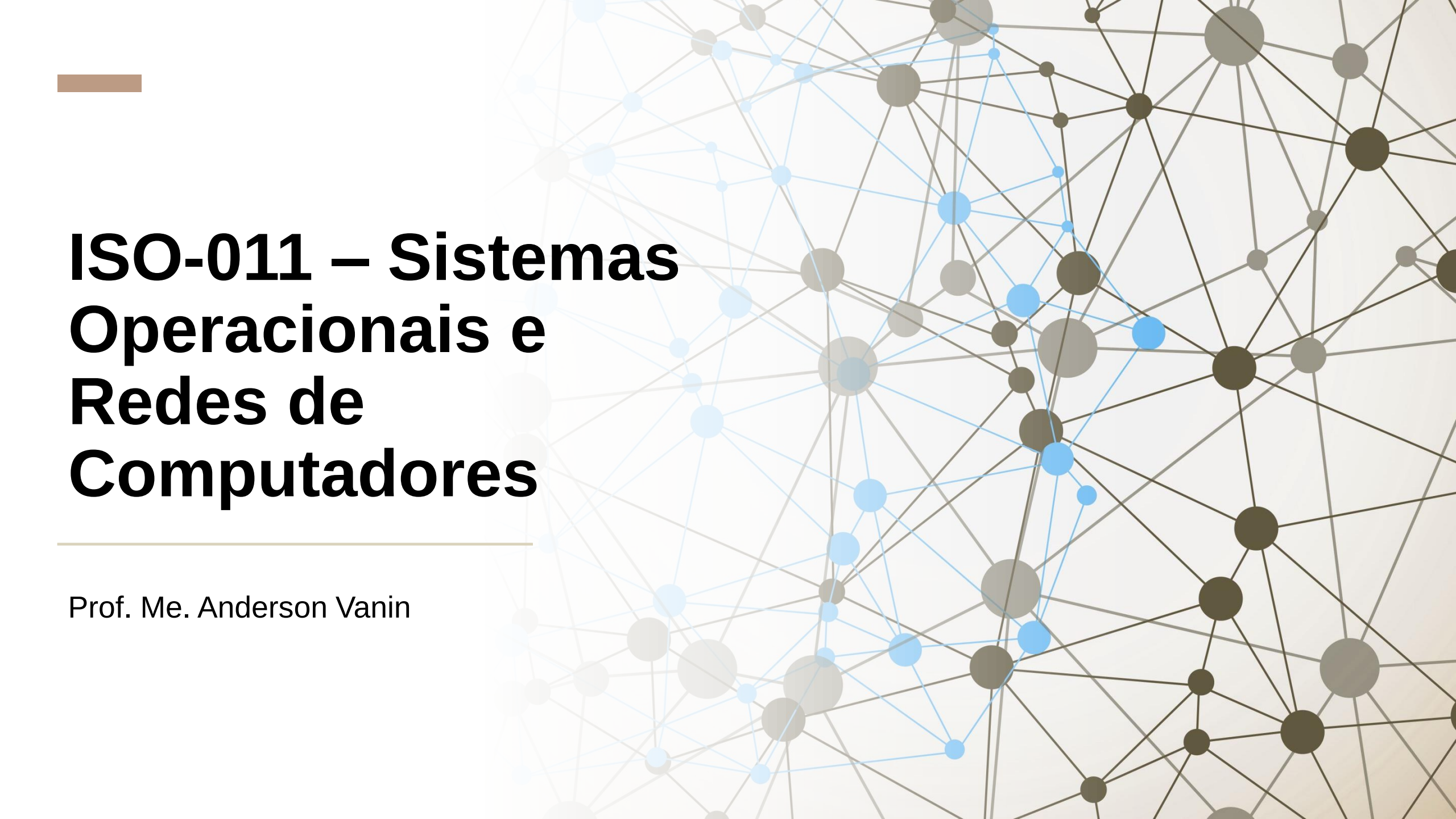
ISO-011 – Sistemas Operacionais e Redes de Computadores

Prof. Me. Anderson Vanin



Processos

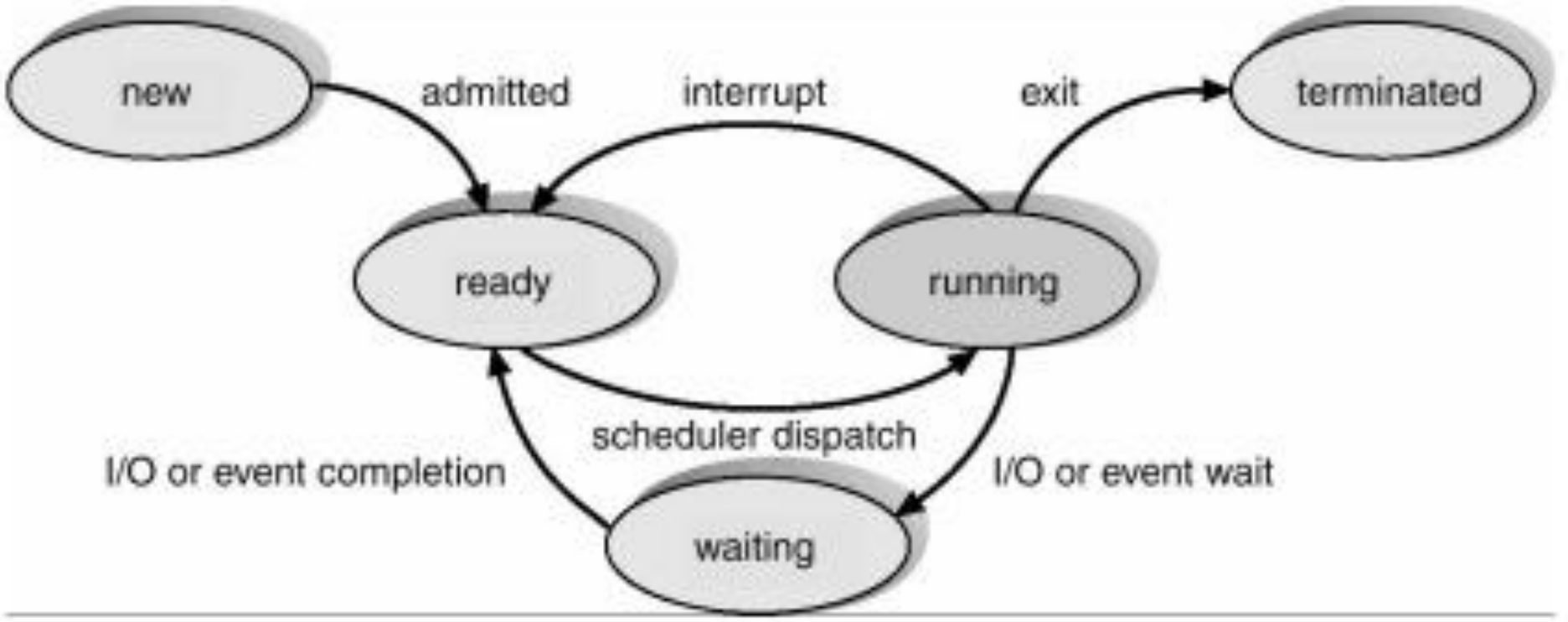
- Um Sistema Operacional executa uma variedade de programas:
  - Sistemas de processamento em lotes (batch) − processa jobs
  - Sistemas de tempo partilhado (time−shared) − roda processos de usuários ou tarefas (tasks)
- **Processo: é um programa em execução**
- Um processo inclui:
  - Contador de programa (PC)
  - Pilha
  - Segmento (área) de dados

Estados de Processos

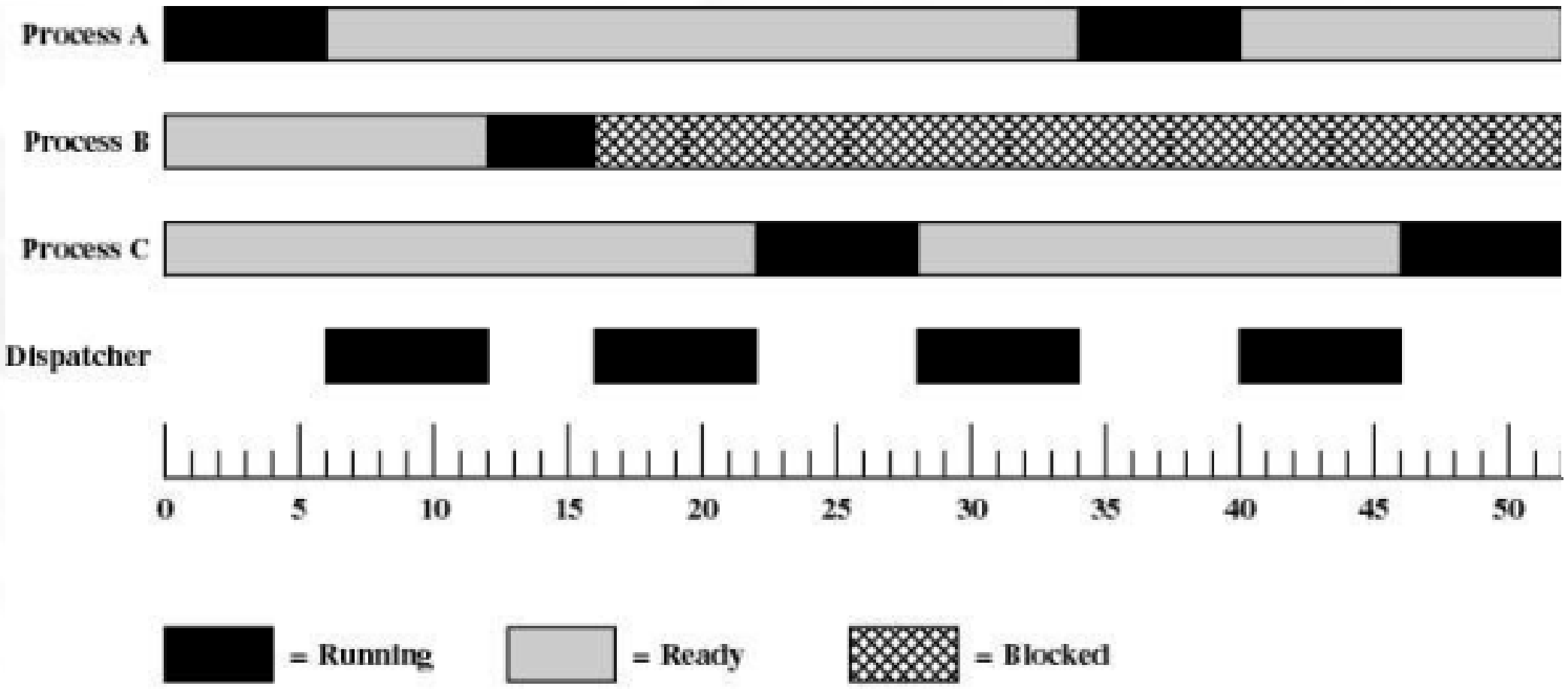
- Ao executar, o processo muda de “estados”. Durante a sua execução, um processo passa por diversos estados, refletindo o seu comportamento dinâmico, isso é, a sua evolução no tempo.
  - **New**: processo está sendo criado
  - **Ready**: pronto para execução.
  - **Running**: instruções do processo estão executando
  - **Waiting**: processo está esperando ocorrência de algum evento
  - **Terminated**: processo terminou a execução

Apenas um único processo pode estar no estado “running” num dado processador, num dado instante.

Estados de Processos



Modelo de 5 Estados



Transições de Estados de um Processo

- **Null → New**:
  - Um novo processo é criado para executar o programa.
    - Novo batch *job*.
    - Logon interativo (usuário se conecta ao sistema).
    - S.O. cria processo para prover um serviço (ex: impressão).
    - Processo cria um outro processo (“*process spawning*”).

Swapping: procedimento que consiste em mover todo ou parte de um processo da memória para o disco.

Transições de Estados de um Processo

- **New → Ready**:
  - No estado New, recursos foram alocados pelo S.O. mas não existe um compromisso de que o processo será executado.
    - Número de processos já existentes;
    - Quantidade de memória virtual requerida, etc.
  - Manter um bom desempenho do sistema é o fator limitante da criação de novos processos.

Transições de Estados de um Processo

- **Ready → Running**:
  - Definido pela política de escalonamento de processos adotada pelo S.O.
- **Running → Exit**:
  - Processo terminou as suas atividades ou foi abortado.
  - Término normal;
  - Término do processo pai (em alguns sistemas)
  - Excedeu o limite de tempo;
  - Memória não disponível;
  - Erro aritmético ou de proteção;
  - Execução de instrução inválida ou de instrução privilegiada no modo usuário;
  - Intervenção do S.O.(ex: ocorrência de ***deadlock***);

Transições de Estados de um Processo

- **Running → Ready** :
  - Tempo máximo de execução sem interrupção foi atingida;
  - Processo é “***preemptado***” pelo S.O.
- **Running → Blocked**:
  - Processo requisitou alguma coisa pela qual deve esperar
- **Blocked → Ready**:
  - Evento pelo qual o processo espera aconteceu.
- **Ready → Exit**:
  - Processo pai termina um processo filho.
  - Processo pai é terminado, e os processos filhos associados são também finalizados.
- **Blocked → Exit**:
  - Idem anterior.

Troca de Contexto de um Processo

- Contexto de um processo são todas as informações necessárias para que o S.O. possa restaurar a execução do processo a partir do ponto interrompido.
- A troca de contexto ocorre sempre que um novo processo é selecionado para execução (isso é, quando a UCP é chaveada para um outro processo).
- O tempo de troca de contexto é puro overhead e é dependente de suporte de hardware (ex: salvamento automático do PC).

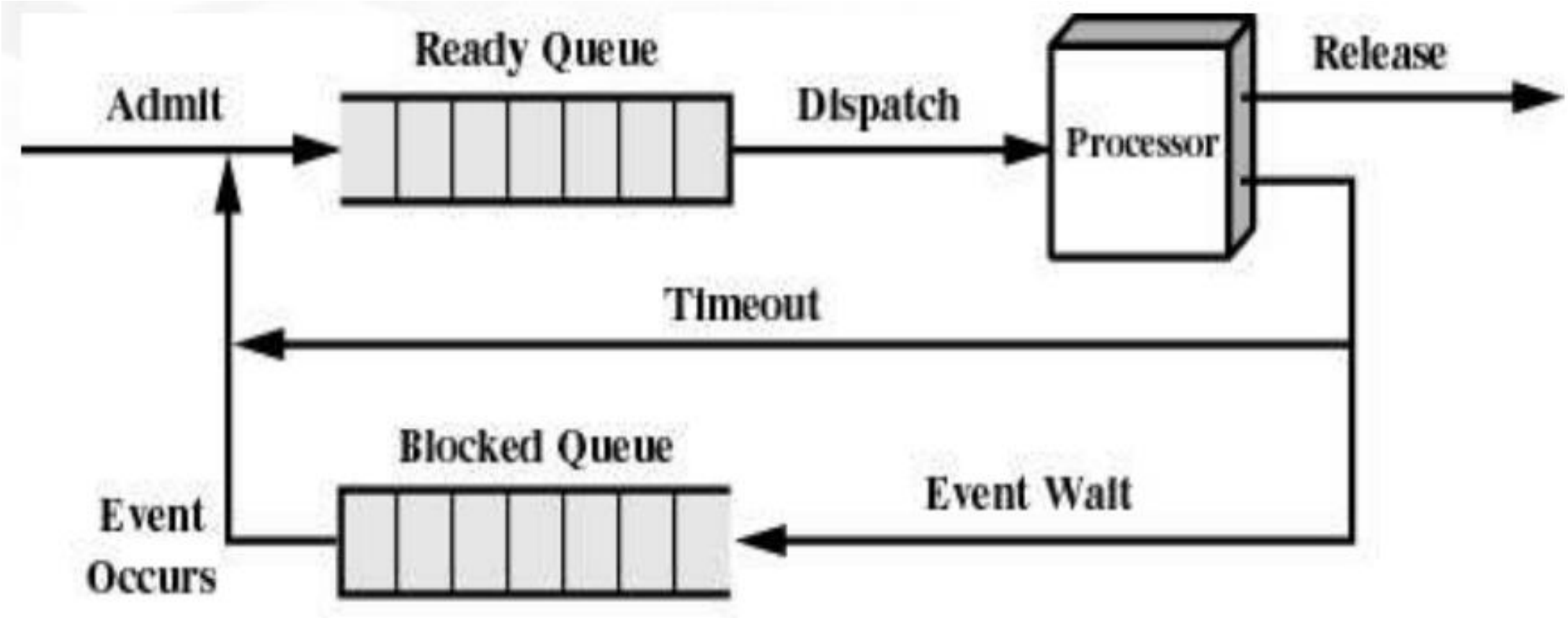
Quando Chavear um Processo

- **Interrupção do relógio**
  - Fatia de tempo de posse da UCP expirou;
- **Interrupção de E/S**
- **Falta de memória**
  - Endereço de memória está na memória virtual (disco); logo deve ser trazido para a memória principal.
- **Trap**
  - Ocorrência de erro.
  - Pode fazer com que o processo seja movido para o estado Exit.

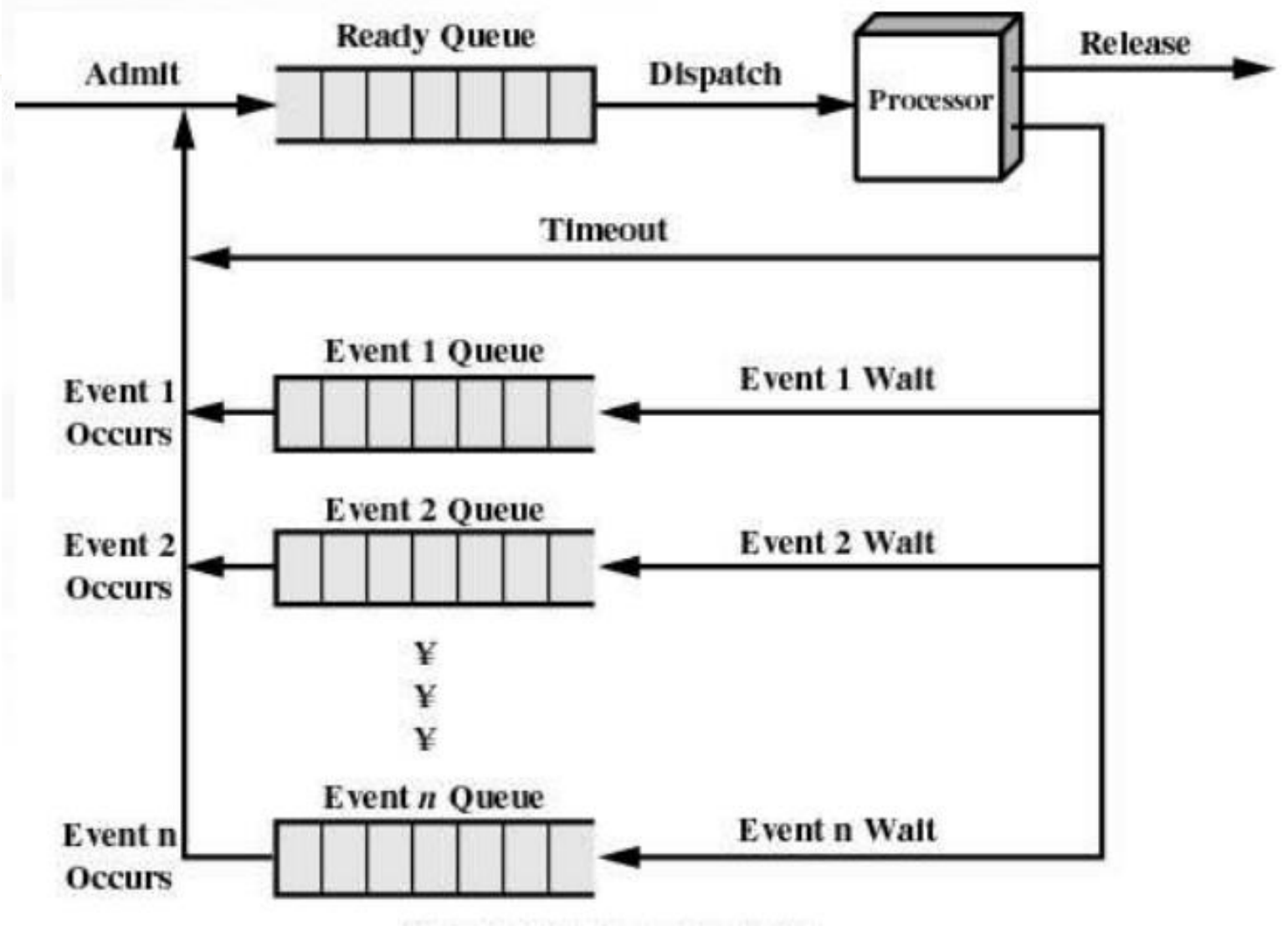
Filas para escalonamento de processos

- **Um processo sempre faz parte de alguma fila**.
- Eventos realizam a transição de uma fila para outra.
- Fila de prontos e uma ou mais filas de bloqueados.

Filas para escalonamento de processos



Filas para escalonamento de processos



Processos Suspensos

- Processador é tão mais rápido que os dispositivos de E/S que todos os processos em memória poderiam ficar em situação de espera.
  - Mesmo com multiprogramação, o processador poderia ficar a maior parte do tempo ocioso!
- Aumento de memória para acomodar mais processos:
  - **Aumento do custo**;
  - **Disponibilidade de mais memória geralmente resulta em processos maiores e não em maior número de processos**.
- Swapping: procedimento que consiste em mover todo ou parte de um processo da memória para o disco.

Processos Suspensos

- Quando nenhum dos processos na memória principal está no estado **Ready** o sistema operacional manda um dos processos bloqueados para o disco, e o coloca numa **fila de processos suspensos**.

Processos Suspensos

- O S.O traz então do disco algum outro processo da fila de suspensos ou atende a uma solicitação de criação de um novo processo.
- O swap é uma operação de E/S e, como tal, existe a possibilidade de tornar o problema ainda pior, caso o sistema de E/S não seja eficiente.
- Modelo mais elaborado: dois novos estados são então
  - **Blocked, suspend**: o processo está em memória secundária e aguardando por um evento.
  - **Ready, suspend**: o processo está em memória secundária mas está disponível para execução, tão logo ele seja carregado na memória principal.
  - OBS: [**Blocked**: processo está na MP e aguardando por um evento]

Classificação de Processos

Processos podem ser classificados em:

- **I/O−bound processes** − delimitados pelo tempo de I/O, gasta mais tempo fazendo I/O do que computações, muitas pequenas rajadas (bursts) de CPU
- **CPU−bound processes** − delimitados pelo tempo de CPU, gastam a maior parte do tempo fazendo computações; Poucas RAJADAS LONGAS de CPU

Threads

- Uma thread é uma linha de execução de código que executa em paralelo com outras linhas do mesmo processo, compartilhando seu espaço de memória.
- Na prática uma thread é equivalente a um “mini-processo” dentro de um processo
- Isto permite que várias ações sejam executadas em paralelo por um mesmo processo

Threads

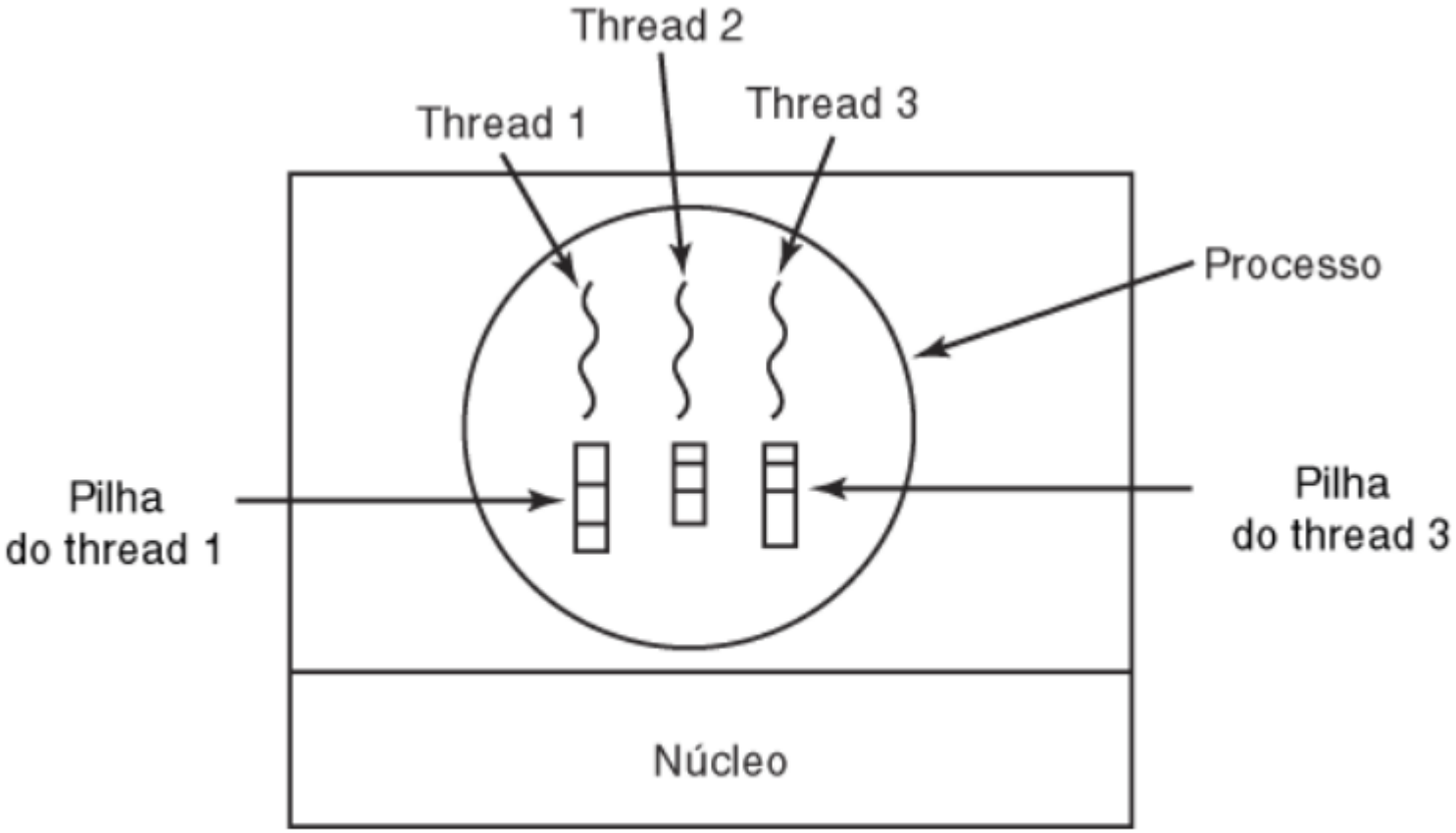
- Em um programa muitas vezes é necessário executar mais de uma atividade ao mesmo tempo
  - ex.: aguardar a entrada de dados do usuário e reproduzir um som enquanto aguarda
- Uma thread é muito mais leve que um processo comum.
- Ganho de performance na criação e destruição de threads se comparada a processos (10 a 100x)
- Quando uma aplicação tem atividade I/O bound e CPU bound as threads podem acelerar a execução, pois não concorrerão por recurso.
- O uso de Threads pode também garantir um uso máximo dos vários processadores existentes em uma CPU

Threads

Uma thread consiste de:

- **Apontador de instruções (PC)**
- **Conjunto de registradores**
- **Espaço de pilha**

Threads



Cada thread tem sua própria pilha

Threads - Vantagens

- São processos “leves”
- Troca de contexto mais rápida
- Tempo de criação menor
- Diminui o tempo de resposta do sistema;
- Maior facilidade para mesclar threads I/O-bound com threads CPU-bound.
- Usa eficientemente as arquiteturas multi-processadas/multicores.

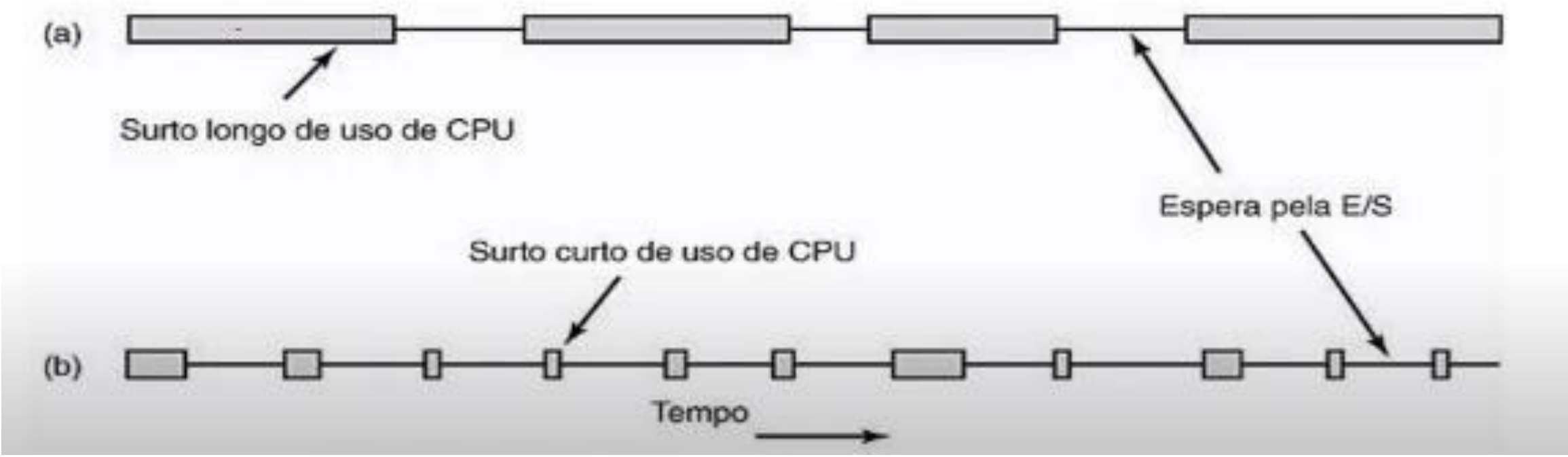
Threads - Principais aplicações

- Aplicações de processamento “paralelo”
  - **CPU bound**: ganho com várias CPU
  - **I/O bound**: ganho sempre
    - **Se uma thread está bloqueada a outra pode executar**
- Aplicações que acessam dispositivos secundários lentos (disco local ou remoto) e não querem ficar esperando – síncronos – pela resposta para poder continuar executando.
- Aplicações que controlam múltiplos servidores ou múltiplos clientes.
- Melhorar funcionalidade e performance da interface gráfica.

Escalonamento de Processos

- Quando um computador é multiprogramado, muitas vezes ele possui variados processos que competem pela CPU ao mesmo tempo.
- Essa situação ocorre sempre que dois ou mais processos estão simultaneamente no **estado de pronto**.
- A parte do SO que faz a escolha de qual processo deve ser executado é chamado de **escalonador**, e o algoritmo que é usado para escolher o processo a ser executado é chamado de **algoritmo de escalonamento**.

Escalonamento de Processos



Escalonamento de Processos

Quando Escalonar?

- Quando se cria um novo processo
- No término de um processo
- Quando um processo é bloqueado
- Quando um processo executa algum evento de Entrada e Saída (I/O)

Escalonamento de Processos

- Em sistemas multiprogramados (multi-tarefa), a cada instante um ou mais processos podem estar no estado pronto, e.g.:
  - Processos do sistema vs processos de usuários
  - Processos curtos vs longos/eternos
  - Processos interativos vs intensivos em CPU (CPUbound) ou jobs em batch
- O **Escalonador** é a parte do núcleo responsável por:
  - Gerenciar a(s) fila(s) de prontos, ajustando prioridades dos processos
  - Escolher qual dos processos prontos vai ser o próximo a usar CPU (de acordo com as prioridades dos processos)

Objetivos do Escalonamento

- **Garantir justiça:** dar a cada processo uma porção justa de CPU
- **Aplicação da política**: verificar se a política estabelecida é cumprida
- **Equilíbrio:** manter ocupada todas as partes do sistema (CPU, E/S, etc...)

Escalonamento

1. **Escalonamento de longo prazo**
   - Determina quando um processo é inserido no sistema para execução
   - Ao ser criado, processo recebe uma prioridade (vai para uma fila dos prontos)
2. **Escalonamento de médio prazo**
   - Determina quando processo “swapped out” é recarregado na memória RAM
3. **Escalonamento de curto prazo (“dispatching”)**
   - Escolhe um dos processos da fila de prontos para executar (usar o processador)

Tipos de Escalonamento

Diferenças com relação…:

- ao grau de interferência nos processos:
  - **Preemptivo**: O processo executa fatias de tempo (quantum) determinado pelo Sistema Operacional
  - **Não preemptivo**: O processo executa até o fim, sem ser interrompido

Tipos de Escalonamento

Diferenças com relação…:

- ao método de seleção do processo mais prioritário:
  - Uso da função Prioridade(processo)
  - Regra de desempate (para processos com mesma prioridade)
    - Escolha aleatória
    - Cronológica (FIFO)
    - Cíclica (Round Robin)
    - Outra informação sobre a tarefa (deadline)

Categorias de Algoritmos de Escalonamento

Três ambientes merecem distinção:

- **Lote**:
  - Em sistemas em Lote, não há terminal com usuários esperando impacientemente por uma resposta rápida. Consequentemente, algoritmos com longo intervalo de tempo (ou que executam até terminar) para cada processo em geral são aceitáveis.
- **Interativo**:
  - Em um ambiente com usuários interativos, a preempção é essencial para evitar que um processo se aposse da CPU e, com isso, negue serviço aos outros.
- **Tempo real**:
  - Em sistemas com restrição de tempo real, a preempção é desnecessária algumas vezes, pois os processos sabem que não podem executar por longos períodos de tempo e em geral fazem seus trabalhos e bloqueiam rapidamente.

Categorias de Algoritmos de Escalonamento

Adicionalmente, alguns critérios se adequam aos seus ambientes de utilização

- **Sistemas em Lote**:
  - **Vazão (Througput):**
    - Maximizar o número de tarefas/Jobs por hora
  - **Tempo de retorno**:
    - Minimizar o tempo entre a submissão e o término da execução do processo.
  - **Utilização da CPU:**
    - Manter a CPU ocupada todo o tempo.

Categorias de Algoritmos de Escalonamento

Adicionalmente, alguns critérios se adequam aos seus ambientes de utilização

- **Sistemas Interativos**:
  - **Tempo de resposta:**
    - Responder rapidamente às requisições
  - **Proporcionalidade**:
    - Satisfazer as perspectivas dos usuários

Categorias de Algoritmos de Escalonamento

Adicionalmente, alguns critérios se adequam aos seus ambientes de utilização

- **Sistemas de Tempo Real**:
  - **Cumprimento dos prazos:**
    - Evitar perda de dados
  - **Previsibilidade**:
    - Evitar a degradação da qualidade em sistemas multimídias

Algoritmos de Escalonamento

- **Algoritmos não preemptivos**
  - First In First Out (FIFO/ FCFS)
  - Shortest Job First (SJF)
- **Algoritmos preemptivos**
  - Escalonamento Round Robin (circular)
  - Baseado em prioridades
- Existem outros algoritmos:
  - Shortest Remaining Time (SRT)
  - High Response Ratio Next (HRRN)
  - Escalonamento de frequência de CPU (Ondemand e Interactive – LINUX/ANDROID)

Simulador SOSim

- http://www.training.com.br/sosim/