Universidade Federal da Bahia

Sistemas Operacionais

MATA58

Prof. Maycon Leone M. Peixoto

mayconleone@dcc.ufba.br

Conceitos Básicos
Chamadas de Sistema

- **Modos de Acesso**:
  - Modo usuário;
  - Modo *kernel* ou Supervisor ou Núcleo;
  - São determinados por um conjunto de bits localizados no registrador de status do processador: PSW (*program status word*);
    - Por meio desse registrador, o hardware verifica se a instrução pode ou não ser executada pela aplicação;
  - Protege o próprio *kernel* do Sistema Operacional na RAM contra acessos indevidos;

Conceitos Básicos
Chamadas de Sistema

- Modo usuário:
  - Aplicações não têm acesso direto aos recursos da máquina, ou seja, ao hardware;
  - Quando o processador trabalha no modo usuário, a aplicação só pode executar **instruções sem privilégios**, **com um acesso reduzido de instruções**;
  - Por que? Para garantir a **segurança** e a **integridade do sistema**;

Conceitos Básicos
Chamadas de Sistema

- Modo *Kernel:*
  - Aplicações têm acesso direto aos recursos da máquina, ou seja, ao hardware;
  - **Operações com privilégios**;
  - Quando o processador trabalha no modo *kernel*, a aplicação tem **acesso ao conjunto total de instruções**;
  - Apenas o SO tem acesso às instruções privilegiadas;

Conceitos Básicos
Chamadas de Sistema

- Se uma aplicação precisa realizar alguma instrução privilegiada, ela realiza uma **chamada de sistema *(system call*)**, que altera do modo usuário para o modo *kernel*;
- Chamadas de sistemas são a **porta de entrada** para o modo *Kernel*;
  - São a interface entre os programas do usuário no modo usuário e o Sistema Operacional no modo kernel;
  - As chamadas diferem de SO para SO, no entanto, os conceitos relacionados às chamadas são similares independentemente do SO;

Conceitos Básicos
Chamadas de Sistema

- Exemplos de chamadas de sistema:
  - Chamadas para gerenciamento de processos:
    - `Fork (CreateProcess – WIN32)` – cria um processo;
  - Chamadas para gerenciamento de diretórios:
    - `Mount` – monta um diretório;
  - Chamadas para gerenciamento de arquivos:
    - `Close (CloseHandle – WIN32)` – fechar um arquivo;
  - Outros tipos de chamadas:
    - `Chmod`: modifica permissões;

Processos

- Introdução
- Escalonamento de Processos
- Comunicação entre Processos
- Threads
- Deadlock

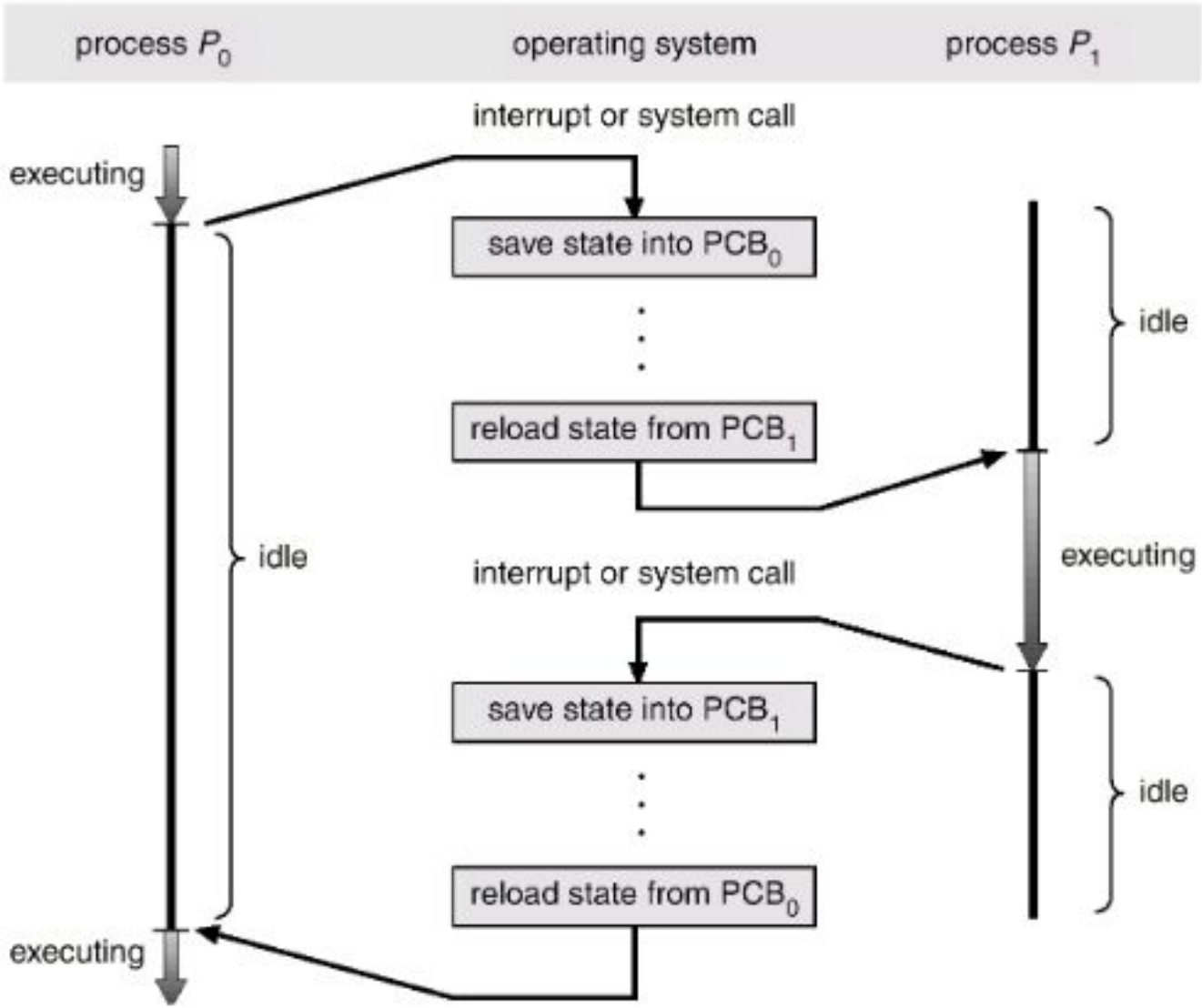
Escalonamento de Processos

- Escalonador de Processos escolhe o processo que será executado pela CPU;
- Escalonamento é realizado com o auxílio do hardware;
- Escalonador deve se preocupar com a eficiência da CPU, pois o chaveamento de processos é complexo e custoso:
  - Afeta desempenho do sistema e satisfação do usuário;
- Escalonador de processo é um processo que deve ser executado quando da **mudança de contexto** (troca de processo);

Escalonamento de Processos

- Mudança de Contexto:
  - Overhead de tempo;
  - Tarefa cara:
    - Salvar as informações do processo que está deixando a CPU em seu BCP conteúdo dos registradores;
    - Carregar as informações do processo que será colocado na CPU copiar do BCP o conteúdo dos registradores;

Troca de Contexto entre processos



Escalonamento de Processos

- Situações nas quais escalonamento é necessário:
  - Um novo processo é criado;
  - Um processo terminou sua execução e um processo pronto deve ser executado;
  - Quando um processo é bloqueado (semáforo, dependência de E/S), outro deve ser executado;
  - Quando uma interrupção de E/S ocorre o escalonador deve decidir por: executar o processo que estava esperando esse evento; continuar executando o processo que já estava sendo executado ou executar um terceiro processo que esteja pronto para ser executado;

Escalonamento de Processos

- Hardware de relógio fornece interrupções de relógio e a decisão do escalonamento pode ser tomada a cada interrupção ou a cada *k* interrupções;
- Algoritmos de escalonamento podem ser divididos em duas categorias dependendo de como essas interrupções são tratadas:
    - Preemptivo: escolhe um processo e o deixa executando por um tempo máximo;
    - Não-preemptivo: estratégia de permitir que o processo que está sendo executado continue sendo executado até ser bloqueado por alguma razão (semáforos, operações de E/S-interrupção) ou que libere a CPU voluntariamente;

Escalonamento de Processos

- Categorias de Ambientes:
  - **Sistemas em Batch**: usuários não esperam por respostas rápidas; algoritmos não-preemptivos ou preemptivos com longo intervalo de tempo;
  - **Sistemas Interativos**: interação constante do usuário; algoritmos preemptivos; Processo interativo espera comando e executa comando;
  - **Sistemas em Tempo Real**: processos são executados mais rapidamente; tempo é crucial sistemas críticos;

Escalonamento de Processos

- Características de algoritmos de escalonamento:
  - Qualquer sistema:
    - **Justiça** (*Fairness*)*:* cada processo deve receber uma parcela justa de tempo da CPU;
    - **Balanceamento**: diminuir a ociosidade do sistema;
    - **Políticas do sistema** – prioridade de processos;

Escalonamento de Processos

- Características de algoritmos de escalonamento:
  - Sistemas em *Batch*:
    - **Vazão** (*throughput*): maximizar o número de *jobs* executados por hora;
    - **Tempo de retorno** (*turnaround time*): tempo no qual o processo espera para ser finalizado;
    - **Eficiência**: CPU deve estar 100% do tempo ocupada;
  - Sistemas Interativos:
    - **Tempo de resposta**: tempo esperando para iniciar execução;
    - **Proporcionalidade**: satisfação do usuários;

Escalonamento de Processos

- Características de algoritmos de escalonamento:
  - Sistemas em Tempo Real:
    - **Cumprimento dos prazos**: prevenir perda de dados;
    - **Previsibilidade**: prevenir perda da qualidade dos serviços oferecidos;

Processos

- Introdução
- Escalonamento de Processos
- Comunicação entre Processos
- Threads
- Deadlock

Algoritmos para Escalonamento de Processos

Exercícios

1. Na sua concepção, qual a importância dos Sistemas Operacionais?
2. Como seria utilizar um computador sem um sistema operacional? Quais são suas duas principais funções?
3. Explique o conceito de máquina virtual. Qual a grande vantagem em utilizar este conceito?
4. O que é um processo? O que é um programa?
5. Quais são os estados que um processo pode assumir?
6. Faça o diagrama dos estados de um processo e explique os eventos que ocorrem entre eles (transições). Na teoria, com três estados poderia haver seis transições, duas para cada estado. Contudo, somente quatro transições são mostradas. Há alguma circunstância na qual uma delas ou ambas as transições não ilustradas possam ocorrer?
7. Quais os tipos (modelos estruturais) de sistemas operacionais que existem?
8. O que significa um processo sofrer preempção?

Algoritmos para Escalonamento de Processos

Processos

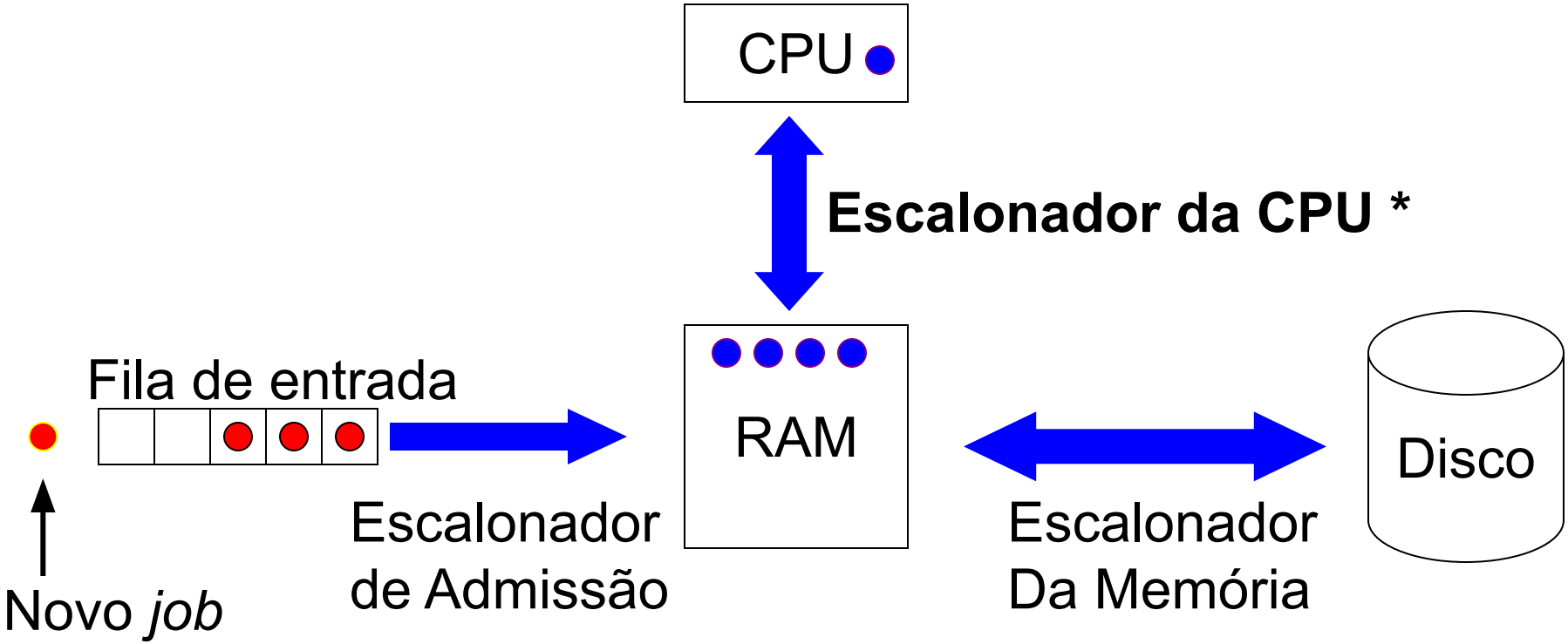
- **Escalonamento de Processos**
  - **Algoritmos de Escalonamento**
    - **Batch**
    - **Interativo**
    - **Tempo Real**

Escalonamento de Processos
Sistemas em *Batch*

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

Escalonamento de Processos
Sistemas em *Batch*

- Escalonamento *Três Níveis*



Escalonamento de Processos
Sistemas em *Batch*

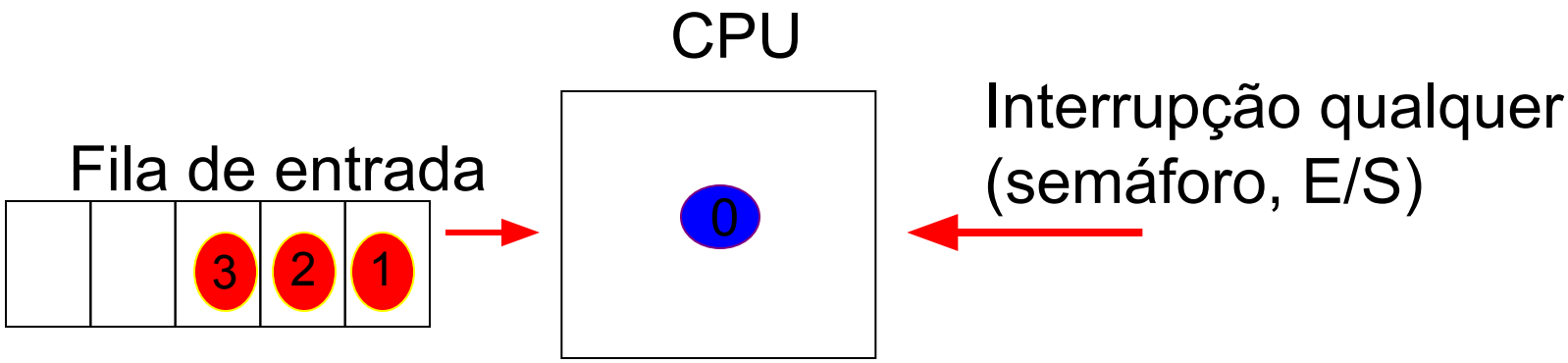
- Escalonamento *Three-Level*
  - **Escalonador de admissão**: decide qual job será admitido no sistema. Por exemplo, uma mescla de jobs orientados a CPU e orientados à E/S; processos com menor tempo de acesso à CPU e maior tempo de interação com dispositivos de E/S;
  - **Escalonador da Memória**: decisões 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;

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;

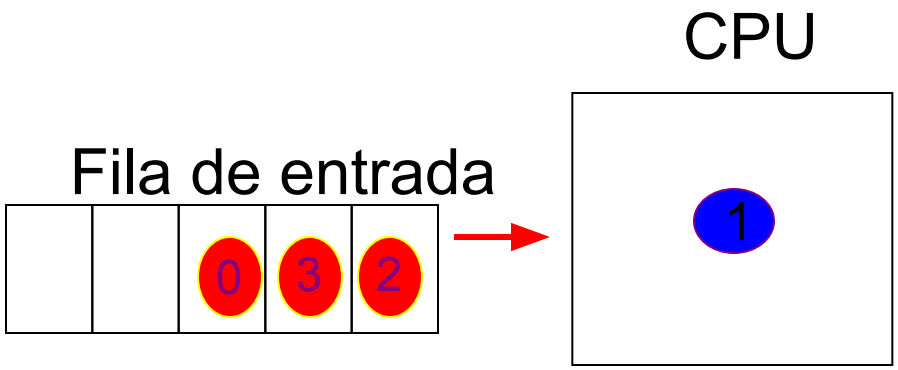
Escalonamento de Processos
Sistemas em *Batch*

- 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*;
  - 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

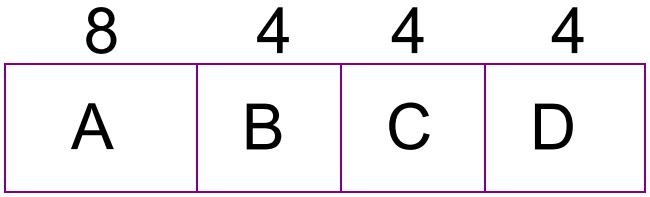
_____

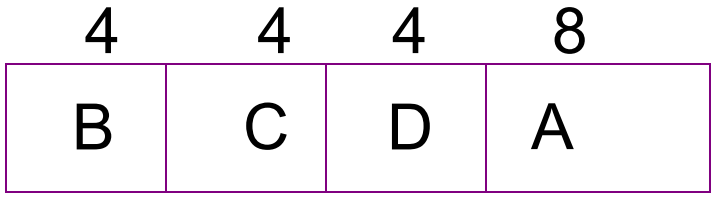
Tempo médio-*turnaround* (4a+3b+2c+d)/4

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 □ 56/4 = 14

Menor *job* primeiro:
Turnaround B = 4
Turnaround C = 8
Turnaround D = 12
Turnaround A = 20
Média □ 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!

Escalonamento de Processos
Sistemas Interativos

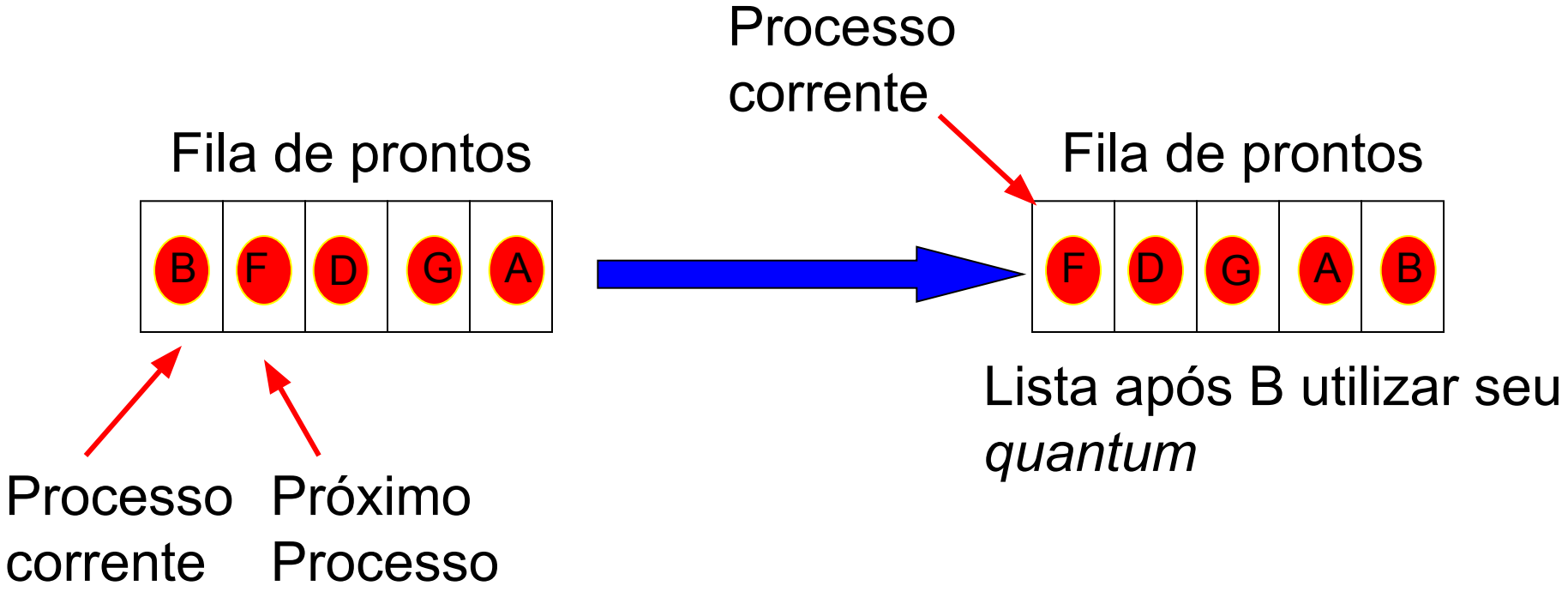
- Algoritmos para Sistemas Interativos:
  - *Round-Robin*;
  - Prioridade;
  - Múltiplas Filas;
  - *Shortest Process Next*;
  - Garantido;
  - *Lottery*;
  - *Fair-Share*
- Utilizam escalonamento em dois níveis (escalonador da CPU e memória);

Escalonamento de Processos
Sistemas Interativos

- Algoritmo *Round-Robin*
  - Antigo, mais simples e mais utilizado;
  - Preemptivo;
  - 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*



Escalonamento de Processos
Sistemas Interativos

- Algoritmo *Round-Robin*
  - Tempo de chaveamento de processos;
  - *quantum*: se for muito pequeno, ocorrem 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:

quantum

Δt = 4 mseg

x = 1mseg 25% de tempo de CPU é perdido menor eficiência

Δt = 100 mseg

x = 1mseg 1% de tempo de CPU é perdido Tempo de espera dos processos é maior

chaveamento

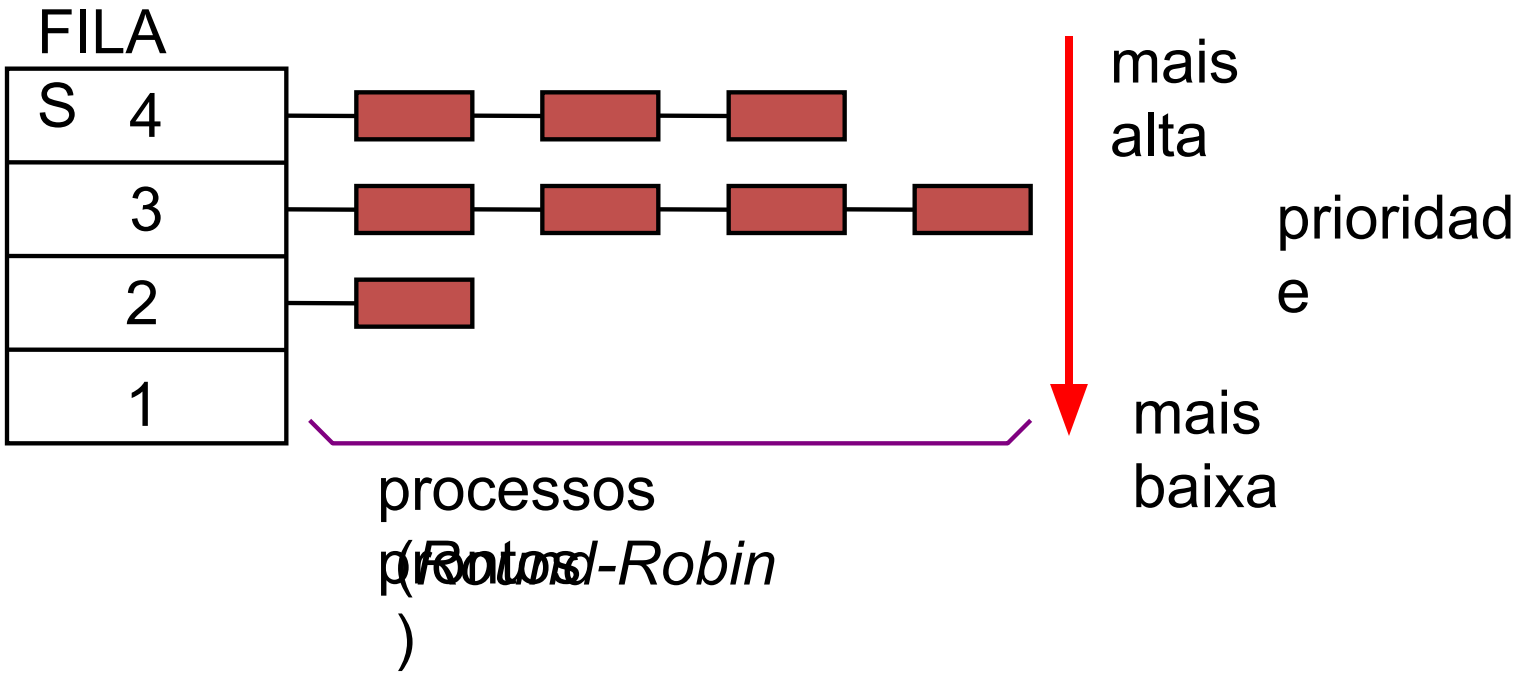
quantum razoável: 20-50 mseg

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

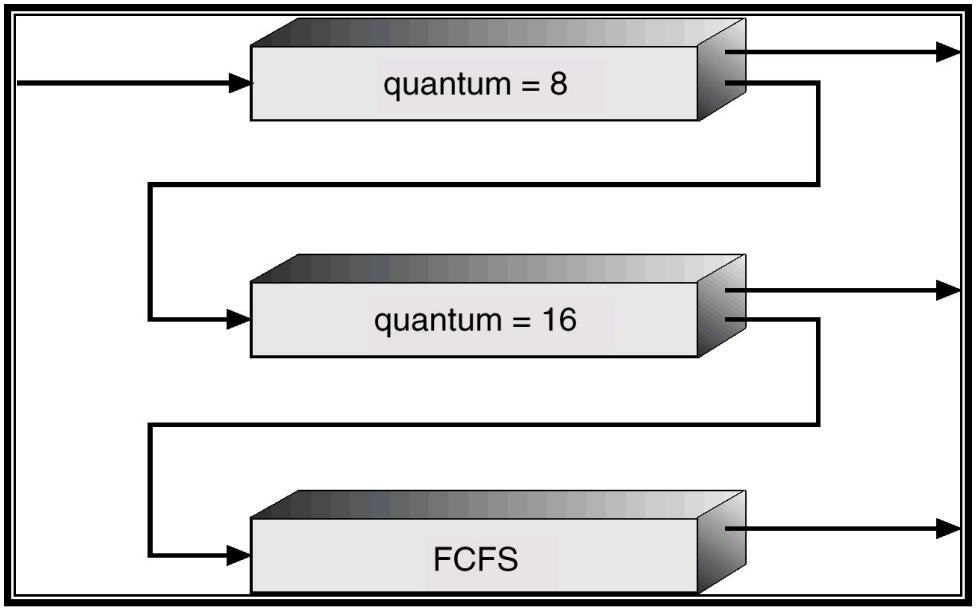
- Algoritmo com Prioridades
  - Como evitar que os processos com maior prioridade sejam executado indefinidamente?
    - Diminuir a prioridade do processo corrente a cada interrupção do relógio e trocá-lo pelo próximo processo assim que sua prioridade caia abaixo da prioridade do próximo processo com prioridade mais alta (chaveamento);
    - Atribuir um quantum máximo no qual o processo pode executar;

Escalonamento de Processos
Sistemas Interativos

- Múltiplas Filas:
  - CTSS (*Compatible Time Sharing System*);
  - Classes de prioridades;
  - Preemptivo;
  - Cada classe de prioridades possui *quanta* diferentes;

Escalonamento de Processos
Sistemas Interativos

- Múltiplas Filas:
  - Assim, a cada vez que um processo é executado e suspenso ele recebe mais tempo para execução mas passa para uma fila com menor prioridade de execução



Escalonamento de Processos
Sistemas Interativos

- **Múltiplas Filas:**
  - Ex.: um processo precisa de 100 *quanta* para ser executado;
    - Inicialmente, ele recebe um *quantum* para execução;
    - Das próximas vezes ele recebe, respectivamente, 2, 4, 8, 16, 32 e 64 *quanta* (7 chaveamentos) para execução;

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;
  - Solução: realizar uma estimativa com base no comportamento passado e executar o processo cujo tempo de execução estimado seja o menor;

Escalonamento de Processos
Sistemas Interativo

- **Algoritmo Garantido:**
  - Garantias são dadas aos processos dos usuários
  - Exemplo: **n** processos **1/n** do tempo de CPU para cada processo;
    - Deve ser mantida taxa de utilização de cada processo
    - Tem prioridade o que estiver mais distante do prometido
  - Difícil de implementar

Escalonamento de Processos Sistemas Interativo

- **Algoritmo por Loteria:**
  - Cada processo recebe "*tickets*" que lhe dão direito de execução;
  - A cada troca de processo um "*tickets*" é sorteado
  - O dono do "*tickets*" *sorteado recebe* o direito de ocupar a CPU
  - Possível definir prioridade entre os processos por meio do número de "*tickets*" atribuído a cada processo
  - Fácil de implementar e de adaptar

Escalonamento de Processos
Sistemas Interativo

- **Algoritmo por Fração Justa (Fair-Share):**
  - O escalonamento é feito considerando o dono dos processos
  - Cada usuário recebe uma fração da CPU e processos são escalonados visando garantir essa fração
  - Se um usuário A possui mais processos que um usuário B e os dois têm a mesma prioridade, os processos de A demorarão mais que os do B

Escalonamento de Processos
Sistemas em Tempo Real

- Tempo é um fator crítico;
- 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;

Earliest Deadline First - EDF

- Corresponde a uma atribuição dinâmica de prioridades que define a ordenação das tarefas segundo os seus **deadlines absolutos**.
  - A tarefa com maior prioridade é a que tem o deadline mais próximo do tempo atual.
- A cada chegada de tarefa a fila de prontos é reordenada, considerando a nova distribuição de prioridades.

Minimum-laxity-first (MLF)

- Associa um *laxity* à cada processo do sistema, e então seleciona o processo com o menor *laxity* para ser executado. *Laxity* é definido como:

  Laxity = deadline – tempo atual – tempo de CPU preciso

- *Laxity* é a medida da flexibilidade disponível para escalonar um processo. Um laxity de zero significa que o processo precisa iniciar a execução ou correrá o risco de alhar no cumprimento o deadline.
  - A principal diferença entre MLP e EDF é que o MLF leva em consideração o tempo de execução de um processo, e que o EDF não considera.

Escalonamento de Processos

- 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;