

Sistemas Operacionais

# Mecanismo de Escalonamento

**Lesandro Ponciano**

# Objetivos da Aula

- Revisar conceitos relevantes ao escalonamento
- Apresentar o escalonador da CPU
- Discutir o escalonamento com e sem preempção
- Analisar os critérios de escalonamento

# Tipos de Escalonadores

- **Lote**
  - Processos são armazenados no disco para posterior execução
  - Forma-se um *spool* de processos
- **Longo prazo** (ou escalonador de tarefas)
  - Seleciona processos no *spool* e os carrega para a memória no estado pronto
- **Curto prazo** (ou escalonador da CPU)
  - Seleciona um processo entre os que estão prontos, e aloca a CPU a ele

# Escalonador de Longo Prazo

- É executado
  - Com pouca frequência, alguns minutos
  - Quando algum processo termina, há liberação de espaço na memória
- É mais lento
  - Tanto para decidir quanto para executar
  - Pode demorar alguns segundos

# Escalonador de Longo Prazo

- Controla o grau de **multiprogramação**
  - Idealmente, a quantidade de processos entrando na memória é igual à quantidade de processos saindo dela
  - Deve fazer uma escolha cuidadosa, com um *mix* de
    - Processos intensivos em I/O (ou *I/O intensive*)
    - Processos intensivos em CPU (ou *CPU intensive*)
- Pode não existir em alguns sistemas

# Escalonador de Curto Prazo

- É executado frequentemente
  - Ex.: uma vez a cada 100 milissegundos
  - 1 segundo = 1000 milissegundos
- Precisa ser rápido
  - Se demorar muito, tomará tempo dos processos do usuário
    - Significa que o escalonador está sendo muito intrusivo
- Busca a alta utilização da CPU
  - Sempre existir algum processo usando a CPU

# Utilização da CPU

- Sem multiprogramação
  - Um processo é executado por vez
  - Em algum momento, o processo vai realizar E/S
  - A CPU fica ociosa
- Com multiprogramação
  - O tempo que a CPU fica ociosa é usado para outro processo
  - Busca-se haver sempre algum processo pronto para execução para que a utilização da CPU seja otimizada

# Escalonamento

- O objetivo de multiprogramação
  - Sempre existir algum processo em execução para a otimização do uso da CPU
- O objetivo de compartilhamento de tempo
  - Alternância de processos na CPU com tanta frequência que os usuários possam interagir com eles enquanto estão sendo executados
- O escalonador de processos contribui para esses atingir esses objetivos



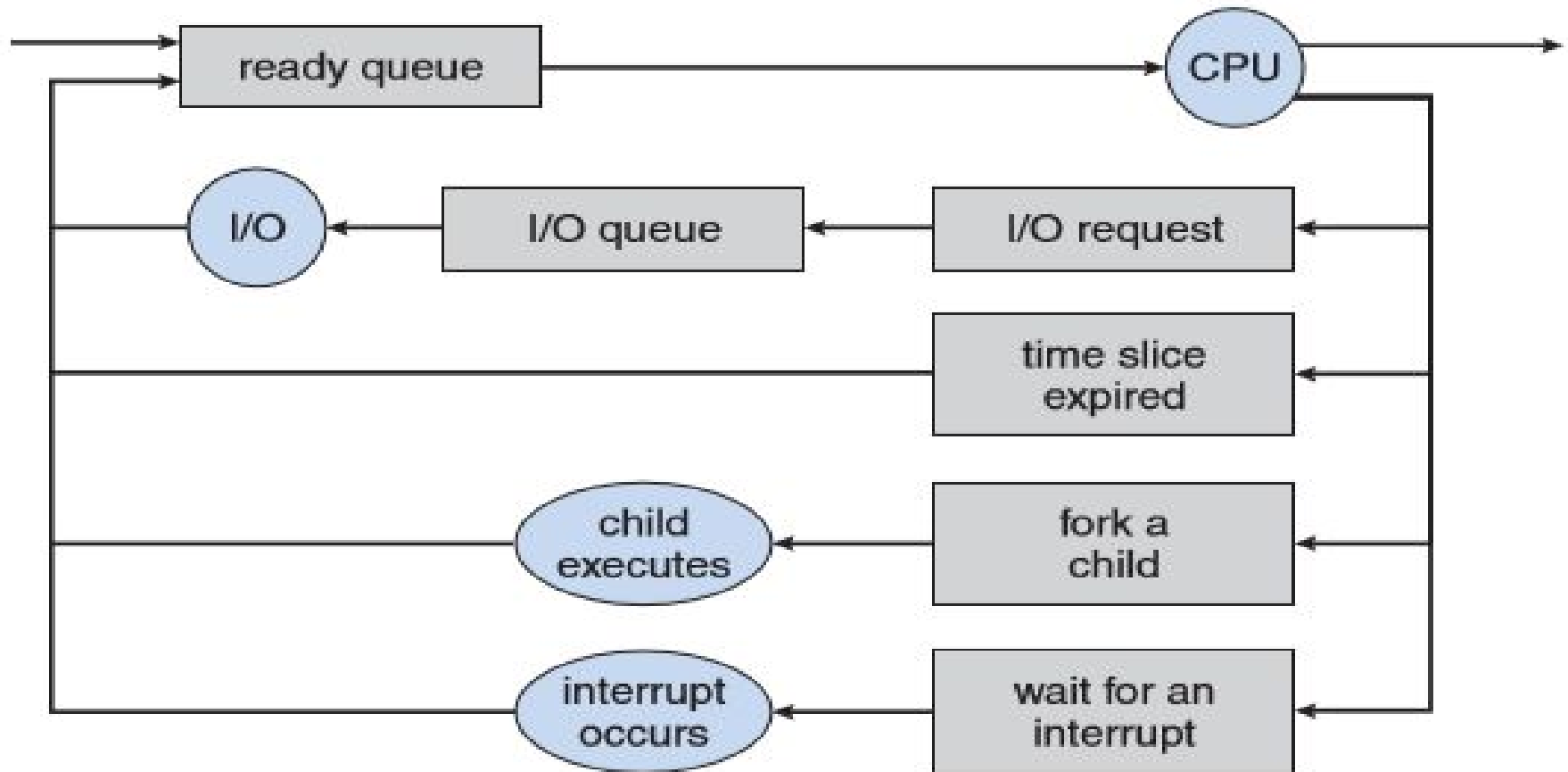
# Filas de Escalonamento

- Fila de tarefas
  - É composta por todos os processos em execução
- Fila de prontos
  - É composta por todos os processos que estão na memória em estado pronto
- Fila de dispositivo
  - É composta por todos os processos que estão aguardando para usar um dado dispositivo

# Fluxo nas Filas

- Inicialmente um processo fica na fila de prontos, até ser escalonado para a CPU
- Após o processo executar na CPU, um entre vários eventos pode ocorrer
  - O processo pode emitir uma solicitação de I/O e, então, ser inserido em uma fila de I/O
  - O processo pode criar um novo subprocesso e esperá-lo terminar
  - O processo pode ser removido à força da CPU (interrupção e/ou *timeout*)

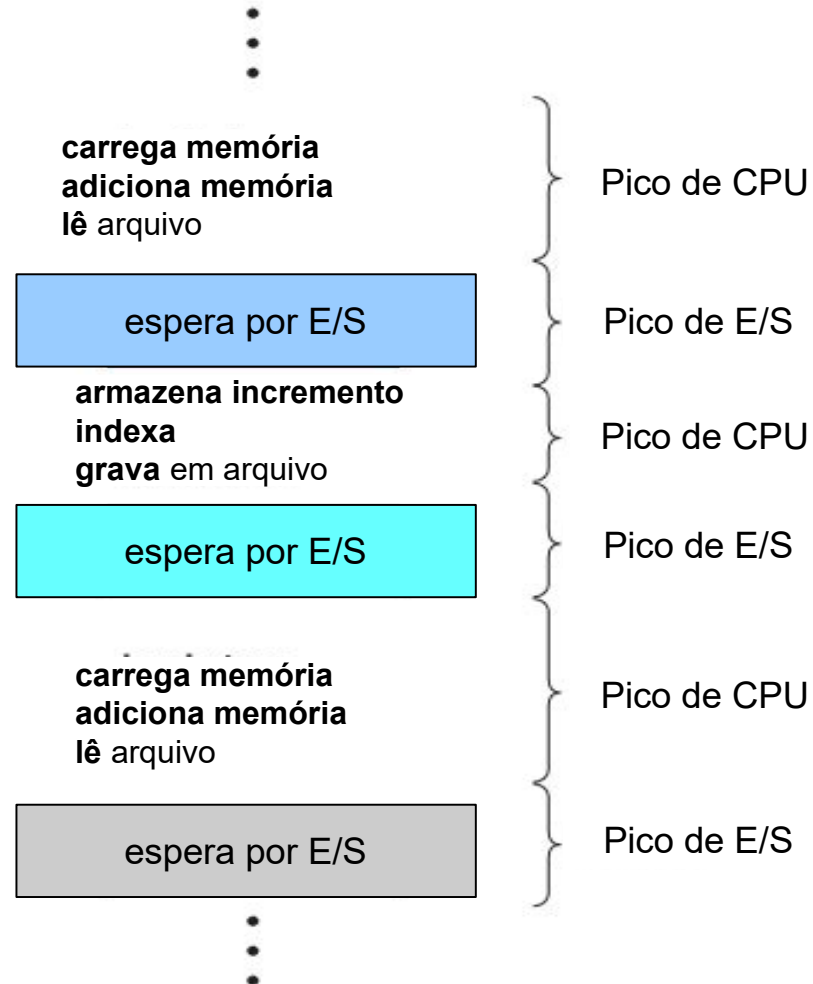
# Diagrama de Enfileiramento



# Ciclos de CPU e E/S

- Propriedade dos processos
  - Ciclos de execução da CPU
  - Ciclos de espera por E/S
  - Alternância entre esses ciclos
- Todo processo começa e termina com ciclo de CPU
  - Criação do processo
  - Término do processo
- Alguns processos podem ser intensivos em CPU e outros intensivos em E/S

# Alternância entre Picos



# Escalonador da CPU

- Sempre que a CPU fica ociosa, o SO
  - seleciona um processo entre os processos na memória que estão prontos para execução
  - aloca a CPU a esse processo
- A parte do sistema operacional que faz essa seleção e alocação é o **escalonador**

# Eventos de Escalonamento

- Quatro eventos originam uma decisão de escalonamento
  - 1) Um processo passa do estado de execução para o estado de espera
  - 2) Um processo passa do estado de execução para o estado pronto
  - 3) Um processo passa do estado de espera para o estado pronto
  - 4) Quando um processo termina

# Preempção

- Escalonamento **sem preempção**
  - A decisão de escalonamento ocorre apenas nas situações 1 e 4 já apresentadas
  - A CPU é utilizada por um processo até que ele termine ou até que ele entre em estado de espera
- Escalonamento **com preempção**
  - A decisão de escalonamento também ocorre nas situações 2 e 3 já apresentadas
  - A CPU pode ser retirada de um processo para ser alocada a outro que está na fila de prontos
    - Ex. quando ocorre uma intervenção (2) ou término de uma espera (3)



# Despachante

- O despachante é o módulo que passa o controle da CPU para o processo selecionado pelo escalonador de curto prazo, o que envolve
  - Troca de contexto
  - Mudança para a modularidade de usuário
  - Salto para a locação apropriada do programa do usuário para que ele seja reiniciado

# Avaliação do Escalonamento

- Utilização da CPU

- Manter a CPU tão ocupada quanto possível
- Conceitualmente a ocupação máxima é de 100%

- *Throughput*

- Maximizar a quantidade de processos que são concluídos por unidade de tempo

- Tempo de *turnaround*

- Trata-se do total de tempo decorrido entre o momento em que o processo é iniciado e o tempo em que é concluído
- Busca-se minimizar esse total de tempo

# Avaliação do Escalonamento

- **Tempo de espera**
  - Trata-se da soma do tempo que o processo permanece esperando na fila de prontos
  - Busca-se minimizar esse tempo
- **Tempo de resposta**
  - Trata-se do tempo decorrido entre o envio da primeira requisição e a primeira resposta ser obtida
  - Busca-se minimizar esse tempo

# Atividade de Fixação

1. Por que a existência de processos com picos de CPU e picos de espera por E/S favorece o paralelismo no sistema?
2. Qual a diferença do escalonador de longo prazo e o escalonador de curto prazo?
3. Qual a diferença do escalonamento com preempção e o escalonamento sem preempção?

# Referências

TANENBAUM, Andrew S. Sistemas operacionais modernos. 3. ed. São Paulo: Pearson Prentice Hall, 2009. xvi, 653 p. ISBN 9788576052371

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Fundamentos de sistemas operacionais: princípios básicos. Rio de Janeiro, RJ: LTC, 2013. xvi, 432 p. ISBN 9788521622055

PONCIANO, L; Brasileiro, Francisco . Assessing Green Strategies in Peer-to-Peer Opportunistic Grids. Journal of Grid Computing, v. 11, p. 129-148, 2013.

Ponciano, Lesandro; Brito, Andrey ; Sampaio, Livia ; Brasileiro, Francisco. Energy Efficient Computing through Productivity-Aware Frequency Scaling. In: 2012 International Conference on Cloud and Green Computing (CGC), 2012, Xiangtan. p. 191-198.

# Sistemas Operacionais

**Prof. Dr. Lesandro Ponciano**

<https://orcid.org/0000-0002-5724-0094>