


Sistemas Operacionais

Maria Helena Schneid Vasconcelos

maria.vasconcelos@sertao.ifrs.edu.br

09/11/2021



Conteúdo Programático

1 - Histórico de Sistemas Operacionais

2 - Tipos de Sistemas Operacionais e suas características.

1ª Etapa

3 - Gerência de processador.

4 - Gerência de memória.

5- Gerência de arquivos.

6 - Gerência de entrada e saída.

7 – Estudo de Caso.

2ª Etapa



3 - Gerência de processador

- A gerência do processador pode ser considerada a atividade mais importante de um Sistema Operacional.
- Geralmente temos 2 ou mais processos aptos a utilizar o processador para serem executados.



Gerência de processador

- Nesse instante, o sistema operacional deve decidir qual dos processos aptos, armazenados em uma fila, será escolhido para rodar primeiro..
- Essa tarefa e a tomada de decisão é feita pelo escalonador de processos (parte do sistema operacional) através da implementação de alguns algoritmos de seleção, denominados algoritmos de escalonamento.



Conceito de Processos

- Um processo é um fluxo de controle sequencial e seu espaço de endereçamento (registradores, memória e arquivos).
- Informalmente um processo é a execução de um programa junto com os dados usados por ele.
- **A criação, execução e o encerramento de processos são tratados diretamente pelo sistema operacional.**



Escalonador de processos

- O escalonador é a entidade do sistema operacional responsável por selecionar um processo apto a executar no processador e dividir o tempo do processador de forma justa entre os processos que estão aptos.
- O objetivo dos escalonadores é implementar uma política de escalonamento de processos.



Escalonador de processos

- O sistema operacional possui um módulo responsável por efetuar a troca de contexto entre a execução de processos distintos, chamado de Dispatcher (**expeditor**).
- O escalonador está relacionado com a implementação e aplicação das políticas de seleção adotadas.



Objetivos do escalonamento

1. Maximizar a utilização do processador ;
2. Privilegiar aplicações que são críticas;
3. Maximizar a produção do sistema, com o maior número de processos executados por unidade de tempo;

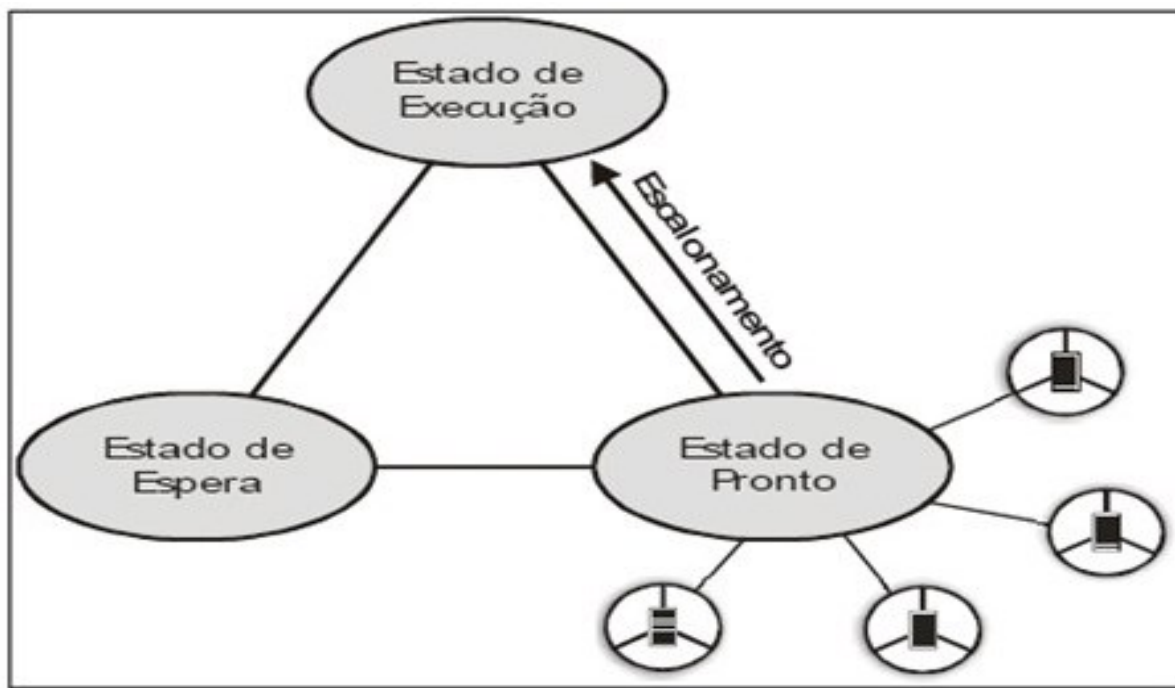


Objetivos do escalonamento

4. Minimizar o tempo de execução, ou seja, o tempo que um processo gasta desde a sua criação até seu término;
5. Minimizar o tempo de espera, ou seja, o tempo que um processo permanece na lista de aptos ;
6. Minimizar o tempo de resposta, ou seja, o tempo decorrido entre uma requisição e a sua realização.



Objetivos do escalonamento



Tipos de escalonadores

Existem 2 tipos de escalonadores:

- 1. Não-preemptivo:** escalonadores que permitem que os processos rodem até o fim de sua execução sem ser interrompidos por eventos externos;
- 2. Preemptivo:** escalonadores que são capazes de suspender processos que poderiam continuar executando.



Tipos de escalonadores

Para cada um desses tipos, os processos poderão utilizar o processador até que:

- No caso do tipo **Não preemptivo:**

1. Término de execução do processo;



2. Execução de uma requisição de entrada/saída ou sincronização;

3. Liberação voluntária do processador a outro processo.



Tipos de escalonadores

- No caso do tipo **Preemptivo**:

1. Término de execução do processo;
2. Execução de uma requisição de entrada/saída ou sincronização;
3. Liberação voluntária do processador a outro processo;



4. Interrupção de relógio;

5. Processo de mais alta prioridade esteja pronto para executar.



Algoritmos de escalonamento

- Diversos mecanismos (algoritmos) foram sendo desenvolvidos ao longo dos anos.
- Cada um possui uma vantagem ou desvantagem, mas os escolhidos são aqueles que oferecem um bom desempenho, ou seja, evitam ao máximo o tempo de espera e mantêm os recursos ocupados em ambientes de processos heterogêneos



Algoritmos de escalonamento

Alguns algoritmos de escalonamento:

- Algoritmos **não preemptivos**:

1. FIFO

2. SJF

3. Cooperativo



Algoritmos preemptivos:

1. Round robin (circular)
2. Múltiplas filas (e suas variações)



Algoritmo escalonamento FIFO

O algoritmo **FIFO (First in – First out)** mais simples de implementar, onde o processador possui uma fila associada para armazenar os processos que estão aptos a executar.

Funcionamento:

1. Processos que se tornam aptos são inseridos no final da fila;



2. Processo que está no início da fila é o próximo a executar;

3. Processo executa até que:

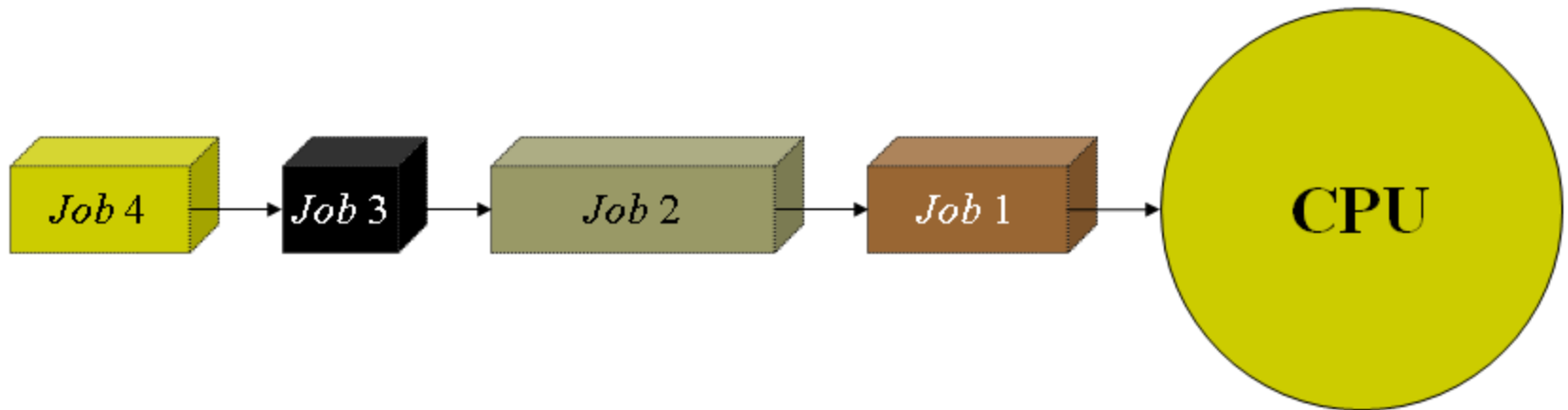
a. Libere explicitamente o processador

b. Realize uma chamada de sistema (bloqueado)

c. Termine sua execução.



Algoritmo escalonamento FIFO



- Não há preempção, ou seja, o próximo da fila só é atendido quando o atual tiver encerrado todas as suas operações.

Algoritmo escalonamento FIFO

- Desvantagem: O tempo médio de espera na fila de execução depende a ordem:

Exemplo:

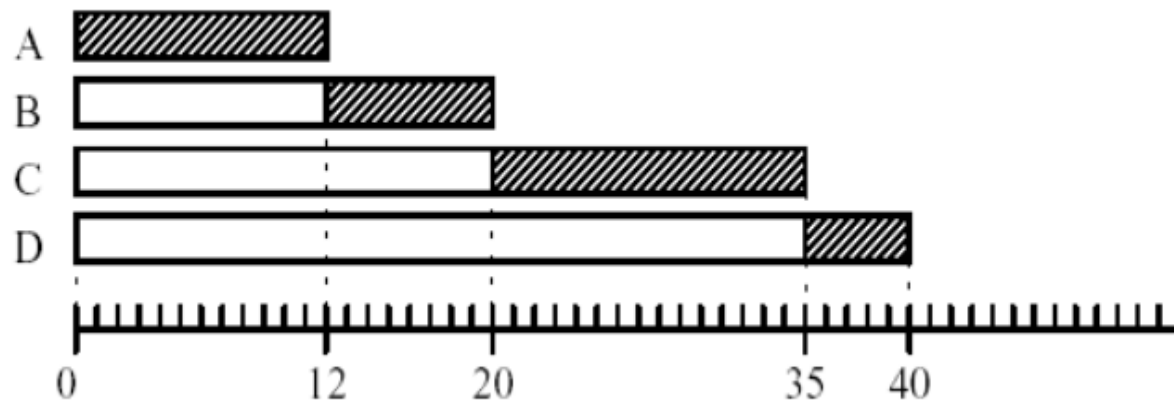
a. Ordem A-B-C-D = $(0 + 12 + 20 + 35) / 4 = 16.75$ u.t.

b. Ordem D-A-B-C = $(0 + 5 + 17 + 25) / 4 = 11.7$ u.t.



Algoritmo escalonamento FIFO

Processo	Tempo
A	12
B	8
C	15
D	5



Algoritmo escalonamento SJF

- No algoritmo anterior (FIFO) os processos que levam menos tempo podem ser prejudicados se ficarem atrás dos que levam mais tempo.
- Neste caso, o tempo médio que um processo espera para processar pode ser muito grande.



- O algoritmo pode ser modificado de forma que estes (os menores) sejam processados antes, e com isto, o tempo de espera médio diminui.
- O Shortest Job First ou algoritmo do “Menor Processo Primeiro” busca resolver este problema.



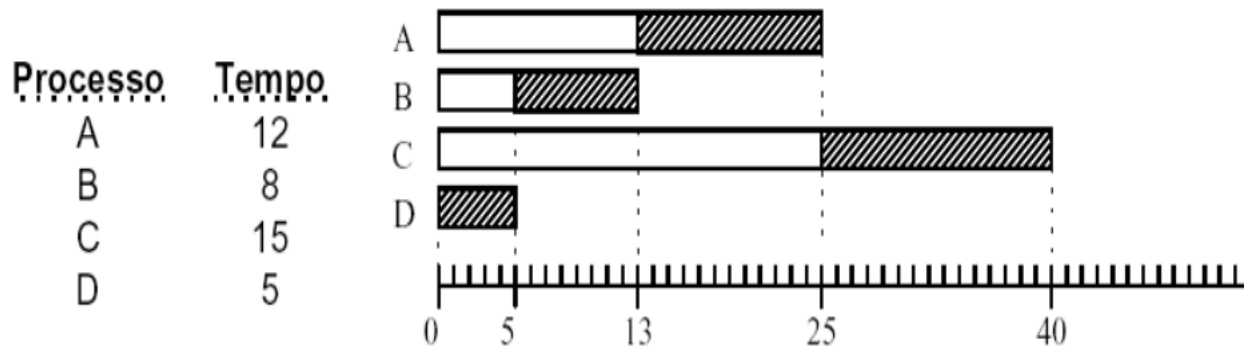
Algoritmo escalonamento SJF

- Para usar este algoritmo precisamos conhecer antecipadamente o tempo de execução de cada processo, o que é difícil.
- A ideia é alocar o processador para o menor job da fila.



Algoritmo escalonamento SJF

O fato é que o menor tempo médio é obtido quando se executa primeiro os processos de menor ciclo de processador (I/O bound).



$$\text{Tempo médio: } (0 + 5 + 13 + 25)/4 = 10.75 \text{ u.t}$$

Atividade

Faça uma pesquisa e descrição sobre:

Algoritmo escalonamento Cooperativo

Data de entrega no Moodle (16/11/2021)

