

SISTEMAS OPERACIONAIS

AULA 05: ESCALONAMENTO DE PROCESSOS

28 de maio de 2025

Prof. Me. José Paulo Lima

IFPE Garanhuns



**INSTITUTO
FEDERAL**
Pernambuco

SUMÁRIO I



Escalonamento de processos

Quando escalonar?

Funções básicas

Objetivos

Dispatcher

Preemptivo x Não-preemptivo

Tipos de algoritmos

SUMÁRIO II



Algoritmos de escalonamento

Escalonamento em Lotes

First-Come, First-Served

Shortest Job First

Shortest Remaining Time Next

Escalonamento em Sistemas Interativos

Round-Robin Scheduling

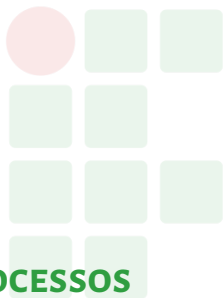
Priority Scheduling

Multiple Queues

Lottery Scheduling

Escalonamento em Sistemas de Tempo Real

Referências



ESCALONAMENTO DE PROCESSOS

**INSTITUTO
FEDERAL**
Pernambuco

ESCALONAMENTO DE PROCESSOS



- ▶ Quando um computador é multiprogramado, ele frequentemente tem múltiplos processos ou threads competindo pela CPU ao mesmo tempo;
 - ▶ Essa situação ocorre sempre que dois ou mais deles estão simultaneamente no estado pronto.
- ▶ Se apenas uma CPU está disponível, uma escolha precisa ser feita sobre qual processo será executado em seguida;
- ▶ A parte do sistema operacional que faz a escolha é chamada de escalonador, e o algoritmo que ele usa é chamado de algoritmo de escalonamento.

ESCALONAMENTO DE PROCESSOS

INTRODUÇÃO



- ▶ Escalonamento em Sistemas em *Batch*:
 - ▶ Escalonamento simples;
 - ▶ Apenas executa a próxima tarefa que esta na fila.
- ▶ Escalonamento em Sistemas multiprogramados:
 - ▶ O algoritmo de escalonamento se tornou mais complexo;
 - ▶ Vários usuários esperando por um serviço.
- ▶ Um bom escalonador pode fazer a diferença no desempenho atendendo os usuários de forma eficiente;

ESCALONAMENTO DE PROCESSOS

QUANDO ESCALONAR?



1. Quando se cria um novo processo:
 - ▶ É necessário tomar uma decisão entre executar processos pai ou o processo filho.
2. Término de um processo:
 - ▶ Quando um processo termina sua execução um outro é escolhido entre os processos prontos.
3. Quando um processo bloqueia para E/S:
 - ▶ Outro processo precisa ser selecionado. O motivo da espera pode influenciar na escolha.
 - ▶ Processo A importante pode espera o processo B sair da sua região crítica.
4. Quando ocorre uma interrupção de E/S:
 - ▶ Se a interrupção vem de um dispositivo de E/S que acabou de fazer o seu trabalho, o processo que estava bloqueado pode agora ficar pronto para a execução.

ESCALONAMENTO DE PROCESSOS

QUANDO ESCALONAR?

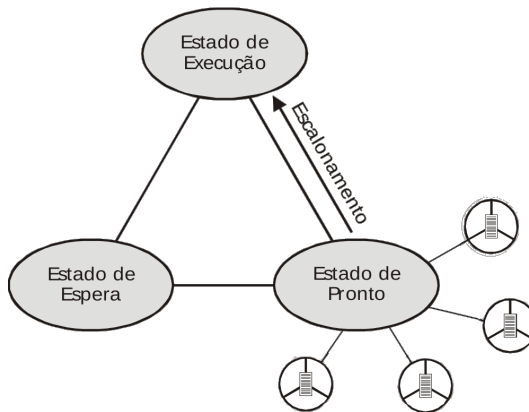


Figura extraída de Machado e Maia (2017, p. 135).

ESCALONAMENTO DE PROCESSOS

FUNÇÕES BÁSICAS



- ▶ Manter a UCP ocupada a maior parte do tempo;
- ▶ Balancear o uso da UCP entre processos;
- ▶ Privilegiar a execução de aplicações críticas;
- ▶ Maximizar o *throughput*;
 - ▶ “*Throughput* representa o número de processos executados em um determinado intervalo de tempo. Quanto maior o *throughput*, maior o número de tarefas executadas em função do tempo. A maximização do *throughput* é desejada na maioria dos sistemas.” (MACHADO; MAIA, 2017, p. 137).
- ▶ Oferecer tempos de resposta razoáveis para usuários interativos;
- ▶ Implementada pelo *scheduler* (escalonador) e *dispatcher* (dispachante).

ESCALONAMENTO DE PROCESSOS

FUNÇÕES BÁSICAS



- ▶ Tempo de Processador / Tempo de UCP
 - ▶ “é o tempo que um processo leva no estado de execução durante seu processamento. As políticas de escalonamento não influenciam o tempo de processador de um processo” (MACHADO; MAIA, 2017, p. 137).
- ▶ Tempo de Espera
 - ▶ “é o tempo total de um processo permanece na fila de pronto durante seu processamento, aguardando para ser executando.” (MACHADO; MAIA, 2017, p. 137).
- ▶ Tempo de *Turnaround*
 - ▶ “é o tempo que um processo leva desde de sua criação até ao seu término, levando em consideração todo o tempo gasto na espera para alocação de memória, espera na fila de pronto (tempo de espera), processamento na UCP (tempo de processador)” (MACHADO; MAIA, 2017, p. 137).

ESCALONAMENTO DE PROCESSOS

OBJETIVOS



► Todos os sistemas:

- Justiça: dar a cada processo uma porção justa da CPU;
- Aplicação da política: verificar se a política estabelecida é cumprida;
- Equilíbrio: manter ocupadas todas as partes do sistema.

► Sistemas em lote:

- Vazão (*throughput*): maximizar o número de tarefas por hora;
- Tempo de retorno: minimizar o tempo entre a submissão e o término;
- Utilização de CPU: manter a CPU ocupada o tempo todo.

► Sistemas interativos:

- Tempo de resposta: responder rapidamente às requisições;
- Proporcionalidade: satisfazer às expectativas dos usuários.

► Sistemas de tempo real:

- Cumprimento dos prazos: evitar a perda de dados;
- Previsibilidade: evitar a degradação da qualidade em sistemas multimídia.

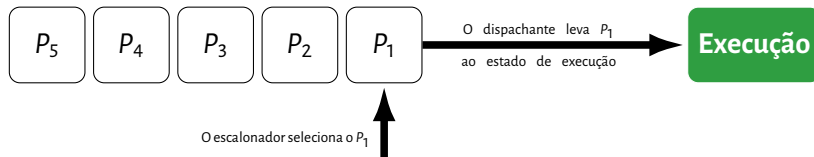
ESCALONAMENTO DE PROCESSOS

DISPATCHER



- ▶ O escalonador é responsável pela escolha do processo “eleito” para usar a CPU;
- ▶ O dispatchante é responsável pelo lado técnico de:
 - ▶ Salvar o processo que estava usando a CPU;
 - ▶ Executar o processo eleito na CPU;
 - ▶ Isso se chama efetuar a troca de contexto.

Fila de processos



ESCALONAMENTO DE PROCESSOS

PREEMPTIVO X NÃO-PREEMPTIVO



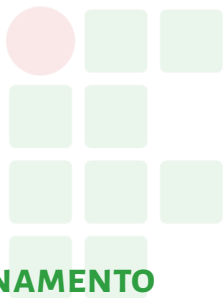
- ▶ **Preemptivo**
 - ▶ Escolhe um processo e o deixa em execução por um tempo máximo fixado (*quantum*);
 - ▶ Se ao final do tempo o processo ainda estiver executando, o processo será suspenso e o escalonador escolhe outro processo para executar.
- ▶ **Não-preemptivo**
 - ▶ Processo executa até bloquear (E/S);
 - ▶ Ou executa até liberar CPU voluntariamente;
 - ▶ Ou se um processo de mais alta prioridade estava aguardando o *clock*.

ESCALONAMENTO DE PROCESSOS

TIPOS DE ALGORITMOS



- ▶ Lote:
 - ▶ Não possuem usuários aguardando a vez com frequência;
 - ▶ Algoritmos preemptivos e não preemptivos com longo intervalo de tempo para cada processo são aceitáveis.
- ▶ Interativo:
 - ▶ Preempção essencial;
 - ▶ Servidores caem no mesmo grupo.
- ▶ Tempo real:
 - ▶ Preempção não é obrigatória;
 - ▶ Os processos sabem que não podem executar por muito tempo;
 - ▶ Fazem o seu trabalho e bloqueiam rapidamente.
 - ▶ Aceitam programas de propósito geral.
 - ▶ Havendo pouco controle sobre o que está sendo executado.



ALGORITMOS DE ESCALONAMENTO

**INSTITUTO
FEDERAL**
Pernambuco

ALGORITMOS DE ESCALONAMENTO EM LOTE

FIRST-COME, FIRST-SERVED



- ▶ Primeiro a chegar, primeiro a ser servido;
- ▶ Não preemptivo;
- ▶ A CPU é atribuída aos processos na ordem que eles chegam;
 - ▶ Um processo ao ser iniciado é autorizado a executar por quanto tempo quiser;
 - ▶ Quando um processo fica bloqueado, o próximo processo da fila entra em execução;
 - ▶ Quando o processo fica pronto, ele é posto no final da fila.
- ▶ Há uma fila única de processos.

ALGORITMOS DE ESCALONAMENTO EM LOTE

FIRST-COME, FIRST-SERVED

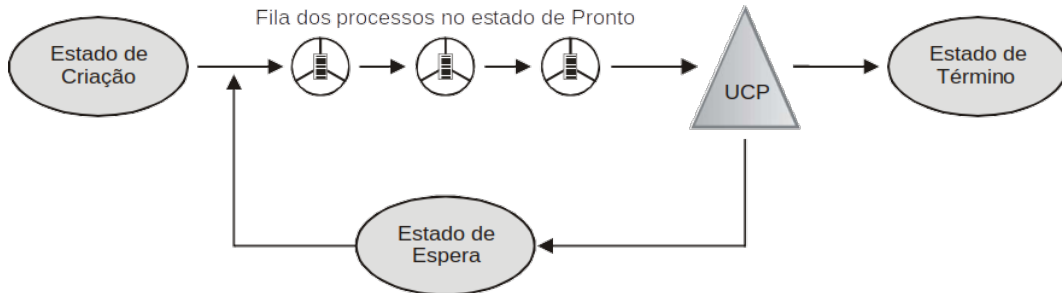
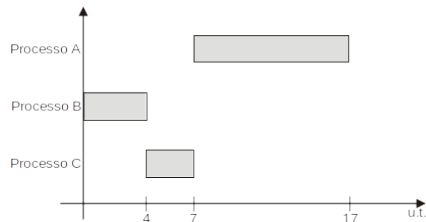
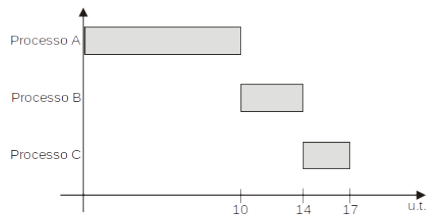


Figura extraída de Machado e Maia (2017, p. 138).

ALGORITMOS DE ESCALONAMENTO EM LOTE

FIRST-COME, FIRST-SERVED



Processo	Tempo de processador (u.t.)
A	10
B	4
C	3

ALGORITMOS DE ESCALONAMENTO EM LOTE

FIRST-COME, FIRST-SERVED



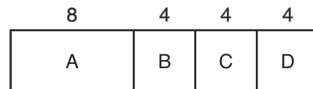
► Problemas:

- O FCFS apresenta uma desvantagem com processos orientados à computação e limitados por E/S;
 - Por exemplo, cada um processo limitado por E/S lê um bloco por segundo, então, demorará vários segundos para terminar.
- Favorece processos limitados em CPU, em vista que processos limitados a E/S têm que esperar pelo fim dos processos limitados em CPU.

ALGORITMOS DE ESCALONAMENTO EM LOTE

SHORTEST JOB FIRST

- ▶ Tarefa mais curta primeiro;
- ▶ Não preemptivo;
- ▶ Todas as tarefas precisam ser conhecidas previamente;
- ▶ Se tarefas curtas chegarem continuamente, as tarefas longas nunca serão escalonadas;
- ▶ Adequado apenas em casos em que todas as tarefas estão disponíveis simultaneamente.



(a)



(b)

Figura extraída de Tanenbaum e Bos (2016, p. 109).

ALGORITMOS DE ESCALONAMENTO EM LOTE

SHORTEST JOB FIRST

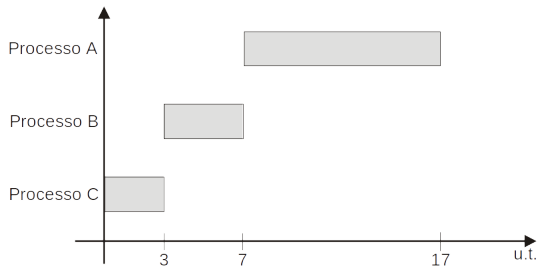


Figura extraída de Machado e Maia (2017, p. 140).

ALGORITMOS DE ESCALONAMENTO EM LOTE

SHORTEST REMAINING TIME NEXT



- ▶ Versão preemptiva do SJF;
- ▶ O escalonador sempre escolhe o processo cujo o tempo de execução restante seja o menor;
- ▶ Quando chega uma nova tarefa, seu tempo total é comparado ao tempo restante do processo em curso;
- ▶ Se, para terminar, a nova tarefa precisa de menos tempo que o processo corrente, então esse será suspenso e a nova tarefa será iniciada.

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

ROUND-ROBIN SCHEDULING



- ▶ Escalonamento por chaveamento circular;
- ▶ Um dos mais antigos, simples e justos;
- ▶ Amplamente utilizado;
- ▶ Preemptivo;
- ▶ Fila de tarefas;
- ▶ Cada processo tem atribuído um intervalo de tempo que é permitido executar: “*Quantum*”.
 - ▶ Se ao final do seu *quantum* o processo ainda estiver em execução:
 - ▶ A CPU sofrerá preempção;
 - ▶ O processo irá para o fim da fila;
 - ▶ Será dada vez de executar a outro processo.

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

ROUND-ROBIN SCHEDULING

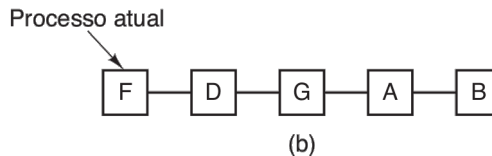
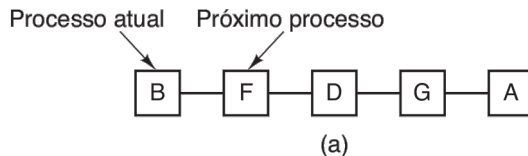


Figura extraída de Tanenbaum e Bos (2016, p. 110).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

ROUND-ROBIN SCHEDULING

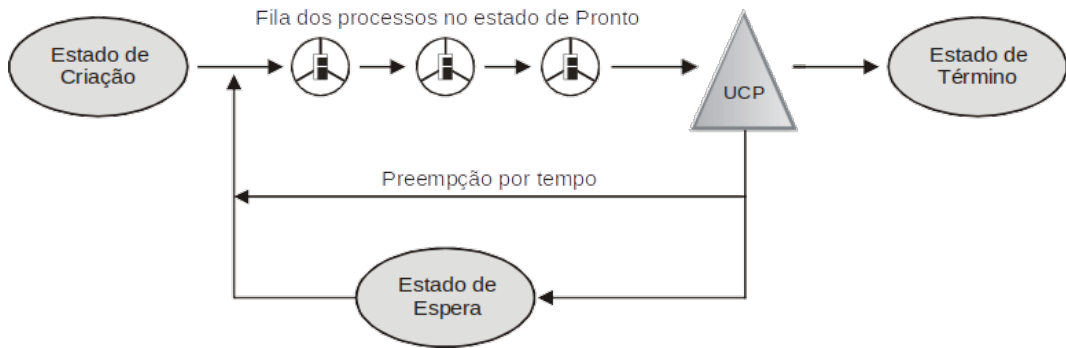


Figura extraída de Machado e Maia (2017, p. 142).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

ROUND-ROBIN SCHEDULING

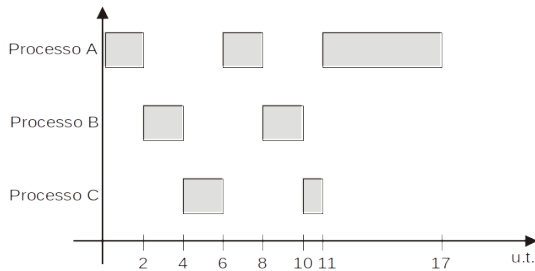
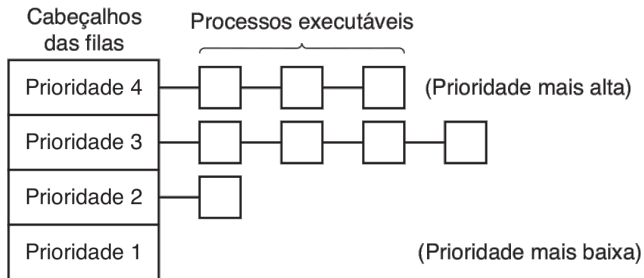


Figura extraída de Machado e Maia (2017, p. 143).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

PRIORITY SCHEDULING

- ▶ Escalonamento por prioridades;
- ▶ O escalonamento circular pressupõe que todos os processos sejam igualmente importantes;
- ▶ A ideia básica é que em cada processo seja atribuída uma prioridade;
- ▶ O processo com maior prioridade é permitido a execução.

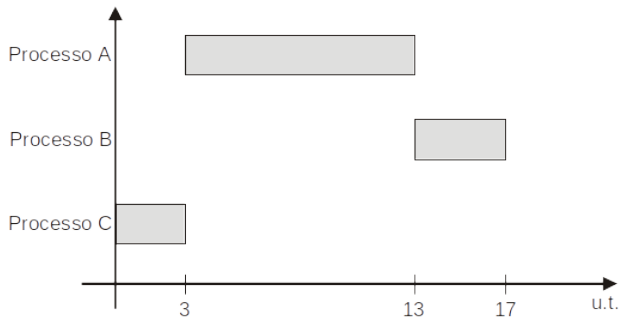


ALGORITMOS DE ESCALONAMENTO INTERATIVOS

PRIORITY SCHEDULING



25



Processo	Tempo de processador (u.t.)	Prioridade
A	10	2
B	4	1
C	3	3

Figura extraída de Machado e Maia (2017, p. 145).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

PRIORITY SCHEDULING

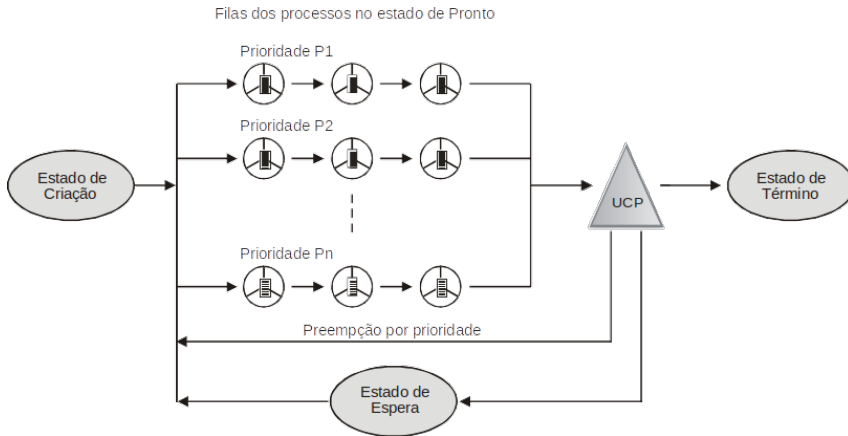


Figura extraída de Machado e Maia (2017, p. 145).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

MULTIPLE QUEUES



- ▶ Múltiplas filas;
- ▶ Filas de prioridades;
- ▶ *Quantum* cedidos em tamanhos diferentes;
- ▶ Quanto maior o *quantum*, menor a prioridade e vai para o final da fila;
- ▶ Se um processo usou todo o seu *quantum* e não acabou, recebe um *quantum* maior para a próxima execução.

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

MULTIPLE QUEUES

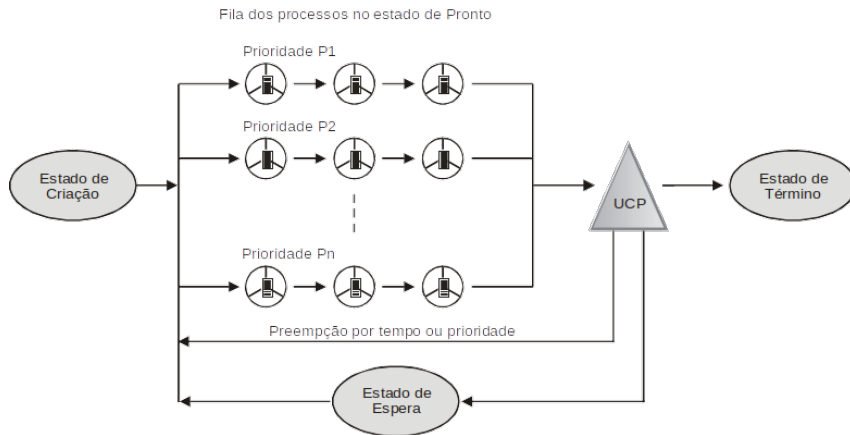


Figura extraída de Machado e Maia (2017, p. 145).

ALGORITMOS DE ESCALONAMENTO INTERATIVOS

LOTTERY SCHEDULING



- ▶ Escalonamento por loteria;
- ▶ Dar bilhetes de loteria aos processos com recursos de máquina como “prêmio”;
- ▶ Bilhete é dado aleatoriamente através de um sorteio;
- ▶ Se forem sorteados quanta, 50 sorteios de bilhetes com 20 ms de *quantum* são feitos e distribuídos;
- ▶ Maior prioridade (mais bilhetes) para processos com maior prioridade;
- ▶ Processos podem ser cooperativos e trocarem de bilhetes entre si.
 - ▶ Exemplo: Processo cliente envia mensagem para um servidor. Para que o servidor execute logo, o cliente envia uns bihetes.
- ▶ Útil para situações complexas.

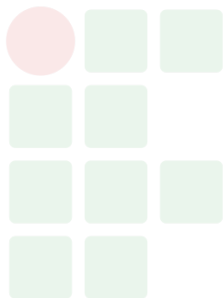
ESCALONAMENTO EM SISTEMAS DE TEMPO REAL

- ▶ Os processos em geral tem vida curta e podem ser executados rapidamente;
- ▶ O escalonador é responsável por escalonar os processos de tal maneira que todos os prazos sejam cumpridos;
- ▶ Resposta a eventos:
 - ▶ Periódicos: ocorrem em intervalos regulares;
 - ▶ Aperiódicos: acontece de modo imprevisível;
- ▶ “Por exemplo, se há m eventos periódicos e o evento i ocorre com o período P_i e exige C_i segundos de tempo da CPU para lidar com cada evento, então a carga só pode ser tratada se:

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

- ▶ Diz-se de um sistema de tempo real que atende a esse critério que ele é **escalonável**.” (TANENBAUM; BOS, 2016, p. 113).



REFERÊNCIAS

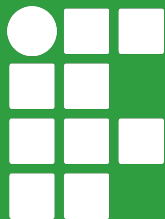


INSTITUTO
FEDERAL
Pernambuco

REFERÊNCIAS I



-  MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 5. ed. Rio de Janeiro: LTC, 2017. ISBN 978-85-216-2210-9.
-  TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016.



INSTITUTO FEDERAL

Pernambuco