

Disciplina: Sistemas Operacionais
Curso: Análise e Desenvolvimento de Sistemas
Turno: Noite
Professor: Me. José Paulo Lima
Data de entrega: 18/06/2025
Aluno(a): Vinícius Henrique Costa Soares
Matrícula: 20232ADSGR0276

1. O que é **escalonamento**?

A escolha e execução do próximo processo a ser executado num SO, afetando diretamente na resposta dos mesmos ao usuário.

2. Quais as **condições** que fazem uma CPU escalonar?

Quando um processo é criado, terminado, ou quando ele bloqueia para E/S, ou quando ocorre uma interrupção de E/S (a resposta do trabalho feito, por exemplo).

3. Quais os **objetivos** do algoritmo de escalonamento?

No geral, dar uma porção justa da CPU à cada processo, verificar se a política estabelecida está sendo cumprida, e fazer com que todas as partes do sistema estejam trabalhando, evitando a ociosidade.

Porém em sistemas em lote, são atribuídas também as responsabilidades de maximizar o throughput (a vazão de tarefas), minimizar o tempo de retorno (entre a submissão e o término) e maximizar o uso da CPU.

Nos sistemas interativos, minimizar o tempo de resposta e ter boa proporcionalidade.

E nos de tempo real, evitar a perda de dados e evitar a degradação da qualidade em sistemas multimídia.

4. Qual a **diferença** entre escalonamento **preemptivo** e **não preemptivo**?

O preemptivo possui o conceito de "quantum" (um tempo máximo de execução fixo), que faz os processos serem suspensos se precisarem executar em mais tempo que o estabelecido, e o não-preemptivo executa o processo até bloquear pela E/S, até liberar a CPU ou se existe um processo de mais prioridade aguardando o clock.

5. Diferencie os termos abaixo que envolvem o escalonamento de processos:

(a) **Processos preemptivos** e **processos não-preemptivos**.

Processos baseados em tempo fixo, que liberam espaço para outros se não conseguirem executar nesse tempo, e em contraste, processos que executam até seu bloqueio, conclusão (usando totalmente da CPU até esse resultado) ou liberação para um processo de maior prioridade.

(b) Turnaround

Tempo completo da criação até o término do processo.

(c) Throughput

Quantidade de processos executados em determinado tempo.

(d) Tempo de espera

Tempo total de espera do estado de pronto até sua execução.

6. IF-SC - 2014 - IF-SC - Professor - Informática (Adaptada) Sobre algoritmos de escalonamento de processos:

A. Múltiplas Filas

B. Round-Robin

C. Shortest Job

D. Escalonamento Loterias

E. First Come First Served

(E) Nesse algoritmo o primeiro processo a chegar será o primeiro a ser executado.

(B) Nesse algoritmo é definido um *quantum* (fatia de tempo) para cada processo. Após encerrar a fatia de tempo, o processo escalonado deve ceder o lugar na CPU a outro.

(A) Nesse algoritmo são usadas várias filas de processos prontos para executar. Cada processo é colocado em uma fila e cada fila tem uma política de escalonamento.

(D) Nesse algoritmo o sistema distribui bilhetes aos processos, e faz um sorteio cada vez que precisa selecionar um processo para a CPU.

(C) Nesse algoritmo os processos menores terão prioridade, ou seja, serão executados primeiro.

7. Assinale V para as alternativas verdadeiras e F para as falsas:

A. V () F (X) Os sistemas interativos não fazem necessário a utilização de algoritmos de escalonamento preemptivos devido ao seu grande número de usuários.

B. V () F (X) O algoritmo de escalonamento First Come, First Served é flexível permitindo a podendo escolher qualquer processo da sua lista para executar na CPU.

C. V (X) F () No algoritmo de escalonamento First Come, First Served os processos podem executar quanto tempo quiser na CPU.


D. V (X) F () O algoritmo Shortest Job First permite que tarefas mais curtas possam ser executadas primeiramente antes das tarefas mais longas.


E. V (X) F () O algoritmo Remaining Time Next é similar ao algoritmo Short Job First, ou seja, ao receber um novo processo cujo o tempo de execução é maior do que o tempo restante do processo corrente, o algoritmo suspenderá o processo corrente o executará o novo processo.


8. Cinco processos, de A até E, chegam ao computador ao mesmo tempo. Eles têm seus tempos de processamento estimados em 10, 6, 2, 4 e 8 minutos respectivamente. Suas prioridades (atribuídas externamente) são 3, 5, 2, 1 e 4, respectivamente, sendo 5 o representante da prioridade mais alta. Nenhum dos processos faz I/O. Para cada um dos algoritmos de escalonamento abaixo, determine o tempo médio de *turnaround* dos processos. Ignore o *overhead* causado pela troca de contexto.

(a) Round Robin (fila começa em A, indo em ordem até E; *quantum* = 4)


$A = 10m, P(3); B = 6m, P(5); C = 2m, P(2); D = 4m, P(1); E = 8m, P(4);$


Execução 1: $A = 10 - Q \rightarrow 6m$;  $4m$


E2: $B = 6 - Q \rightarrow 2m$;  $8m$

E3: $C = 2 - Q/2 \rightarrow 0m$;  $10m - C$ ✓


E4: $D = 4 - Q \rightarrow 0m$;  $14m - D$ ✓

E5: $E = 8 - Q \rightarrow 4m$;  $18m$

E6: $A = 6 - Q \rightarrow 2m$;  $22m$

E7: $B = 2 - Q/2 \rightarrow 0m$;  $24m - B$ ✓

E8: $E = 4 - Q \rightarrow 0m$;  $28m - E$ ✓

E8: $A = 2 - Q/2 \rightarrow 0m$;  $30m - A$ ✓

Turnaround: $A = 30m; B = 24m; C = 10m; D = 14m; E = 28m.$

Média de Turnaround: **21,2 minutos.**

(b) Escalonamento com prioridade.

$A = 10m, P(3); B = 6m, P(5); C = 2m, P(2); D = 4m, P(1); E = 8m, P(4);$

E1: B 👉 6m

E2: E 👉 14m

E3: A 👉 24m

E4: C 👉 26m

E5: D 👉 30m

Turnaround: $A = 24m; B = 6m; C = 26m; D = 30m; E = 14m.$

Média de Turnaround: **20 minutos.**

(c) FIFO (ordem de execução: A, B, C, D, E)

$A = 10m, P(3); B = 6m, P(5); C = 2m, P(2); D = 4m, P(1); E = 8m, P(4);$

E1: A 👉 10m

E2: B 👉 16m

E3: C 👉 18m

E4: D 👉 22m

E5: E 👉 30m

Turnaround: $A = 10m; B = 16m; C = 18m; D = 22m; E = 30m.$

Média de Turnaround: **19,2 minutos.**

(d) SJF

$A = 10m, P(3); B = 6m, P(5); C = 2m, P(2); D = 4m, P(1); E = 8m, P(4);$

E1: C 👉 2m

E2: D 👉 6m

E3: B 👉 12m

E4: E 👉 20m

E5: A 👉 30m

Turnaround: $A = 30m; B = 12m; C = 2m; D = 6m; E = 20m.$

Média de Turnaround: **14 minutos.**

9. Considere o seguinte conjunto de processos, com o tamanho do tempo de *burst* de CPU dado em milissegundos:

| Processo | Tempo de Serviço | Prioridade |

| P1 | 10 | 3 |

| P2 | 1 | 1 |

| P3 | 2 | 3 |

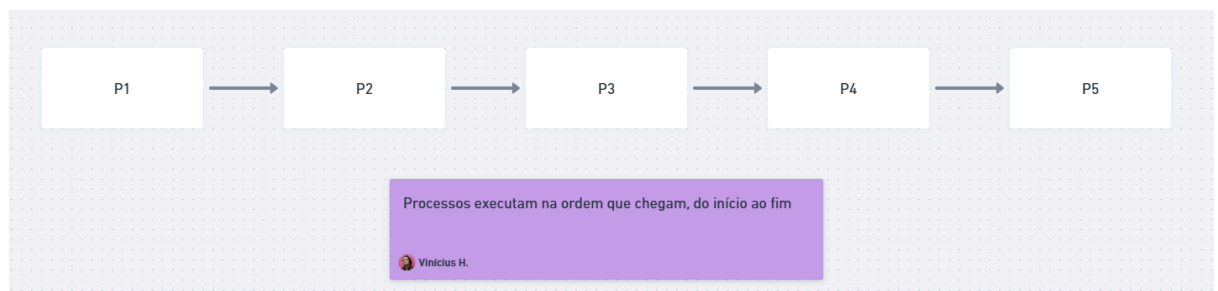
| P4 | 1 | 4 |

| P5 | 5 | 2 |

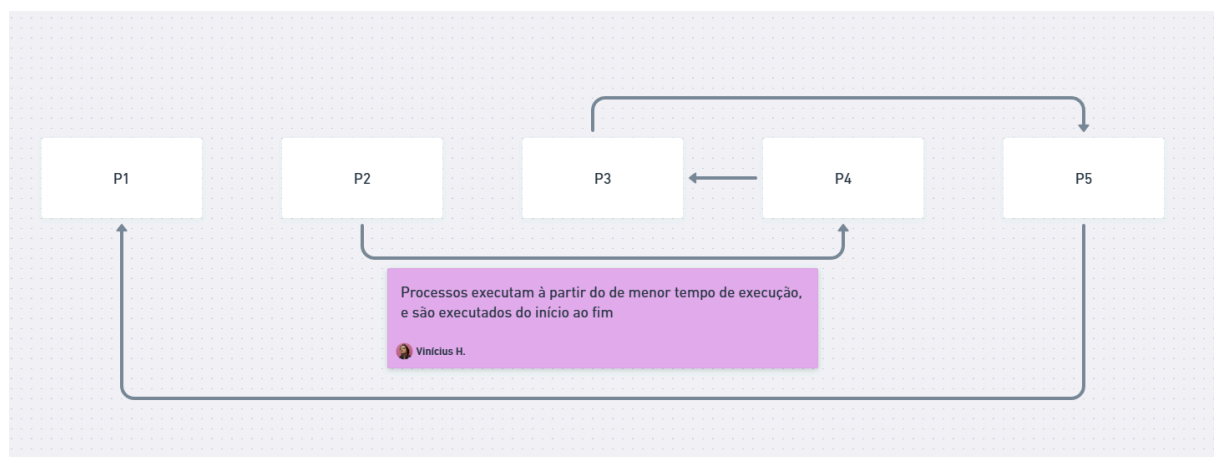
Considere que os processos chegaram na ordem P1, P2, P3, P4, P5 no momento 0.

(a) Desenhe quatro gráficos que ilustrem a execução desses processos usando FCFS, SJF, prioridade não-preemptiva (um número de prioridade menor significa uma prioridade mais alta) e o escalonamento Round Robin (*quantum* = 1).

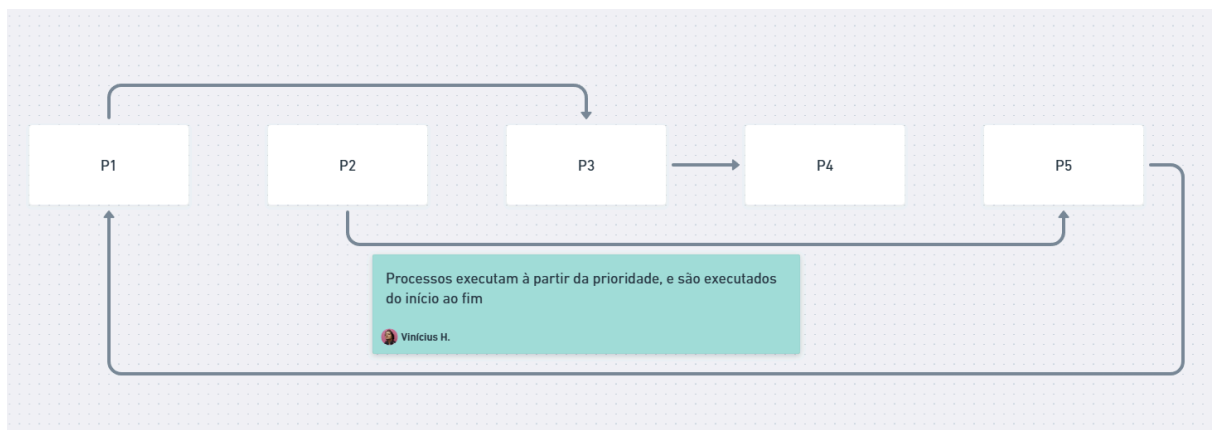
First Come, First Served:



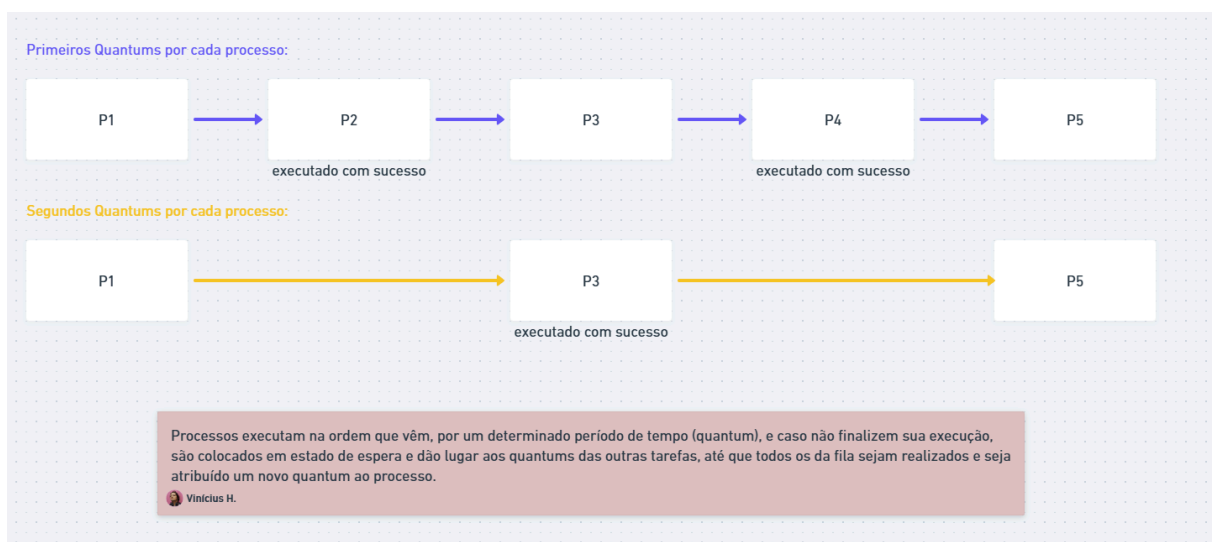
Shortest Job First:



Prioridade não-preemptiva:



Round Robin (Q=1):



(b) Qual é o *turnaround* de cada processo para cada um dos algoritmos de escalonamento no item (a)?

First Come, First Served: $P1 = 10ms$; $P2 = 11ms$; $P3 = 13ms$; $P4 = 14ms$; $P5 = 19ms$.

Shortest Job First: $P1 = 1ms$; $P2 = 2ms$; $P3 = 4ms$; $P4 = 9ms$; $P5 = 19ms$.

Prioridade não-preemptiva: $P1 = 1ms$; $P2 = 6ms$; $P3 = 16ms$; $P4 = 18ms$; $P5 = 19ms$.

Round Robin (Q=1): $P1 = 19ms$; $P2 = 2ms$; $P3 = 7ms$; $P4 = 4ms$; $P5 = 14ms$.

(c) Qual é o tempo de espera de cada processo para cada um dos algoritmos de escalonamento no item (a)?

First Come, First Served: $P1 = 0ms$; $P2 = 10ms$; $P3 = 11ms$; $P4 = 13ms$; $P5 = 14ms$.

Shortest Job First: $P1 = 0ms$; $P2 = 1ms$; $P3 = 2ms$; $P4 = 4ms$; $P5 = 9ms$.

Prioridade não-preemptiva: $P1 = 0ms$; $P2 = 1ms$; $P3 = 6ms$; $P4 = 16ms$; $P5 = 18ms$.

Round Robin ($Q=1$): $P1 = 9ms$; $P2 = 1ms$; $P3 = 5ms$; $P4 = 3ms$; $P5 = 9ms$.

(d) Qual dos escalonamentos no item (a) resulta no menor tempo de espera médio (em relação a todos os processos)?

O Shortest Job First.

- 10. A tabela abaixo apresenta alguns processos a serem organizados por um algoritmo de escalonamento. Considerando que cada processo tem seu respectivo tempo de criação/chegada, tempo de execução e prioridade, organize estes processos de acordo com os algoritmos não preemptivos abaixo:**

| Processo | Chegada | Execução | Prioridade |

| A | 3 | 6 | 1 |

| B | 4 | 7 | 3 |

| C | 1 | 3 | 2 |

| D | 8 | 8 | 6 |

| E | 7 | 2 | 4 |

| F | 5 | 4 | 7 |

| G | 2 | 9 | 8 |

| H | 0 | 5 | 5 |

| I | 10 | 1 | 9 |

(a) FIFO (First In First Out) ou FCFS (First Come First Served)

HCGABFEDI

(b) SJF (Shortest Job First)

HCEFABDGI

(c) Por prioridades

HACBEDFGI

(d) Round-Robin com *quantum* igual 3.

HCGABFEDI

(e) Represente detalhadamente como ficaria a ordem de execução dos algoritmos SJF e por Prioridades caso eles fossem preemptivos.

(f) Calcule o tempo de espera de cada processo em cada algoritmo e indique qual algoritmo teve em média menores tempo de espera.

O Shortest Job First apresentou uma média de 10.78 de resposta, quase 7 segundos de diferença para o First Come, First Served.

11. Cinco processos são criados na seguinte ordem: P1, P2, P3, P4 e P5, com os seguintes tempos:

| Processo | Tempo de serviço | Prioridade | Tempo de chegada |

| P1 | 13 | 3 | 0 |

| P2 | 11 | 4 | 4 |

| P3 | 7 | 1 | 5 |

| P4 | 8 | 2 | 7 |

| P5 | 16 | 5 | 10 |

Ilustre a execução dos processos através de um diagrama usando os seguintes esquemas de escalonamento:

(a) FIFO

(b) SJF

(c) Prioridade (número de prioridade menor implica prioridade maior)

(d) Circular com fatia de tempo = 4 u.t.

(e) Mostre os tempos de *turnaround* individuais por processo e a média resultante.

Desconsidere E/S ou tempo de escalonamento ou troca de contexto entre processos.