de Processos

Sistemas Interativos

■ Algoritmo *Round-Robin*:

Exemplos:

$$\Delta t = 4 \text{ mseg}$$

$x = 1 \text{ mseg} \rightarrow 25\%$ de tempo de CPU é perdido \rightarrow menor eficiência

$$\Delta t = 100 \text{ mseg}$$

$x = 1 \text{ mseg} \rightarrow 1\%$ de tempo de CPU é perdido
 \rightarrow Tempo de espera dos processos é maior

quantum razoável: 20-50 mseg

Escalonamento de Processos

Sistemas Interativos

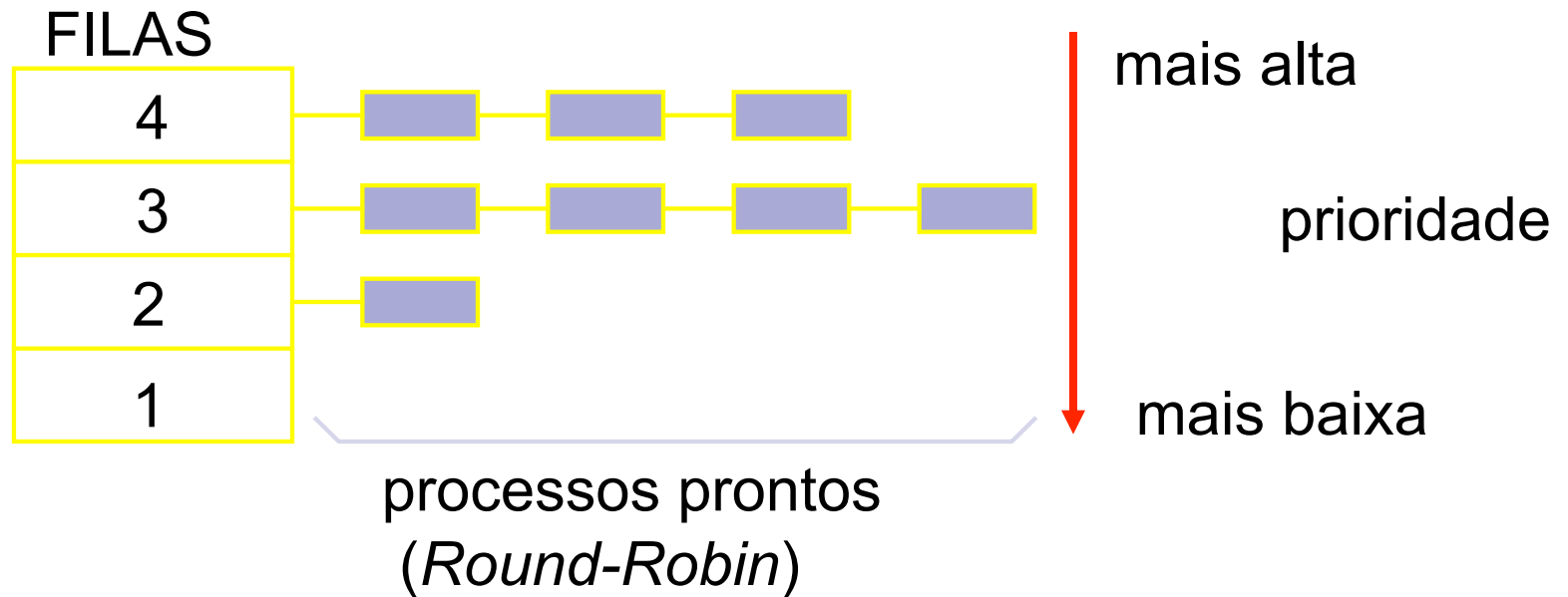
■ Algoritmo com Prioridades

- Cada processo possui uma prioridade → os processos prontos com maior prioridade são executados primeiro;
- Prioridades são atribuídas dinâmica ou estaticamente;
- Classes de processos com mesma prioridade;
- Preemptivo;

Escalonamento de Processos

Sistemas Interativos

■ Algoritmo com Prioridades



Escalonamento de Processos

Sistemas Interativos

■ Algoritmo com Prioridades

- Como evitar que os processos com maior prioridade sejam executado indefinidamente?
 - Diminuir a prioridade do processo corrente e troca-lo pelo próximo processo com maior prioridade (chaveamento);
 - O problema da inaninação dos processos de prioridades inferiores
 - Cada processo possui um *quantum*;

Escalonamento de Processos

Sistemas Interativos

■ Múltiplas Filas:

- CTSS (*Compatible Time Sharing System*);
- Classes de prioridades;
- Cada classe de prioridades possui *quanta* diferentes;
- Assim, a cada vez que um processo é executado e suspenso ele recebe mais tempo para execução;
- Preemptivo;

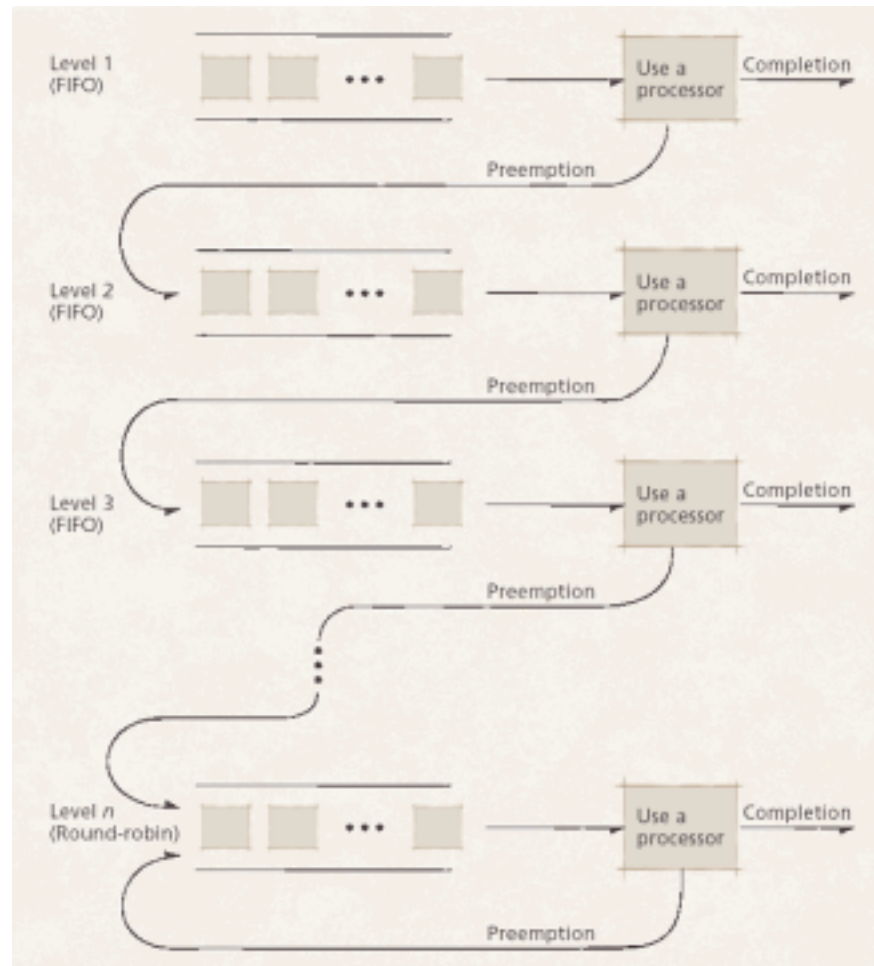
Escalonamento de Processos

Sistemas Interativos

■ Múltiplas Filas:

- Ex.: um processo precisa de 100 *quanta* para ser executado;
 - Inicialmente, ele recebe um *quantum* para execução;
 - Objetivo é reduzir as trocas de processos;
 - Das próximas vezes ele recebe, respectivamente, 2, 4, 8, 16, 32 e 64 *quanta* (7 chaveamentos) para execução;
 - Quanto mais próximo de ser finalizado, menos frequente é o processo na CPU → eficiência

Escalonamento de Múltiplas Filas



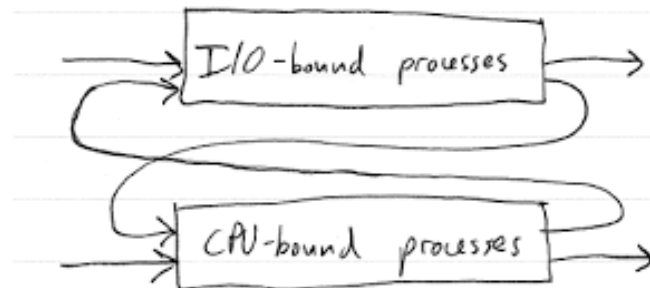
Escalonamento de Processos

Sistemas Interativos

- Sistemas Linux e Windows

- *Multilevel feedback queue*

- Dá preferência a processos curtos
- Dá prioridades para processos I/O bound
- Estuda o processo e escalona de acordo com o que inferiu
- Cada fila usa um escalonamento round-robin
- Os I/Os promovem os processos para as filas com maior prioridade



Escalonamento de Processos

Sistemas Interativo

■ Algoritmo *Shortest Process Next*

- Mesma idéia do *Shortest Job First*;
- Processos Interativos: não se conhece o tempo necessário para execução;
- Como empregar esse algoritmo: ESTIMATIVA de TEMPO!
- Verificar o comportamento passado do processo e estimar o tempo.

Escalonamento de Processos

Sistemas Interativo

■ Outros algoritmos:

□ Algoritmo Garantido:

- Garantias são dadas aos processos dos usuários:

- n usuários $\rightarrow 1/n$ do tempo de CPU para cada usuário;

□ Algoritmo *Lottery*:

- Cada processo recebe “*tickets*” que lhe dão direito de execução;
- Processo com maior prioridade podem receber mais “*tickets*”.

Escalonamento de Processos

Sistemas Interativo

■ Algoritmo *Fair-Share*:

- O dono do processo é levado em conta;
- Se um usuário A possui mais processos que um usuário B, o usuário A terá prioridade no uso da CPU;

Usuário 1 → A, B, C, D

- Usuário 2 → E
- Garantia de 50%

Circular → A, E, B, E, C, E, D, E

50% a mais para Usuário 1 → A, B, E, C, D, E



Escalonamento de Processos Sistemas em Tempo Real

- Tempo é um fator crítico; possui um deadline
- Sistemas críticos:
 - Aviões;
 - Hospitais;
 - Usinas Nucleares;
 - Bancos;
 - Multimídia;
- Ponto importante: obter respostas em atraso é tão ruim quanto não obter respostas;

Escalonamento de Processos

Sistemas em Tempo Real

■ Tipos de STR:

- **Hard Real Time**: atrasos não são tolerados;

- Aviões, usinas nucleares, hospitais;

- **Soft Real Time**: atrasos são tolerados;

- Bancos; Multimídia;

■ Programas são divididos em vários processos;

■ Eventos causam a execução de processos:

- **Periódicos**: ocorrem em intervalos regulares de tempo;

- **Aperiódicos**: ocorrem em intervalos irregulares de tempo;

Escalonamento de Processos

Sistemas em Tempo Real

- Algoritmos podem ser estáticos ou dinâmicos;
 - **Estáticos**: decisões de escalonamento antes do sistema começar;
 - Informação disponível previamente;
 - **Dinâmicos**: decisões de escalonamento em tempo de execução;



Perguntas?

- Concluimos a Seção 2.4 do livro SO (4a. Edição – a última lançada)