

Sistemas Operacionais

Escalonamento

Sumário

- 1 Definição
- 2 Preempção
- 3 Algoritmos

- 4 Algoritmos não Preemptivos
- 5 Algoritmos Preemptivos
- 6 Bibliografia

Definição

Relembrando...

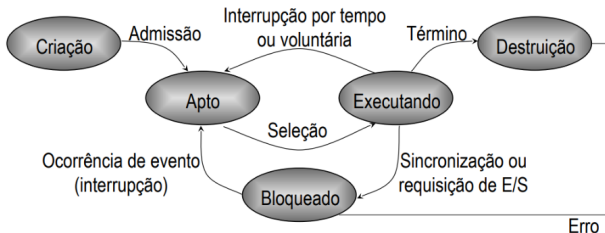
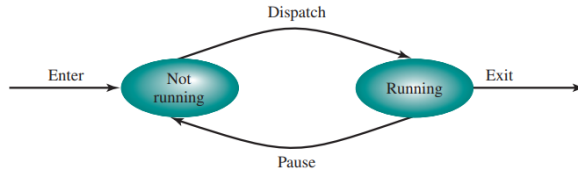
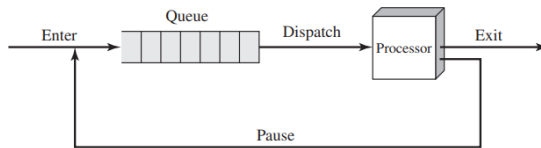


Figura 1: Estados de um processo (SILBERSCHATZ; GALVIN; GAGNE, 2001).

- **Scheduler (Escalonador)**: decide qual processo deve usar a CPU.
- **Dispatcher**: realiza a troca de contexto e entrega a CPU ao processo escolhido.
- Portanto, em resumo, o *scheduler* escolhe, o *dispatcher* executa a escolha.



(a) State transition diagram



(b) Queueing diagram

Figura 2: Estados de um processo (SILBERSCHATZ; GALVIN; GAGNE, 2001).

- Em sistemas multiprogramados, vários processos ou threads podem competir pela CPU quando estão no estado pronto.
- Com apenas uma CPU disponível, o sistema operacional precisa decidir qual processo será executado.
- Essa decisão é feita pelo **escalonador**, usando um algoritmo de escalonamento.



- › Maximizar a utilização do processador
 - » Maximizar a produção do sistema (**throughput**)
 - » Número de processos executados por unidade de tempo
- › Minimizar o tempo de execução (**turnaround**)
 - » Tempo total para executar um determinado processo
- › Minimizar o tempo de espera
 - » Tempo que um processo permanece na lista de aptos
- › Minimizar o tempo de resposta
 - » Tempo decorrido entre uma requisição e a sua realização

Tipo de Tarefa	Resumo
Tarefas de tempo real	Exigem previsibilidade nos tempos de resposta, geralmente associadas a sistemas críticos.
Tarefas interativas	Respondem rapidamente a eventos externos, sem exigência de previsibilidade, como editores de texto e navegadores.
Tarefas em lote (batch)	Executam sem requisitos temporais explícitos, como backups e cálculos longos, sem intervenção do usuário.

Tabela 1: Tipos de tarefas em sistemas operacionais

> Todos:

- » **Justiça:** dar a cada processo uma porção justa da CPU;
- » **Aplicação Política:** verificar se a política de escalonamento está sendo seguida;
- » **Equilíbrio:** manter todas as partes ocupadas.

› Sistemas em lote:

- › **Maximizar a utilização da CPU:** Garantir que o processador esteja sempre em operação.
- › **Maximizar a vazão (Throughput):** Maximizar a quantidade de tarefas concluídas em um certo período de tempo.
- › **Minimizar o tempo de execução:** Medir quantas tarefas são finalizadas dentro de um intervalo.

› Sistemas Interativos:

- › **Reduzir o tempo de resposta:** Diminuir o tempo entre o pedido de uma ação e o momento em que ela é realizada.
- › **Proporcionalidade:** Satisfazer às expectativas dos usuários.

- › **Sistemas de tempo real:**

- › **Cumprimento de prazos:** Evitar perda de dados.
- › **Previsibilidade:** Evitar a degradação da qualidade.

Preempção

› Sistemas preemptivos:

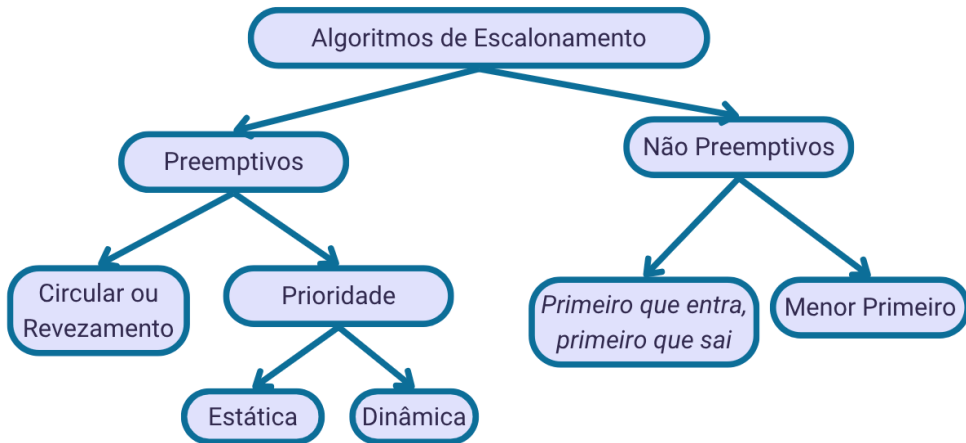
- ›› O processo pode ser interrompido se: **terminar seu tempo, fazer uma chamada de sistema ou uma tarefa de maior prioridade precisar da CPU.**
- ›› O escalonador avalia os processos a cada interrupção para decidir se troca o processo em execução.

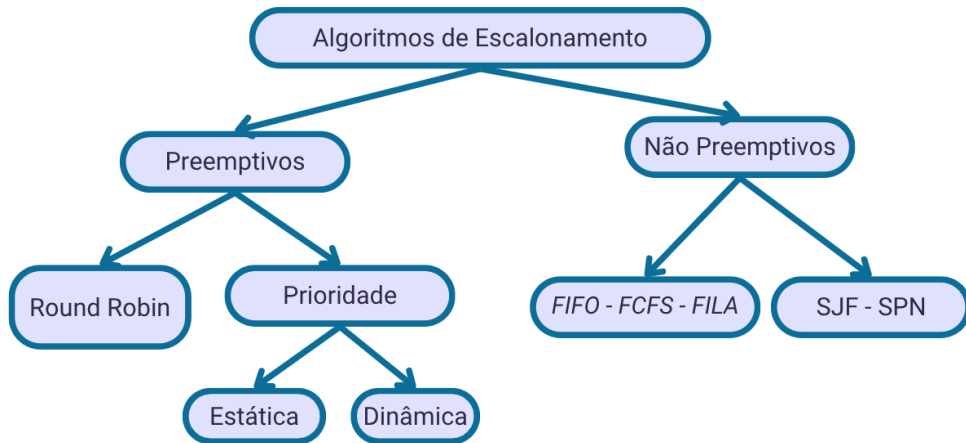
> Sistemas não preemptivos:

- » O processo em execução só libera o processador ao:
 - Terminar;
 - Pedir uma operação de E/S;
 - Liberar explicitamente o processador.

Algoritmos

- › Algoritmos não preemptivos (cooperativos):
 - » *First-In First-Out (FIFO) / First Come First-Served (FCFS)*
 - » *Shortest Job First (SJF) / Shortest Process Next (SPN)*
- › Algoritmos preemptivos:
 - » *Round Robin (circular)*
 - » *Prioridades*





Considere os seguintes processos para analisar os algoritmos de escalonamento: ¹

Processo	T1	T2	T3	T4	T5
Ordem de Chegada ²	0	0	1	3	5
Tempo de Execução	5	2	4	1	2
Prioridade ³	2	3	1	4	5

Tabela 2: Tabela de processos para análise de escalonamento

Nota:

Quantum: Duas unidades de tempo discreto.

¹Nesse cenário, será considerado um sistema monoprocessoado e sem interrupções.

²Se os processos chegam ao mesmo tempo, o critério de desempate é o índice. Exemplo: T1 e T2 chegam ao mesmo tempo, T1 executa primeiro.

³Escala de Prioridade entre 1 e 5. Quanto maior o valor maior a prioridade.

O **tempo de espera** de um processo é o período que ele permanece na **fila de aptos** antes de começar a execução.

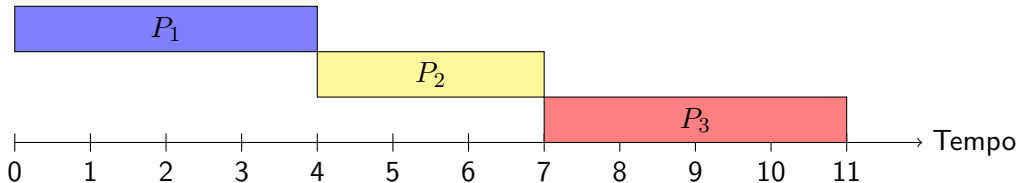
Fórmula do tempo médio de espera:

$$T_{\text{espera médio}} = \frac{\sum_{i=1}^n T_{\text{espera},i}}{n}$$

- › $T_{\text{espera},i}$: tempo de espera do processo i na fila
- › n : número total de processos

Tempo Médio de Espera na Fila II

Exemplo com 3 processos: P_1 , P_2 , P_3



Cálculo do tempo médio de espera:

$$T_{\text{médio}} = \frac{0 + 4 + 7}{3} = \frac{11}{3} \approx 3.67$$

Algoritmos não Preemptivos

First-In First-Out (FIFO) / First Come First-Served (FCFS)

- O primeiro processo a chegar na fila de aptos é o primeiro a executar.
- Em resumo: FILA.

- › Processos aptos são inseridos no final da fila.
- › O processo no início da fila será o próximo a ser executado.
- › O processo executa até:
 - ›› Liberar o processador;
 - ›› Ser bloqueado;
 - ›› Encerrar.

First-In First-Out (FIFO)

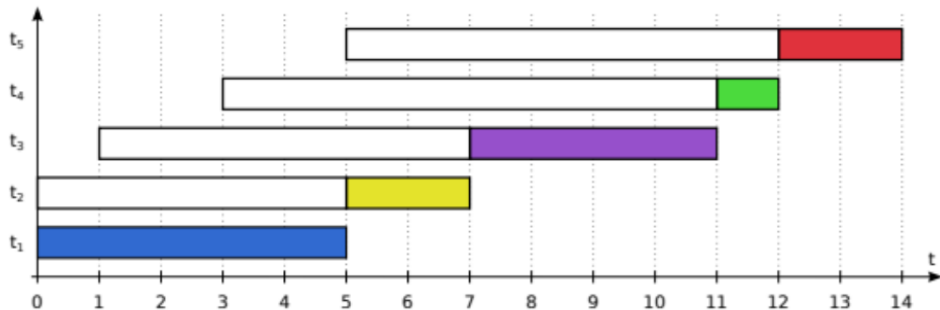


Figura 3: FIFO (Maziero, 2019).

- › Processos que utilizam entrada e saída podem ser prejudicados.
- › **Exemplo:**
 - ›› Processo que utiliza mais tempo de CPU, como um algoritmo de ordenação.
 - ›› Processo que utiliza mais tempo de E/S, como um servidor web.

Shortest Job First (SJF) I

Shortest Process Nest (SPN)

O processo mais curto é o primeiro a executar.

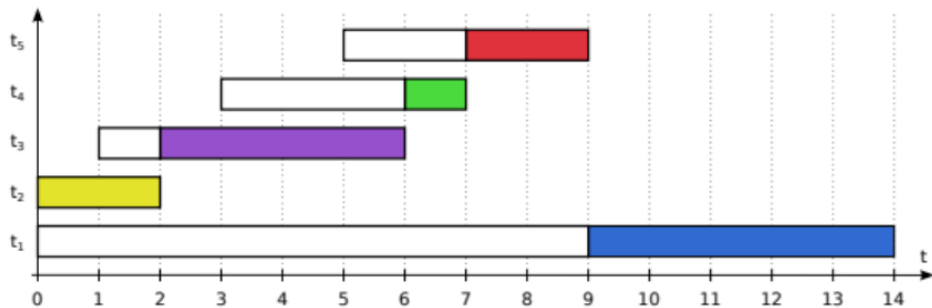


Figura 4: SJF (Maziero, 2019).

› *Shortest Job First (SJF) / Shortest Process Next (SPN)*

- » Solução ótima quando todos os processos estão disponíveis.
- » Dificuldade para estimar o tempo de duração de um processo.
- » É possível usar o histórico de execuções anteriores para prever a duração do processo atual.
- » Quando a chegada de processos mais curtos é maior que a de processos longos, pode causar inanição desses processos.

Shortest Remaining Time First (SRTF) I

- O escalonador compara o tempo previsto de novos processos com o tempo restante dos processos atuais.
- Se o novo processo for mais rápido, ele ganha acesso ao processador.

Shortest Remaining Time First (SRTF) II

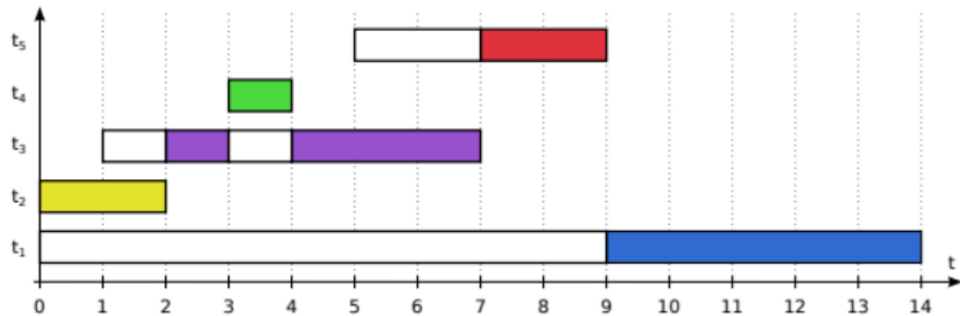


Figura 5: SRTF (Maziero, 2019).

Algoritmos Preemptivos

- Semelhante ao *FIFO / FCFS*, porém a preempção é incluída neste algoritmo (*quantum*).

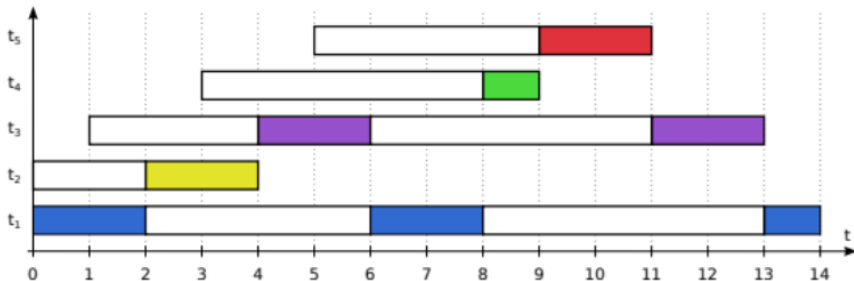


Figura 6: RR (Maziero, 2019).

- › Aumenta o tempo médio de execução para aplicações de tarefas em lote.
- › Pode proporcionar tempo de execução melhor para aplicações interativas.
- › Aumento de troca de contexto.
- › O processo libera a CPU quando:
 - › Libera a CPU;
 - › É bloqueado;
 - › Encerra;
 - › Ocorre preempção.

Prioridades:

- O algoritmo de escalonamento por prioridade define um modelo mais genérico de escalonamento, que permite modelar várias abordagens.
 - » *FIFO (FCFS)*
 - » *SJF (SPN)*
 - » *RR (Round Robin)*

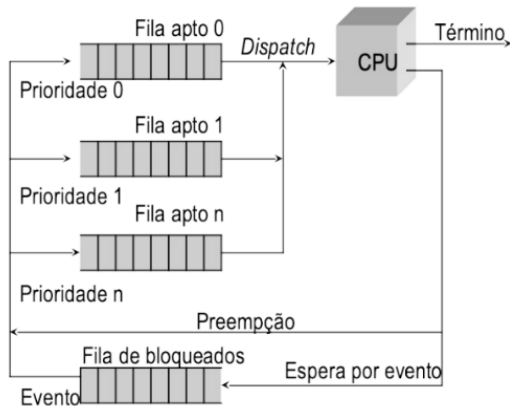


Figura 7: Múltiplas Prioridades (SILBERSCHATZ; GALVIN; GAGNE, 2001).

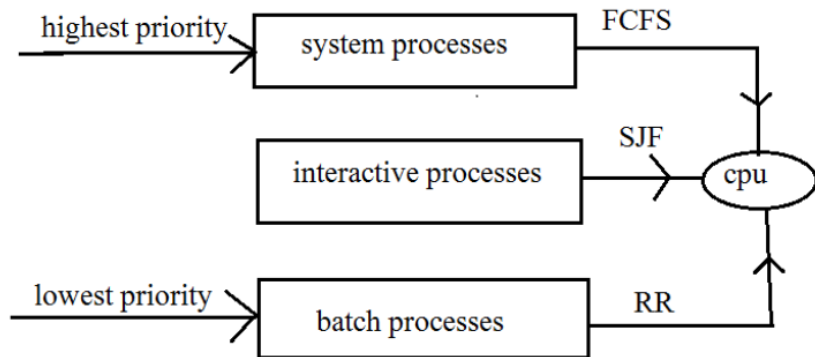


Figura 8: Exemplo de Prioridades (SILBERSCHATZ; GALVIN; GAGNE, 2001).

Estática:

- › Definida na criação do processo.
- › Não muda até o término da execução.
- › Processos de baixa prioridade podem nunca executar.
- › Isso é chamado de *starvation* (inanição).

Dinâmica:

- › Usar *aging* (envelhecimento).
- › A prioridade aumenta conforme o tempo de espera.
- › Considera tempo de espera ou uso de CPU.
- › Melhora a *justiça* e evita *inanição*.

Prioridade Dinâmica

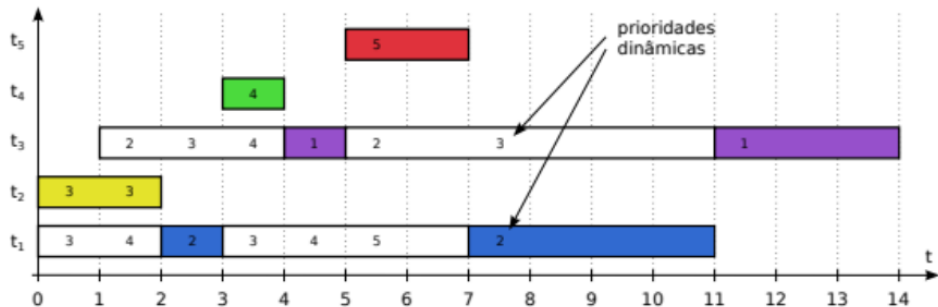


Figura 9: Prioridade Dinâmica (Maziero, 2019).

Capítulo 4 (OLIVEIRA; CARISSIMI; TOSCANI, 2001)



Bibliografia



OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas Operacionais**. 2. ed. Porto Alegre: Sagra-Luzzatto, 2001.



STALLINGS, William. **Operating Systems: Internals and Design Principles**. 6. ed. Upper Saddle River, NJ: Prentice-Hall, 2009.



TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

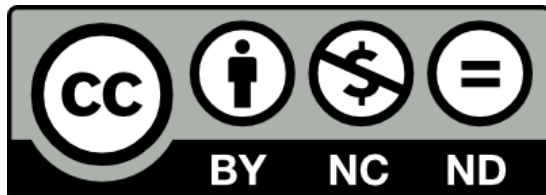
 SILBERSCHATZ, Abraham; GALVIN, Peter; GAGNE, Greg. **Sistemas Operacionais: Conceitos e Aplicações**. Rio de Janeiro: LTC, 2001.

 TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais: Projeto e Implementação**. 2. ed. Porto Alegre: Bookman, 2000.



MAZIERO, Carlos Alberto. **Sistemas Operacionais: Conceitos e Mecanismos [recurso eletrônico]**. Curitiba: DINF - UFPR, 2019. ISBN 978-85-7335-340-2.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Sistemas Operacionais

Escalonamento