



Universidade Federal de Uberlândia
Faculdade de Computação
Sistemas Operacionais



Escalonamento de Processos

Prof. Dr. Marcelo Zanchetta do Nascimento

Roteiro

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;
- Leituras sugeridas.

Conceito

- Os processos alternam entre 2 estados: CPU e E/S;
- Começa com um burst (surto) de CPU ou burst de E/S;
 - Programa I/O-bound: muitos burst de CPU curtos;
 - Programa CPU-bound: alguns burst de CPU longos.
- Importante **na seleção de um algoritmo de escalonamento.**

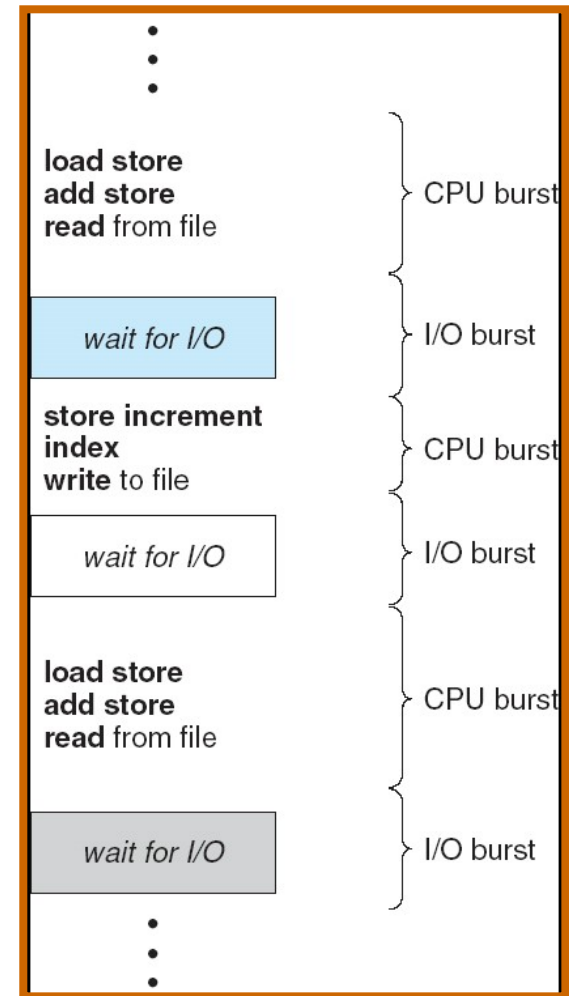


Figura 1: Ciclo de burts

Conceito

Situações nas quais escalonamento é necessário:

- Um novo processo é criado e passa para estado pronto;
- Um processo terminou sua execução e um outro processo “pronto” deve ser executado;
- Processo é bloqueado (dependência de E/S) e outro deve ser executado;
- Interrupção de E/S, o escalonador deve decidir por:
 - Executar o processo que estava esperando esse evento;
 - Continuar executando o processo em execução.

Roteiro

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;

Despachante

- Componente envolvido na função do escalonador de CPU;
- Módulo que dá o controle da CPU ao processo;
- Função:
 - **Trocar o contexto;**
 - **Trocar para o modo usuário;**
 - **Desviar para local apropriado no programa de usuário.**
- Chamado durante a troca de processo;
- Tempo gasto para interromper um processo e iniciar a execução de outro definido por **latência de despacho.**

Roteiro

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;

Objetivos

- Maximizar a taxa de utilização da CPU:
 - Maximizar a vazão do sistema.
- Minimizar o tempo de execução (“turnaround time”):
 - Turnaround: tempo gasto entre a criação do processo e o seu encerramento
- Minimizar o tempo de espera (“waiting time”):
 - Waiting time: tempo ocioso pela tarefa na fila de prontas;
- Minimizar o tempo de resposta (“response time”):
 - Response time: tempo entre requisição e resposta para sistemas interativos.

Algoritmos de Escalonamento

O esquema de escalonamento:

- **Não preemptivo:** Não permite interrupções externa à tarefa até que seja liberado pelo seu término ou pela troca para o estado esperando (p.ex., uma operação de E/S).
 - Exemplo: Windows 3.x.
- **Preemptivo:** Interrompe a execução de uma tarefa e transfere a CPU para outro;
 - Necessário mecanismo para coordenar acesso aos dados;
 - Exemplo: Windows, Mac OS X, etc.

Escalonamento em sistema Batch

- **FCFS** (first-come first-served);
- Processo que solicita a CPU primeiro, a recebe primeiro.
- Quando processo entra na fila de pronto, seu BCP é ligado ao final.
- Quando CPU é liberada, ela é alocada ao processo no início da fila.
- Tempo de espera normalmente longo.
- **Efeito comboio:** todos os outros processos aguardam quando há um grande processo em execução na CPU.

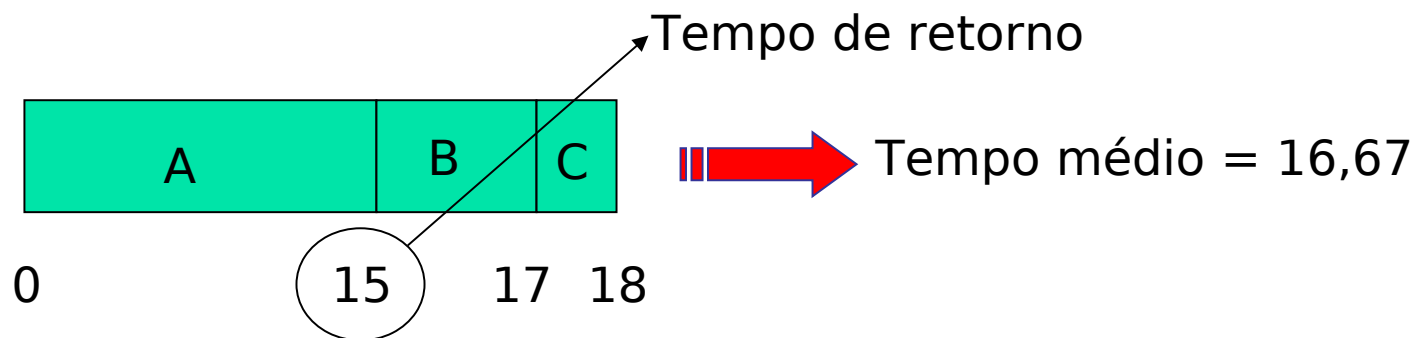
FCFS (first-come first-served);

Exemplo: prog A (15 milissegundos),
B (2 milissegundos)
C (1 milissegundo).

**Tempo pode variar
substancialmente se os
tempos de retorno da CPU
variarem muito**

Suponha que os esses processos cheguem na seguinte ordem (no mesmo instante 0) : P 1 , P 2 , P 3

A linha de tempo (diagrama de Gantt) é



Tempo de chegada 0

FCFS (first-come first-served);

Chegada dos processos

- P1: 15
- P2: 2
- P3: 1

Turnaround time

- P1: 15
- P2: 17
- P3: 18

Tempos de espera na fila (Waiting Time)

- P1: 0 (primeiro a chegar)
- P2: 15
- P3: $(15+2) = 17$

Média do Turnaround time

$$(15+17+18)/3 = 16,67$$

FCFS (first-come first-served);

Exercício 1

Calcule o waiting time, o turnaround time e média para os seguintes casos:

- P1:4 (1),
- P2:7 (2),
- P3:2 (3),
- P4:10 (4)

Ordem de chegada na fila dada pela identificação do processo.

FCFS (first-come first-served);

Exercício 2

Calcule waiting time, turnaround time e média para os seguintes casos:

Processo	Tempo	Tempo de chegada
P1	8	0
P2	10	5
P3	8	10
P4	3	11
P5	10	15

Ordem de chegada na fila dada pela identificação do processo

Shortest job first (SJF)

- Associa a cada processo o tamanho do próximo burst de CPU do processo;
- A CPU atribui o processo que tem o próximo retorno de CPU menor (burst);
- Em caso de empate (tempo de retorno) dos processos, o algoritmo FCFS é utilizado;
- Quanto menor o tempo de execução, maior a prioridade dada pelo escalonador para execução.

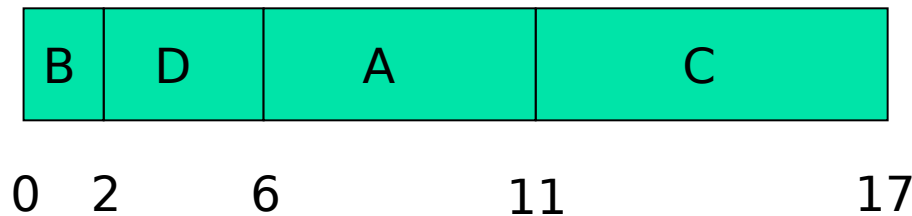
Shortest job first (SJF)

Exemplo: Programas
ciclo de CPU

A	B	C	D
5	2	6	4

Tempo de chegada 0

A linha de tempo (diagrama de Gantt) é



Tempo médio de
retorno
(turnaround) = 9,0

Se fosse utilizado o FCFS, o tempo médio seria de 10,5

Shortest job first (SJF)

Exercício 3

a) Calcule o waiting time, turnaround time e a média para os processos que chegam no mesmo instante nessa ordem:

P1:4,

P2:7,

P3:5,

P4:13.

b) Se P1 chega no instante 0, P2 no instante 2, P3 no instante 5 e P4 no instante 8. Calcule novamente as métricas.

CPU burst

Tamanho do Próximo CPU burst

- A real dificuldade do algoritmo é conhecer o tamanho da próxima requisição de CPU.
- Para **escalonamento de longo prazo** num sistema batch, podemos usar como tamanho o limite de tempo de CPU especificado pelo usuário quando da submissão do processo.
- No nível de **escalonamento de curto prazo** sua implementação pode ser apenas aproximada, já que não há como saber o tamanho da próxima requisição de CPU.

CPU burst

Tamanho do Próximo CPU burst

- Uma maneira de se aproximar do SJF é prever o tamanho do próximo CPU burst.
- Normalmente, isso é feito usando uma média exponencial das medidas dos bursts anteriores.

1. t_n = actual length of n^{th} CPU burst
2. τ_{n+1} = predicted value for the next CPU burst
3. $\alpha, 0 \leq \alpha \leq 1$
4. Define :

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n.$$

CPU burst

Tamanho do Próximo CPU burst

- O parâmetro α controla o peso relativo do histórico recente e passado na equação
- Se $\alpha = 0$
 - $\tau_{n+1} = \tau_n$
 - Histórico recente não é considerado relevante.
- Se $\alpha = 1$
 - $\tau_{n+1} = \tau_n$
 - Apenas o último CPU burst é levado em conta.

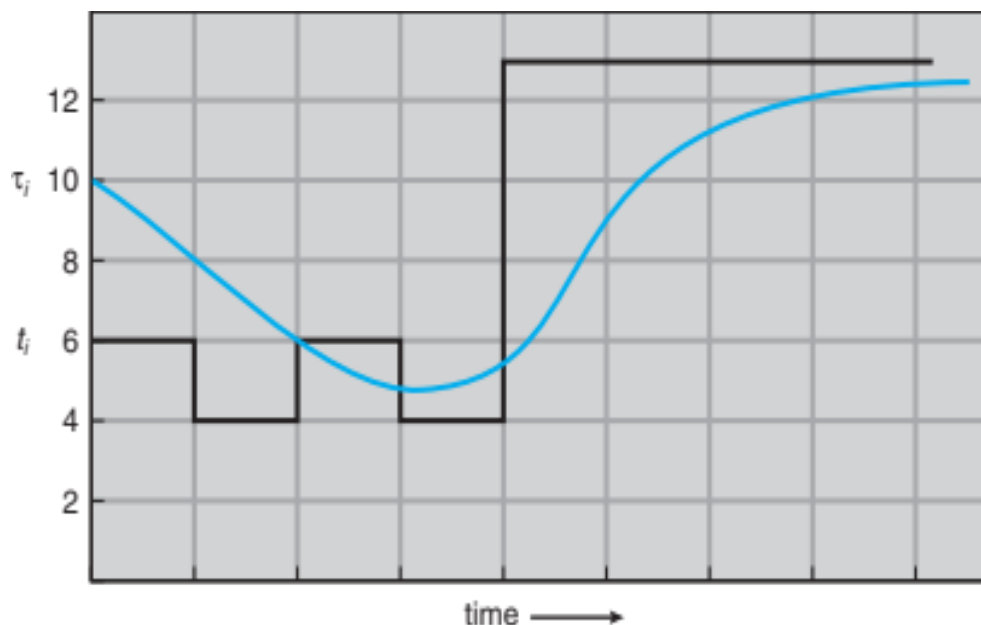
CPU burst

Tamanho do Próximo CPU burst

■ $\alpha = 0,5$

■ $\tau_0 = 10$

$$\tau_{n+1} = \alpha t_n + (1 - \alpha)\tau_n.$$



Estimativa
(predição)

CPU burst (t_i)	6	4	6	4	13	13	13	...	
"guess" (τ_i)	10	8	6	6	5	9	11	12	...

Figura 2: Previsão de duração do próximo ciclo de CPU

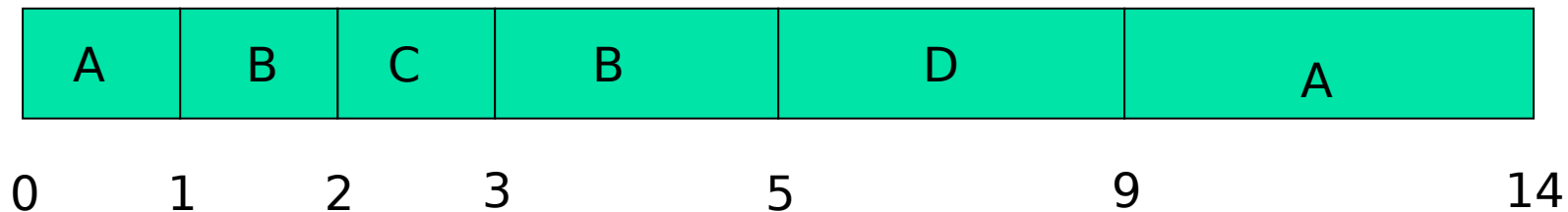
Shortest remaining time next (SRTN)

- Um algoritmo SJF preemptivo que permitirá processos com menor tempo de execução;
- Esse processo é conhecido como shortest-remaining time next scheduling.
- Preemptivo: se chega um novo processo com CPU burst menor que o tempo remanescente do processo corrente ocorre a preempção.
- Requer **conhecimento prévio do tempo** de CPU (sistema em lote).

Shortest remaining time next (SRTN)

Exemplo: Tempo de chegada 0 1 2 3
Programa A B C D
Ciclo de CPU 6 3 1 4

A linha de tempo (diagrama de Gantt) é



Programa A B C D

Tempo retorno (14-0) (5-1) (3-2) (9-3) Tempo médio = ?

SJF - Qual seria esse o tempo de retorno?

Escalonamento em Sistema Interativo

- Alternância circular (*Round-Robin*);
- Prioridades;
- Filas múltiplas;
- Garantido;
- Loteria;
- Fração justa (*Fair-Share*).

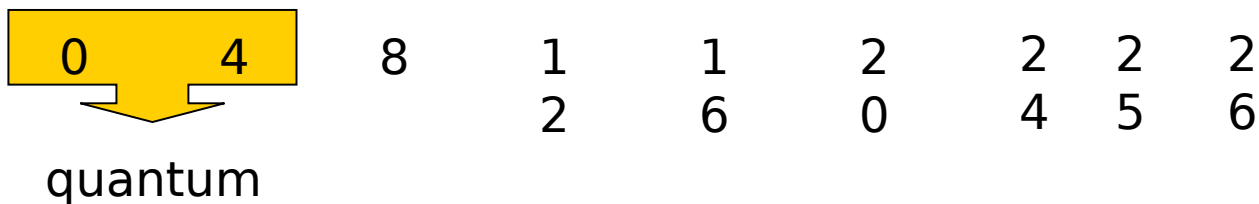
Round-Robin (RR)


- Algoritmo alternância circular
 - Tempo compartilhado;
 - Lista de processos é implementada como uma **fila circular**;
 - Preemptivo;
 - Cada processo recebe um tempo de execução chamado *quantum*; ao final desse tempo, é preemptado e reinserido na cauda da fila de pronto;
 - Escalonador mantém uma lista de processos prontos.

Round Robin

- Trabalha **primeiro a chegar**, primeiro a ser atendido.

Ex.: Tempo de chegada 0 1 2 3
 Programa A B C D
 Ciclo de CPU 8 4 9 5



Programa A B C D
 Tempo retorno 20 7 24 22  Tempo médio = 18,25

Round Robin

Exercício 4

Calcule waiting time, turnaround time e média para os seguintes casos (quantum 5):

Processo	Tempo	Tempo de chegada
P1	10	0
P2	5	3
P3	8	4
P4	3	10
P5	10	12

Prioridade

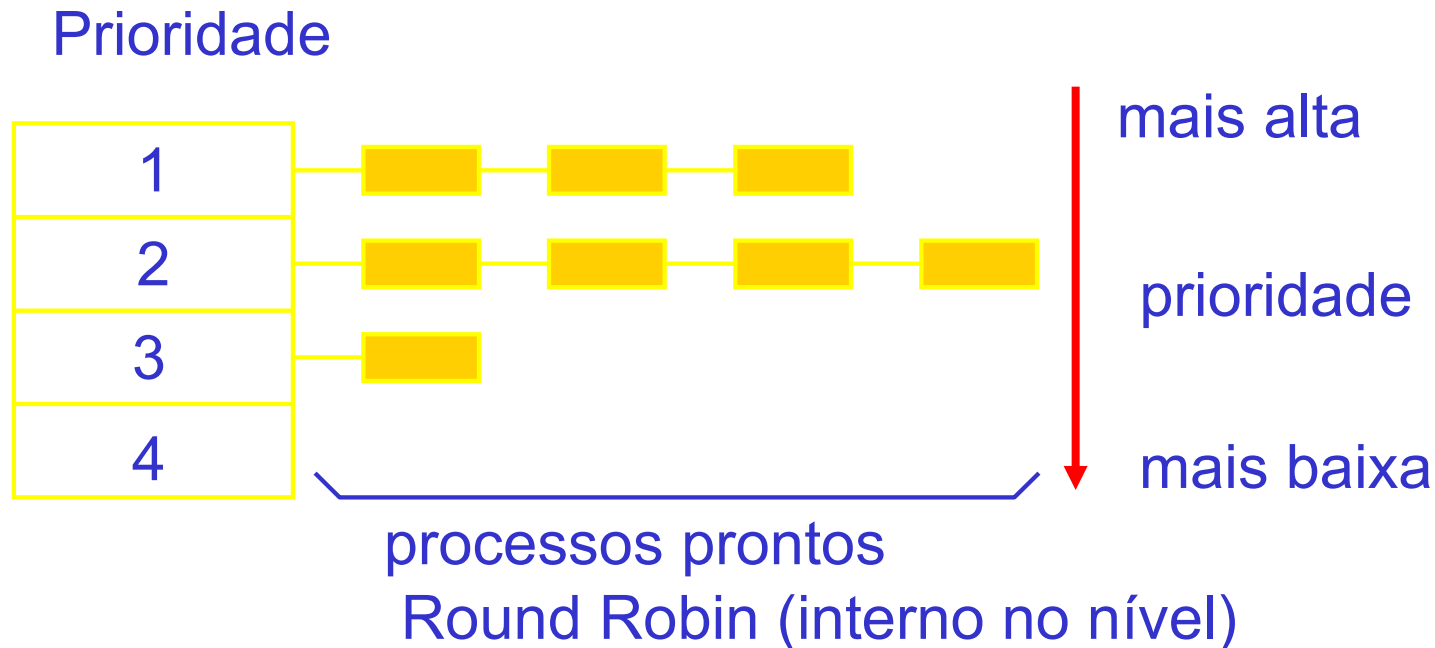
- Associa-se a **cada processo** um **nível de prioridade**, de acordo com os interesses do **sistema**.
- Um número inteiro é associado a cada processo, refletindo a sua prioridade no sistema.
- A CPU é alocada ao processo de maior valor de prioridade na fila de prontos.
- Obs: normalmente, menor valor representa maior prioridade.
- Estratégia muito usada em S.O. de tempo real.
- Problema: “starvation”
 - Processos de baixa prioridade podem nunca executar.

Prioridade

- Prioridades podem ser definidas interna ou externamente.
- Interna:
 - Usa alguma medida (ou uma combinação delas) para calcular o valor da prioridade.
 - Exemplo: limite de tempo, requisitos de memória, n° de arquivos abertos, razão entre average I/O burst e average CPU burst, etc.
- Externa:
 - Definida por algum critério externo ao S.O (tipo do processo, departamento responsável, custo, etc.)

Prioridade

- Agrupar processos em classes de prioridades
- Exemplo: cada classe usa o escalonamento circular



Prioridade

Exercício 5

Calcule o waiting time, turnaround time e média para os seguintes casos (prioridade com ordem mais baixa):

Processo	Ordem (Prioridade)	Tempo	Tempo de chegada
P1	3	10	0
P2	2	5	2
P3	1	8	4
P4	4	3	10
P5	5	10	12

Filas Múltiplas (Multiple Queues)

- **Filas múltiplas (multinível):**
 - Divide a fila de processos prontos em várias filas separadas;
 - Cada fila tem seu próprio algoritmo de escalonamento (ex.: FCFS, Prioridade, etc);
 - Escalonamento entre as filas:
 - Exemplo: Primeiro plano e segundo plano.
 - Preemptivo.

Filas Múltiplas (Multiple Queues)

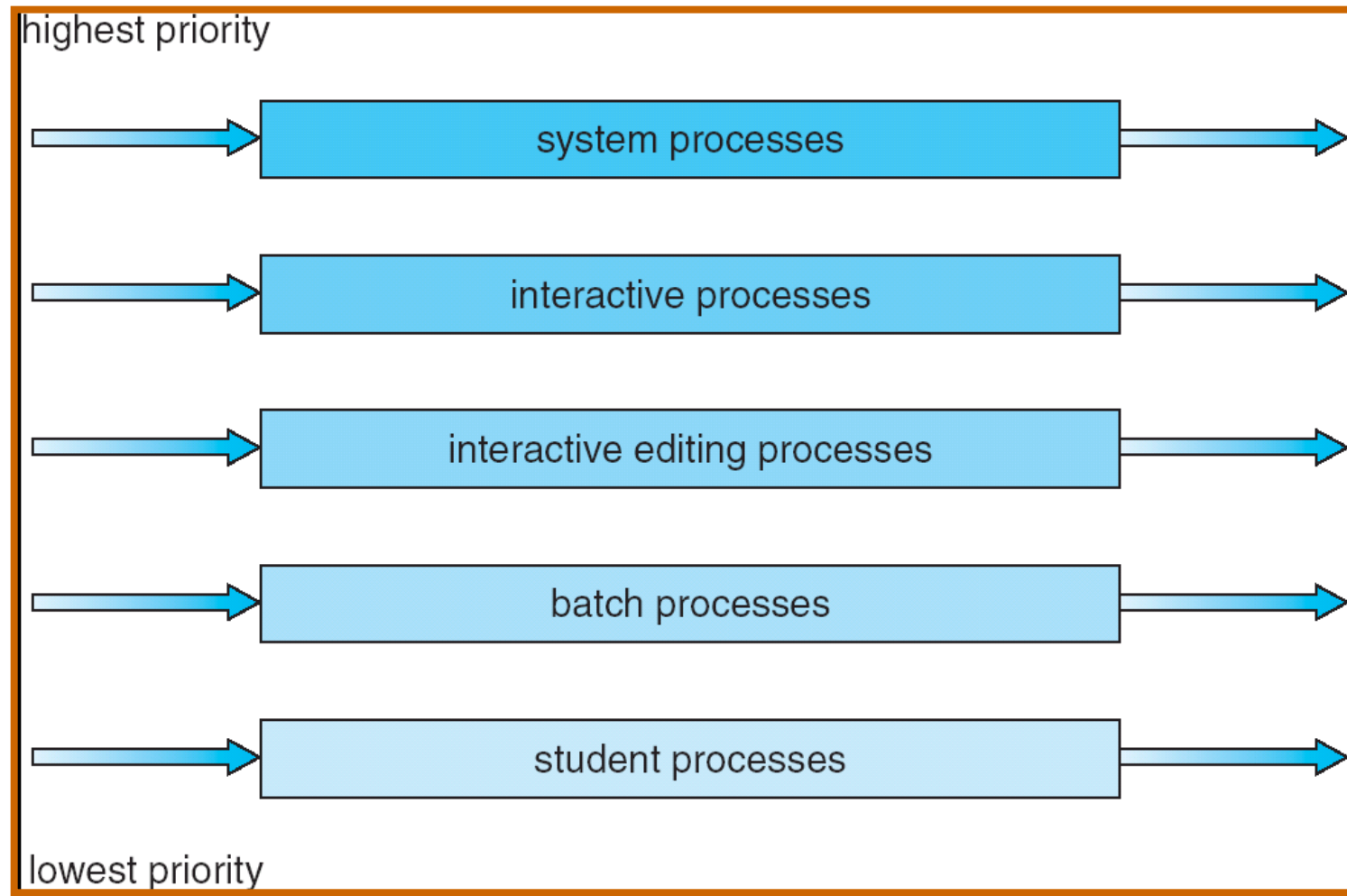
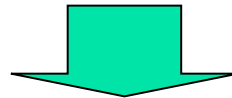


Figura 3: Fila de multiníveis no escalonamento dos processos

Filas Múltiplas com Feedback

- **Filas múltiplas com realimentação**
- Permite que um processo se mova entre as filas;
- Separar de acordo com burst de CPU;
 - Se utilizar muito tempo da CPU é movido para uma fila de menor prioridade;
- Processos de menor prioridade são executados por um quantum;
 - Se necessário dois quantum na próxima classe e quatro quantum e assim por diante.



Este algoritmo diminui o número de comutações da CPU entre os processos ativos

Filas Múltiplas com Feedback

- Um processo entra na fila F_0 , a qual emprega o RR.
 - Quando entra em execução recebe 8 milisegundos.
- Se não finalizar em 8 milisegundos e movido para a fila F_1 .
 - Em F_1 recebe 16 milisegundos para RR;
- Se ainda não concluir e movida para próxima fila.

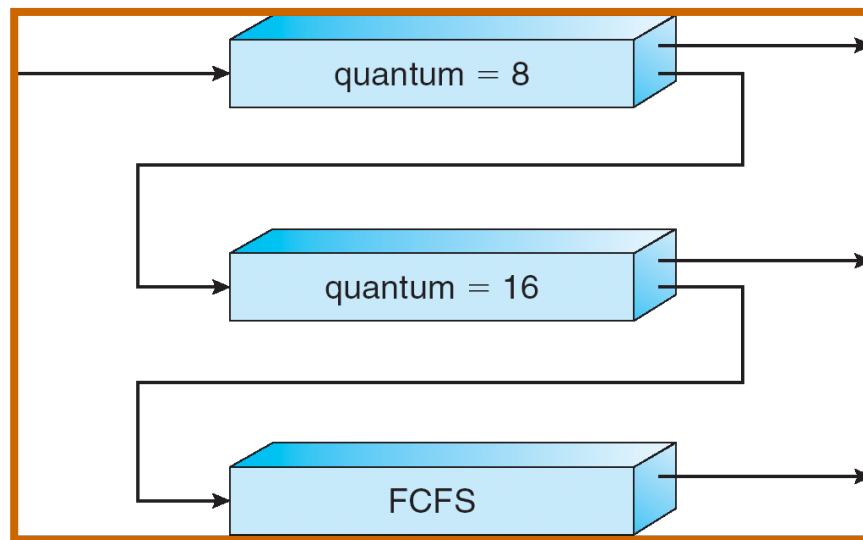


Figura 4: Fila de multiníveis com retroalimentação

Outros algoritmos interativos

- Algoritmo garantido (Guaranteed):
 - Garantias são dadas aos processos dos usuários:
 - n usuários $\rightarrow 1/n$ do tempo de CPU para cada usuário;
- Algoritmo por loteria (Lottery):
 - Cada processo recebe *tickets* que lhe dão direito de execução;
- Algoritmo por fração justa (Fair-Share):
 - Se um usuário A possui mais processos que um usuário B, o usuário A terá prioridade no uso da CPU;

Roteiro

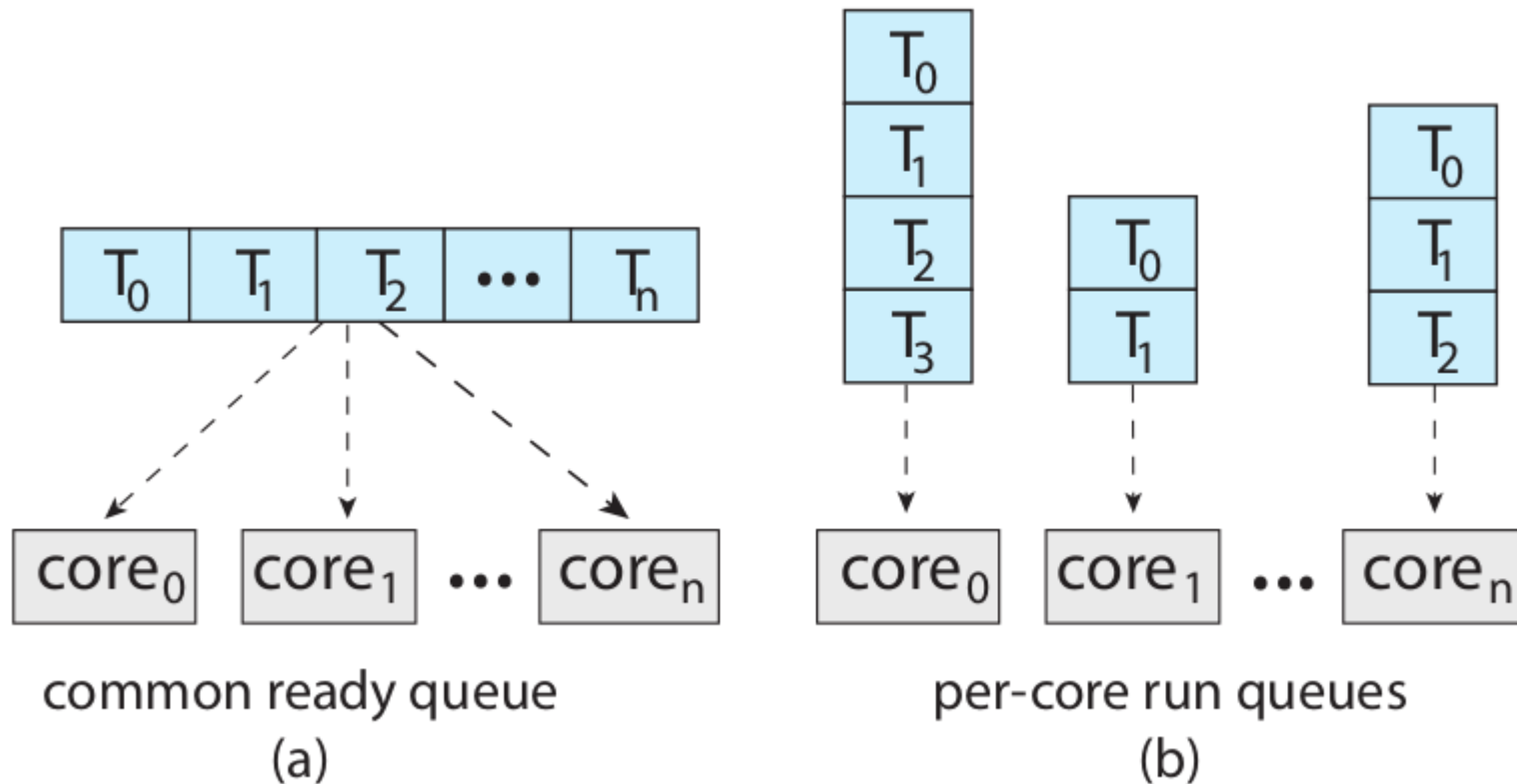
- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;

Escalonamento em múltiplos processadores

- Técnicas de multiprocessamento:
 - **Assimétrica** (Acesso não uniforme à memória (NUMA)): Escalonamento, processamento de E/S são tratadas pelo servidor mestre.
 - Os demais equipamentos executam somente código do usuário;
 - **Simétrico (SMP)**: cada processador é auto-escalonado. Os processos podem estar numa fila de pronto ou cada processador “CPU” tem sua fila de pronto.
 - Exemplos: Windows XP, Linux, Mac OS X, etc.

Escalonamento em múltiplos processadores

- Técnicas de multiprocessamento:



Escalonamento em múltiplos processadores

Balanceamento de Carga:

- O balanceamento busca manter a carga de trabalho distribuída uniformemente entre todas as CPUs;
- Necessário em sistemas em que cada CPU tem sua própria fila privada de processos elegíveis para execução;
- Duas técnicas são aplicadas:
 - **Migração Push:** uma tarefa específica verifica, periodicamente, a carga de cada CPU e eventualmente distribui-a movendo (pushing) a tarefa;
 - **Migração Pull:** a CPU ociosa puxa um processo que está esperando um processador ocupado.
- Alguns sistemas implementam ambas - Exemplo: Linux.

Escalonamento em múltiplos processadores

Afinidade:

- Alto custo de invalidação e preenchimento dos caches pela troca de processos em uma CPU;
- Maioria das arquiteturas SMP tenta evitar a migração de processo de uma CPU para outra: **afinidade de processador**.
- Podemos ter:
 - **Afinidade flexível:** tenta manter o processo em execução no mesmo processador, mas não há garantia.
 - **Afinidade rígido:** especifica o processador e não deve migrar para outras CPUs.

Roteiro

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;

Escalonamento em sistema de tempo real

- Tempo é um fator crítico;
- Produzir saídas corretas em determinado momento;
- Exemplos de sistemas críticos:
 - Aviões;
 - Hospitais;
 - Bancos;
 - Multimídia;
- Importante: obter respostas em atraso é tão ruim quanto não obter respostas;
- Os eventos são categorizados como periódicos e aperiódicos

Escalonamento em sistema de tempo real

- Tipos de STR:
 - *Hard Real Time*: atrasos não são tolerados;
 - Aviões, usinas nucleares, hospitais;
 - Ex. responder as altas temperaturas no núcleo de uma usina nuclear;
 - *Soft Real Time*: atrasos são tolerados;
 - Bancos; Multimídia;
 - Ex. coletar dados de controle de tráfego aéreo a cada segundo;

Escalonamento em sistema de tempo real

- Um sistema pode responder a múltiplos eventos periódicos. Dependendo do tempo de cada evento requisitado para processamento;
- Exemplo: Se há **m** eventos e o evento **i** ocorre com o período **P_i** e requer **C_i** segundos de CPU, então, a execução pode ser considerada apenas se:

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

- O sistema real que satisfaça esse critério é chamado escalonável.

Escalonamento em sistema de tempo real

- Exemplo:
- Eventos periódicos de 100, 200 e 500 ms
- Eventos requerem 50, 30, 100 ms de CPU por evento

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

- Esse sistema pode escalar esses processos?

Roteiro

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;

Windows XP - Escalonador

- Escalonador preemptivo baseado em prioridade;
- Os jobs de prioridade mais alta sempre será executada primeira;
- Prioridade pode mudar – decrementa após o tempo quantum de execução;
- Há 32 níveis de prioridade, sendo separada em filas
 - **Classe de tempo real (fixo):** níveis 16 a 31
 - **Outras classes (variável):** níveis 1 a 15.

Linux - Escalonador

- Baseado em prioridade com dois intervalos (tempo real e não tempo real):
 - Tempo Real: 0 a 99;
 - Nice (ajuste de prioridade): usa “-20” a “19” – completo fatiamento com prioridade a cada nível;
 - Tarefa de mais alta prioridade tem maior quantum de CPU diferente do Windows;

numeric priority	relative priority		time quantum
0 • • • 99	highest	real-time tasks	200 ms
100 • • • 140	lowest	other tasks	10 ms

Sumário

- Conceito
- Despachante
- Algoritmos de escalonamento
 - Exercícios
- Escalonamento em múltiplos processadores;
- Escalonamento em sistema de tempo real;
- Exemplos em sistemas operacionais;
- Leituras sugeridas.

Leituras Sugeridas

- Silberschatz, A., Galvin, P. B. Gagne, G. **Sistemas Operacionais com Java**. 7º , edição. Editora, Campus, 2008.
 - Capítulo 5
- Andrew S. Tanenbaum. **Sistemas Operacionais. Modernos**. 2ª Ed. Editora Pearson, 2003.
 - Capítulo 2

