Curso de

Sistemas Operacionais

Escalonamento de Processos

Prof. Robson Siscoutto

e-mail: robson@unoeste.br

Sistemas Operacionais
Escalonamento de Processos

- Introdução;
- Modelos do processo;
- Estados e mudanças do processo;
- Tipos de processos;
- Threads
- Comunicação entre processos;
- **Escalonamento de Processos;**
- Concorrência entre Processos
  - Problemas de sincronização;
  - Soluções de hardware e software;
- Deadlock
- Threads

Escalonamento de Processos

Escalonamento de Processos - Objetivos

- **Todos os Sistemas**
  - Justiça - dar a cada processo um porção justa da CPU;
  - Aplicação da Política – verificar se a política estabelecida é cumprida;
  - Equilíbrio - manter ocupada todas as partes do sistema

- **Sistema em Lote**
  - Vazão *(throughput) – maximizar o numero de jobs/hora*
  - Tempo de retorno – minimizar o tempo entre a submissão e o término
  - Utilização da CPU – manter a CPU ocupada todo o tempo;

Escalonamento de Processos

Objetivos

- **Sistemas Interativos**
  - Tempo de resposta – responder rapidamente as requisições
  - Proporcionalidade – satisfazer as expectativas dos usuários

- **Sistema Tempo real**
  - Cumprimento dos Prazos – evitar perdas dos dados
  - Previsibilidade – evitar a degradação da qualidade em sistemas multimídia

Escalonamento de Processos

Tipos

- Dois tipos de Escalonamentos:
  - Escalonamento **Não-preemptivo**
    - (Sistema em Lote);
  - Escalonamento **Preemptivo**
    - (Sistema Interativo);

Escalonamento de Processos

Escalonamento Não-preemptivo

ou

Escalonamento em Sistemas em Lote

Escalonamento de Processos
Escalonamento Não-Preemptivo

- Escalonamento Não-preemptivo:
  - Primeiros sistemas multiprogramáveis
    - (processamento Batch);
  - Processador dedicado a um processo por vez;
    - Não existe mudança de contexto;

  - Exemplos de escalonamento:
    - FIFO (First-in First-out) ou FCFS (First Come, First Served)
    - SJF (Shortest Job First) – Job mais curto primeiro

Escalonamento de Processos
Escalonamento Não-Preemptivo

- Escalonamento Não-preemptivo: FIFO (First-in First-out)

  - Processo que chegar primeiro é o primeiro a ser executado;
  - Utiliza-se apenas uma fila, onde os processos no estado de pronto são inseridos no final;
  - Os processos no inicio da fila são escalonados primeiro;

  - Problema:
    - **Impossível prever** quando um processo terá sua execução iniciada, devido a função de tempo de execução dos processos que se encontram na sua frente;

First-in First-out – FIFO ou FCFS

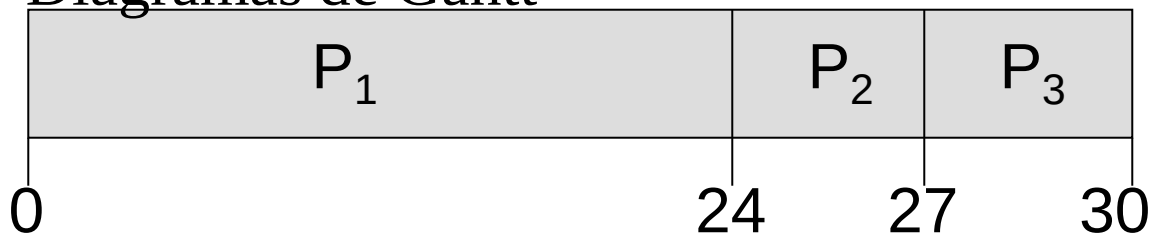
Processo	Tempo Chegada
P_1	24
P_2	3
P_3	3

- Suponha que os esses processos cheguem na seguinte ordem:
  - P_1 , P_2 , P_3

Tempo de Espera.
Tempo que o processo permanece na fila de prontos.
É a soma dos períodos utilizados pelo processo no estado de Pronto.

First-in First-out - FIFO (2)

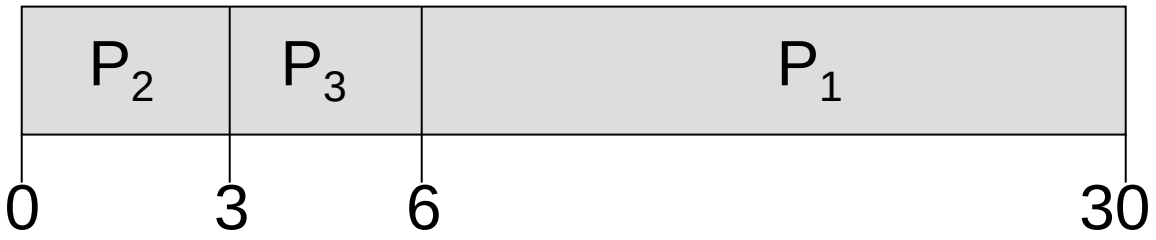
Diagramas de Gantt



- Tempo de Espera para cada processo:
  - *Tempo Espera*: $P_1 = 0$; $P_2 = 24$; $P_3 = 27$

- Tempo Médio de Espera:
  - *Tempo médio de Espera*: (0 + 24 + 27)/3 = 17

First-in First-out - FIFO (3)



- Suponha que os mesmos processos cheguem agora na seguinte ordem:
  - P_2 , P_3 , P_1
- Tempo de espera de cada processo:
  - *Tempo Espera:* $P_1 = 6$; $P_2 = 0$; $P_3 = 3$
- Tempo médio de espera:
  - *Tempo Médio de Espera*: (6 + 0 + 3)/3 = 3

Escalonamento de Processos
Escalonamento Não-Preemptivo

- Escalonamento Não-preemptivo: **SJF (Shortest Job First)**
  - Associa a cada processo o seu **tempo de execução**;
  - Se o processador estiver livre, o processo que **tiver o menor tempo de execução é selecionado** para executar;
  - Este tipo de escalonamento **favorece** os **processos pequenos** com tempos curtos;
  - **Problema**:
    - **Determinar, exatamente, quanto tempo de CPU** cada processo necessita para terminar seu processamento;

Shortest Job First - SJF (1)

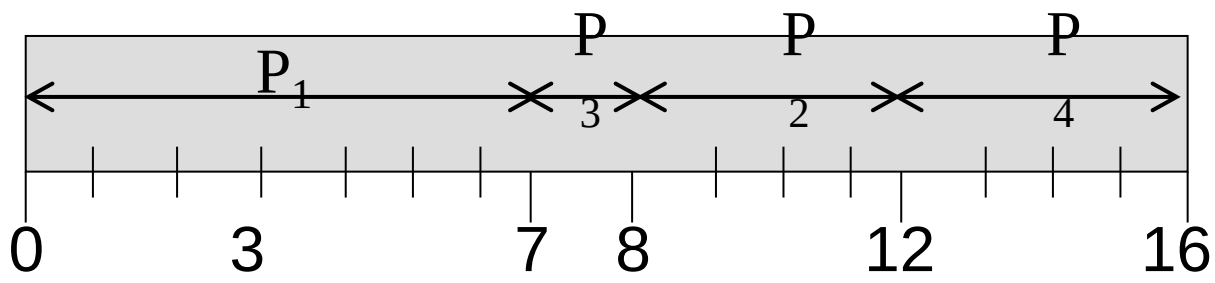
- Abordagem 1:
  - Processo com menor expectativa de tempo de processamento é selecionado para execução.
- Abordagem 2:
  - Associado com cada processo está o tamanho do seu próximo tempo de *CPU (cpu burst)*.
  - Esse tamanho é usado como critério de escalonamento, sendo selecionado o processo de menor próximo tempo de *CPU (cpu burst)*.

Shortest Job First - SJF (2)

- Dois esquemas:
  - **Não-preemptivo** – uma vez a CPU alocada a um processo ela não pode ser dada a um outro antes do término do tempo de CPU corrente.
  - **Preemptivo** – se chega um novo processo com tempo de CPU **menor que o tempo remanescente** do processo corrente ocorre a preempção. Esse esquema é conhecido como *Shortest-Remaining-Time-First (SRTF)*.

Exemplo de SJF Não-Preemptivo

Processo	Tempo Chegada	Time de CPU
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4



Escolhe P1 por ter chegado no tempo 0.0. Quando P1 terminar todos os outros já chegaram, ai escolhe o de menor tempo. (Inicio-Chegada)

8 - 2

- *Tempo Médio de Espera* = (0 + 6 + 3 + 7)/4 = 4
  P1 P2 P3 P4

Escalonamento de Processos
Escalonamento Preemptivo

Escalonamento Preemptivo

ou

Escalonamento em Sistemas Interativos

Escalonamento de Processos
Escalonamento Preemptivo

- **Escalonamento Preemptivo**:
  - Quando o sistema pode **interromper um processo** em execução, para que outro processo utilize o processador;
  - Este tipo de escalonamento é utilizado em sistemas de tempo compartilhado;
  - Permite **tratar processos de maior prioridades**;

Escalonamento de Processos
Escalonamento Preemptivo

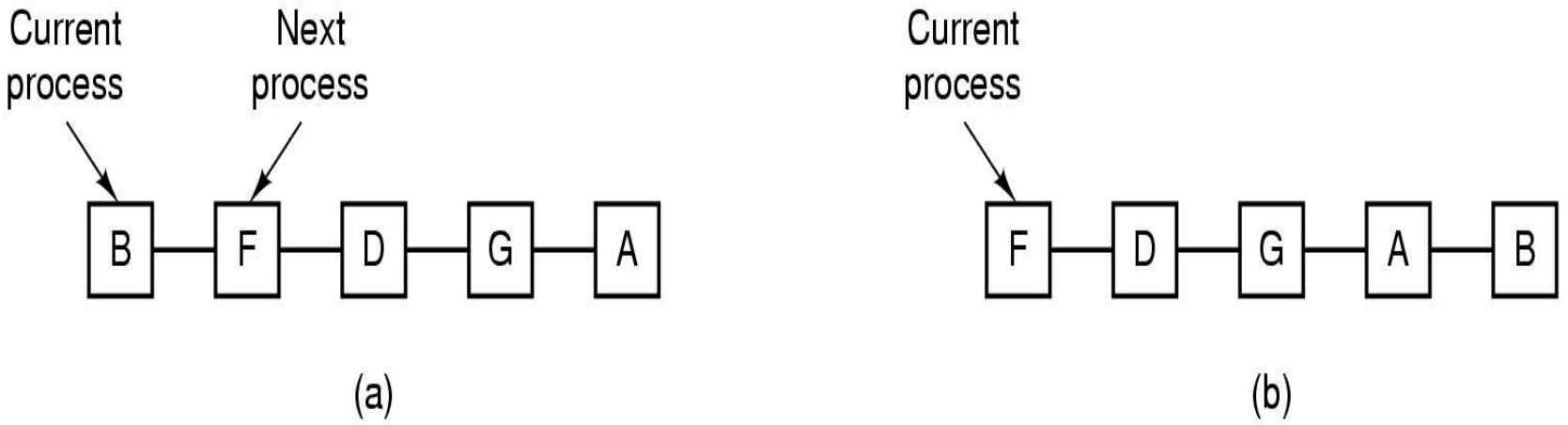
- Escalonamento Preemptivo – Exemplos de Políticas:
  - Circular (Round Robin);
  - por Prioridades;
  - Próximo Processo mais Curto
  - Garantido
  - Por Loteria
  - Por Fração Justa (fair-share)

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Escalonamento Circular**
  - Utiliza o conceito de Fatia de tempo de processador;
  - O processo no **inicio da fila é escalonado primeiro**;
  - Quando ocorre a preempção o processo é retirado da CPU e colocado no final da fila;
  - A **fila de processo no estado de prontos** é tratada como uma **fila circular**;

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Escalonamento Circular**



  - Lista de processos prontos
  - Lista de processos prontos depois de B usar sua fatia de tempo

Escalonamento de Processos
Escalonamento Preemptivo

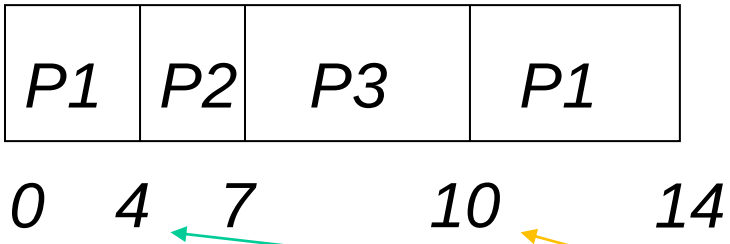
- Escalonamento Preemptivo: **Escalonamento Circular**
  - Quantum ideal:
    - Quantum **muito curto**:
      - Causa muita alternância de processo e reduz a eficiência da CPU
      - gera muito overhead devido às trocas de contexto
    - Quantum **muito longo**
      - Pode gerar uma resposta pobre às requisições interativas curtas;
      - tende a FIFO
    - Quantum razoável: em torno de 10 a 100 ms.

Escalonamento Round Robin (1)

Processo	Tempo de CPU	Tempo de Fatia - Quantum
P_1	14	4
P_2	3	
P_3	3	

Não considerar
Tempo de chegada

Cuidado: processos que rodam mais do uma vez desconsiderar tempos intermediários que ficou parado



Tempo Médio de Espera = ((10-4)+4+7)/3 = 17/3=5,6ms
P1 P2 P3

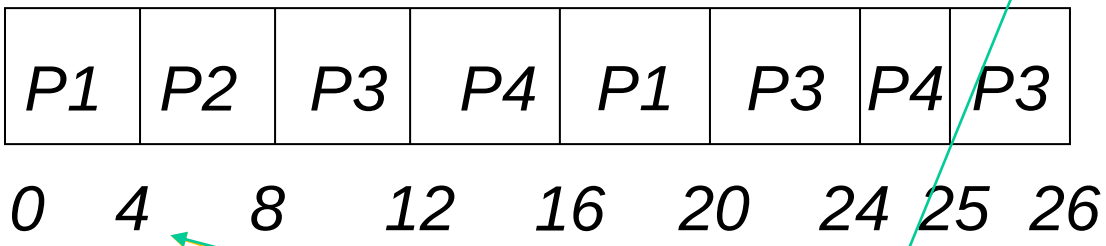
Tempo Médio de Retorno = (14 + 7+10)/3 =31/3=10,3ms

Tempo de retorno (**Turnaround time**).
Tempo transcorrido desde que se lança um processo (entra na fila de prontos) até que finalize sua execução.
É a soma do tempo de espera para ir para a memória, tempo de espera na fila dos prontos, tempo em execução na CPU e o tempo de espera por recursos.

Escalonamento Round Robin (1)

Processo	Tempo de CPU	Chegada	Quantum
P_1	8	0	4
P_2	4	1	
P_3	9	2	
P_4	5	3	

Considerar Tempo de chegada



Tempo Médio de Espera = ((16-4)+(4-1)+(8-2+20-12+25-24)
+(12-3+24-16))/4 = 47/4=11,75ms

Tempo Médio de Retorno = ((20-0)+(8-1)+(26-2)+(25-3))/4 =73/4=18,25ms

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Próximo Processo mais Curto (SPN)**
  - Baseado no SJF
  - Realiza uma estimativa com base no comportamento passado;
    - Menor tempo executa
  - Soma ponderada – tb conhecido como **aging**
    - $\alpha\ T_0 + (1- \alpha)T_1$
  - Exemplo: $\alpha = ½$
    - T0, t0/2 + T1/2, T0/4 + T1/4 .....

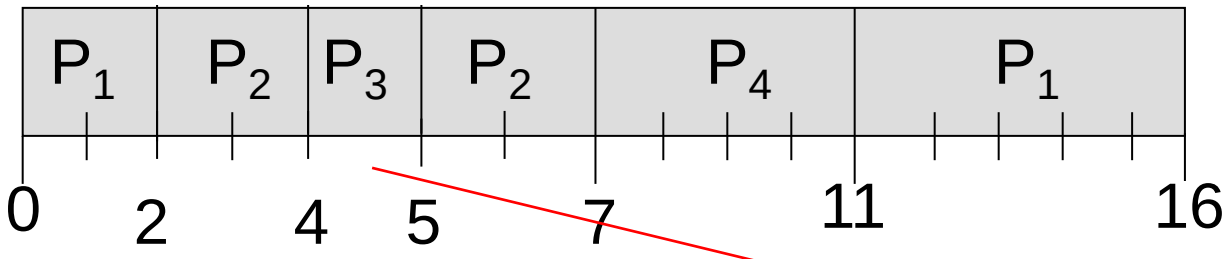
Escalonamento de Processos
Escalonamento Preemptivo

- O parâmetro α controla o peso relativo do histórico recente e passado na fórmula.
  - Se $\alpha = 0$
    - $\tau_{n+1} = \tau_n$
    - Histórico recente não é considerado relevante.
  - Se $\alpha = 1$
    - $\tau_{n+1} = t_n$
    - Apenas o último tempo de CPU é levado em conta.

Escolhe P1 por ter chegado no tempo 0.0. Se chegar outro processo com tempo menor do que o tempo restante daquele que esta executando, este é trocado.
CONSIDERAR O MENOR TEMPO CPU PARA TROCAR DE PROCESSO

Exemplo de SJF Preemptivo (Algoritmo SRTF)

Processo	Tempo Chegada	Time de CPU
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4



P1 = (11 – 2 - 0) = 9 (INICIO 0 E CHEGADA 0)

P2 = (5 – 4-0) = 1 (INICIO 2 E CHEGADA 2)

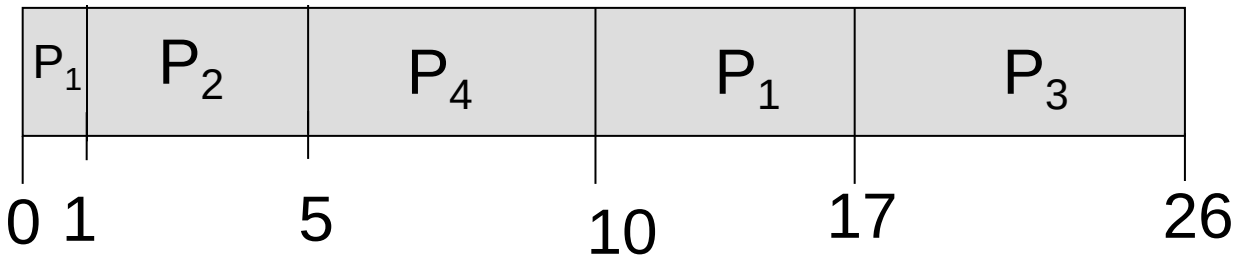
P3 = (4 - 4) = 0 (INICIO 4 E CHEGADA 4)

P4 = (7 – 5) = 2 (INICIO 7 E CHEGADA 5)

- *Tempo Médio de Espera* = (9 + 1 + 0 + 2)/4 = 3
  P1 P2 P3 P4

Exemplo de SJF Preemptivo (Algoritmo SRTF)

Processo	Tempo Chegada	Time de CPU
P_1	0.0	8
P_2	1.0	4
P_3	2.0	9
P_4	3.0	5

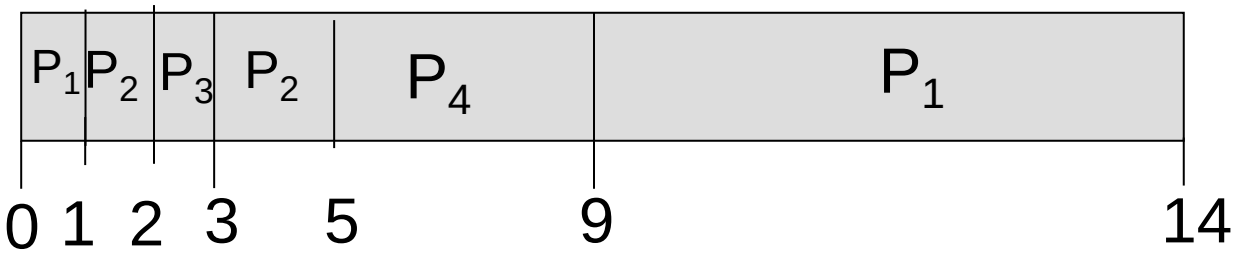


- *Tempo Médio de Espera* = ((10-1) + (1-1) + (17-2) + (5-3))/4 = 26/4 = 6,5

P1 P2 P3 P4

Exemplo de SJF Preemptivo (Algoritmo SRTF)

Processo	Tempo Chegada	Time de CPU
P_1	0.0	6
P_2	1.0	3
P_3	2.0	1
P_4	3.0	4



- *Tempo Médio de Espera* = ((9-1) + (3-1) + (2-1) + (5-3))/4 = 13/4 = 3,2
  P1 P2 P3 P4
- *Tempo de Retorno* = ((14-0) + (5-1) + (3-2) + (9-3))/4 = 25/4 = 6,2

Tempo de Retorno (Turnaround): tempo total – tempo chegada

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Escalonamento por Prioridades**
  - A cada processo é associado uma prioridade;
  - Os processos de maior prioridade são escalonados primeiro;
  - Um processo de menor prioridade deve ceder a CPU para um de maior prioridade;

  - Problema: "*starvation*"
    - Processos de baixa prioridade podem nunca executar

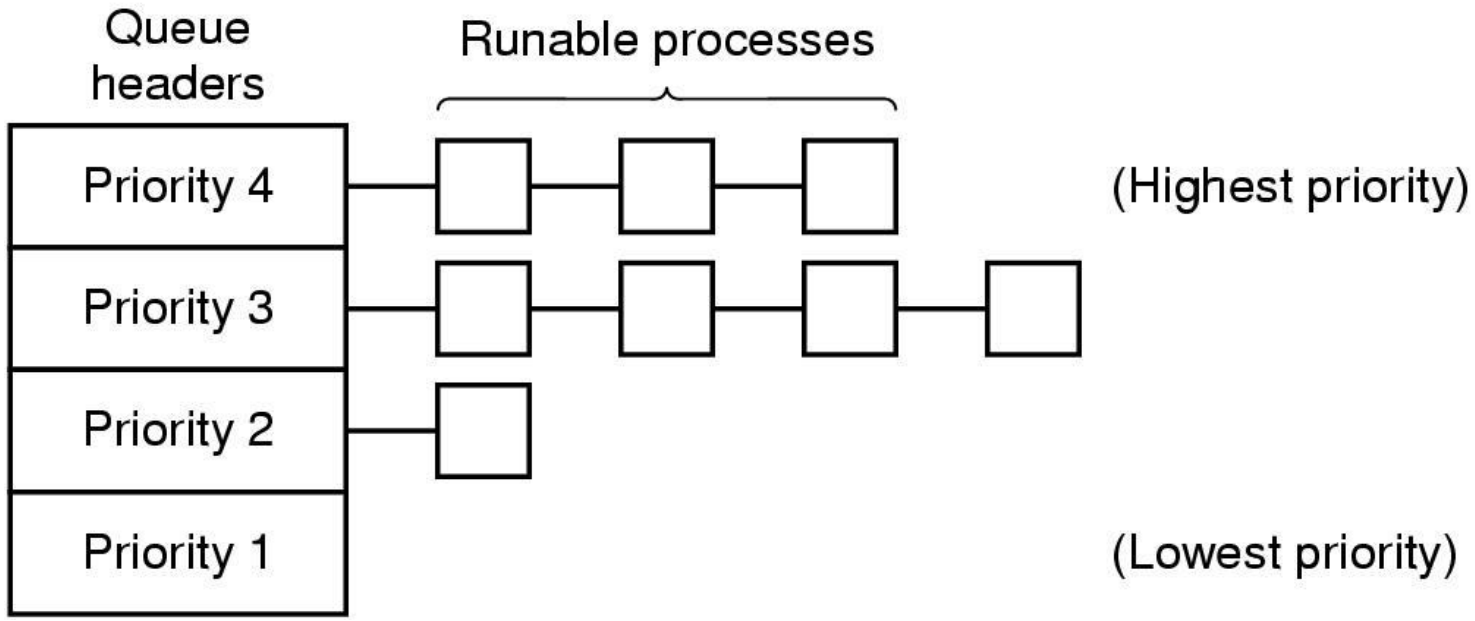
  - Solução: "*Aging*"
    - Prioridade aumenta com o passar do tempo.

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: Escalonamento por Prioridades
  - A prioridade pode ser atribuída a processos de forma:
    - **Estática**:
      - Não é modificada durante a existência do processo;
    - **Dinâmica**:
      - Ajustada de acordo com o tipo de processamento realizado pelo processo e/ou carga do sistema;
      - Pelo usuário: comando nice do UNIX (diminui a prioridade);
      - Pelo sistema, levando em consideração por exemplo:
        - Operações de E/S,

Escalonamento de Processos
Escalonamento Preemptivo

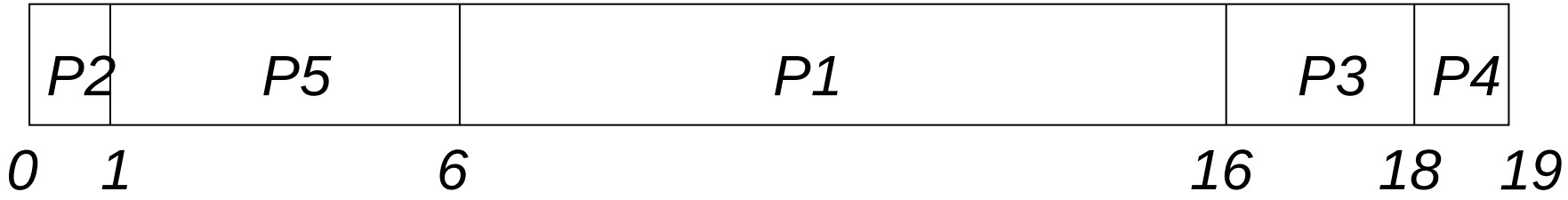
- Escalonamento Preemptivo: Escalonamento por Prioridades



Um algoritmo de escalonamento com 4 classes de prioridades;

Escalonamento por Prioridade (1)

Processo	Tempo de CPU	Prioridade
P_1	10	3
P_2	1	1 - maior
P_3	2	4
P_4	1	5 - menor
P_5	5	2

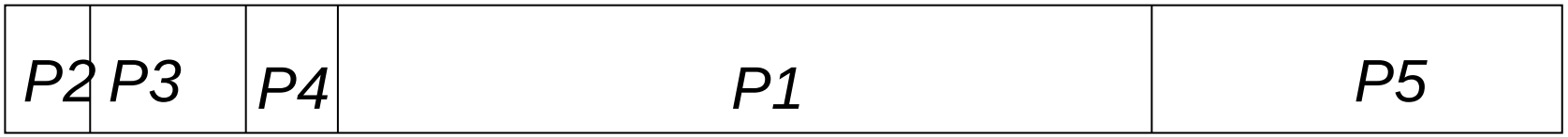


Tempo Médio de Espera = (6+0+16+18+1)/5 = 8,2ms

P1 P2 P3 P4 P5

Escalonamento por Prioridade (2)

Processo		Tempo de CPU	Prioridade
P_1	background	10	1
P_2	Interativo	1	0
P_3	Interativo	2	0
P_4	Interativo	1	0
P_5	background	5	1



0 1 3 4 14 19

Tempo Médio de Espera = (4+0+1+3+14)/5 = 4,4ms

P1 P2 P3 P4 P5

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Garantido**
  - Se hover n usuários conectados, cada um receberá 1/n de CPU.
  - O sistema deve manter o controle da quantidade de CPU que cada processo recebe desde a sua criação:
    - Calcula a taxa entre o tempo consumido e o tempo realmente destinado ao processo;
    - Baseado na diferença decide se vai ou nãoexecutar o processo mais vezes que outros para se igualar ao seu tempo real;

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Por Loteria**
  - A cada processo é dado bilhetes de loteria
  - A CPU faz sorteio e quem tiver o bilhete ganha direito a CPU;
  - Novos processos vão sempre concorrer ao sorteio;
  - Processo parceiros podem trocar bilhetes;

Escalonamento de Processos
Escalonamento Preemptivo

- Escalonamento Preemptivo: **Por Fração Justa (fair-share)**
  - **Evitar que mais processo de 1 usuário obtenha mais CPU que 1 processo de 1 usuário;**
  - **Para cada usuário (não processo) é alocado uma fração da CPU e o escalonador escolhe os processos de forma a garantir essa fração;**

  - **Exemplo: usuário 1 – processos A, B, C e D**

    usuário 2 – processo E

Solução com fila circular:

A E B E C E D E A E B E C E D E

Escalonamento em Tempo Real

Escalonamento de Processos
Escalonamento em Tempo Real

- **Escalonamento em Tempo Real**

Sistema de Tempo Real escalonável:

- Dado
  - *m* eventos periódicos
  - Evento *i* ocorre com periodo P_i e requer C_i segundos.
- Então, carrega-o se:

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

Escalonamento de Processos
Escalonamento em Tempo Real

- **Escalonamento em Tempo Real**
  - Algoritmos de Escolonamento em tempo real:
    - Estáticos
      - Toman decisões de escalonamento antes do sistema começar a executar;
      - Só funciona se há prévia informação perfeita disponível sobre o trabalho necessário a ser feito e os prazos que devem ser cumpridos;

Escalonamento de Processos
Escalonamento em Tempo Real

- **Escalonamento em Tempo Real**
  - Algoritmos de Escolonamento em tempo real:
    - Dinâmicos
      - Tomam decisões em tempo de execução;
      - Não apresentam restrições, pois a informações são adquiridas durante a execução do processo.

Escalonamento em Tempo Real (1)

Processo	Perídos de Eventos	Tempo de CPU
P_1	100, 200 e 500	50, 30 e 100

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1 = (50/100)+(30/200)+(100/500) = 0{,}5+0{,}15+0{,}2 \leq 1$$

Exemplo (2) = Acrescentar um 4º evento com 1 seg (1000 ms).

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1 = (50/100)+(30/200)+(100/500)+(x/1000) \leq 1$$

$(x/1000) \leq 1 - 0{,}85$

$x \leq 0{,}15 * 1000$

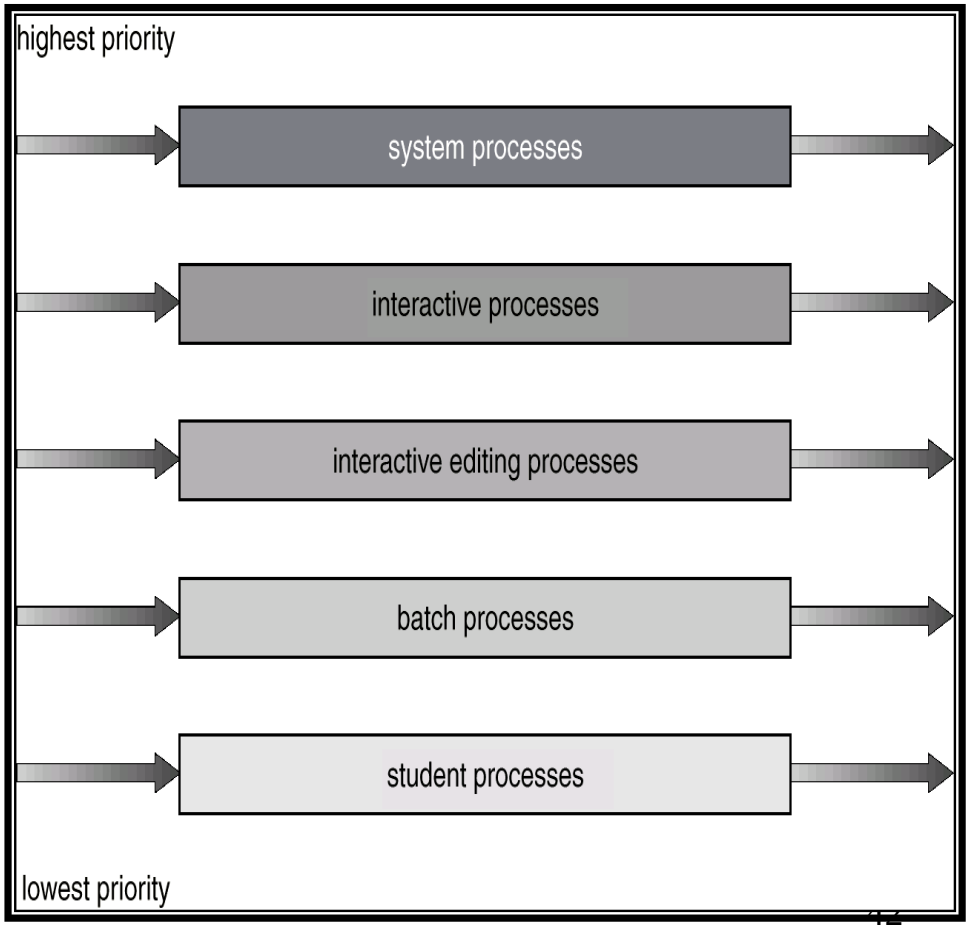
$x \leq 150$ ms

Escalonamento de Processos

Escalonamento de Multinível

Escalonamento Multinível (1)

- A idéia base é dividir os processos em diferentes grupos, com diferentes requisitos de tempos de resposta.
- A cada grupo é associada uma fila de prioridade, conforme a sua importância .



Escalonamento Multinível (2)

- Por exemplo, a fila de prontos pode dividida em duas filas separadas:
  - *foreground* (p/ processos interativos)
  - *background* (p/ processamento *batch*)

- Cada fila apresenta o seu próprio algoritmo de escalonamento:
  - *foreground* – RR
  - *background* – FCFS

Escalonamento Multinível (3)

- Escalonamento deve ser feito entre as filas:
  - Prioridades fixas – atende primeiro aos processos da fila *foreground* e somente depois aos da fila *background*.
  - Time slice – cada fila recebe uma quantidade de tempo de CPU para escalonamento entre os seus processos. Ex: 80% para *foreground* em RR e 20% para background em FCFS.

Escalonamento Multinível (4)

Escalonamento Multinível com Feedback

- O processo pode se mover entre as várias filas. Deste modo, a estratégia de *aging* pode ser implementada.
- O escalonador trabalha com base nos seguintes parâmetros:
  - Número de filas;
  - Algoritmo de escalonamento de cada fila;
  - Método usado para determinar quando aumentar e quando reduzir a prioridade do processo;
  - Método usado para se determinar em que fila o processo será inserido.

Escalonamento Multinível (5)

- Suponha a existência de 3 filas:
  - Q_0 – time quantum 8 milliseconds
  - Q_1 – time quantum 16 milliseconds
  - Q_2 – FCFS
- Escalonamento:
  - Um job novo entra na fila Q_0, que é servida segundo a estratégia RR. Quando ele ganha a CPU ele recebe 8 ms. Se não terminar em 8 ms, o job é movido para a fila Q_1.
  - Em Q_1 o job é novamente servido RR e recebe 16 ms adicionais. Se ainda não completar, ele é interrompido e movido para a fila Q_2.
  - Em Q2, FCFS

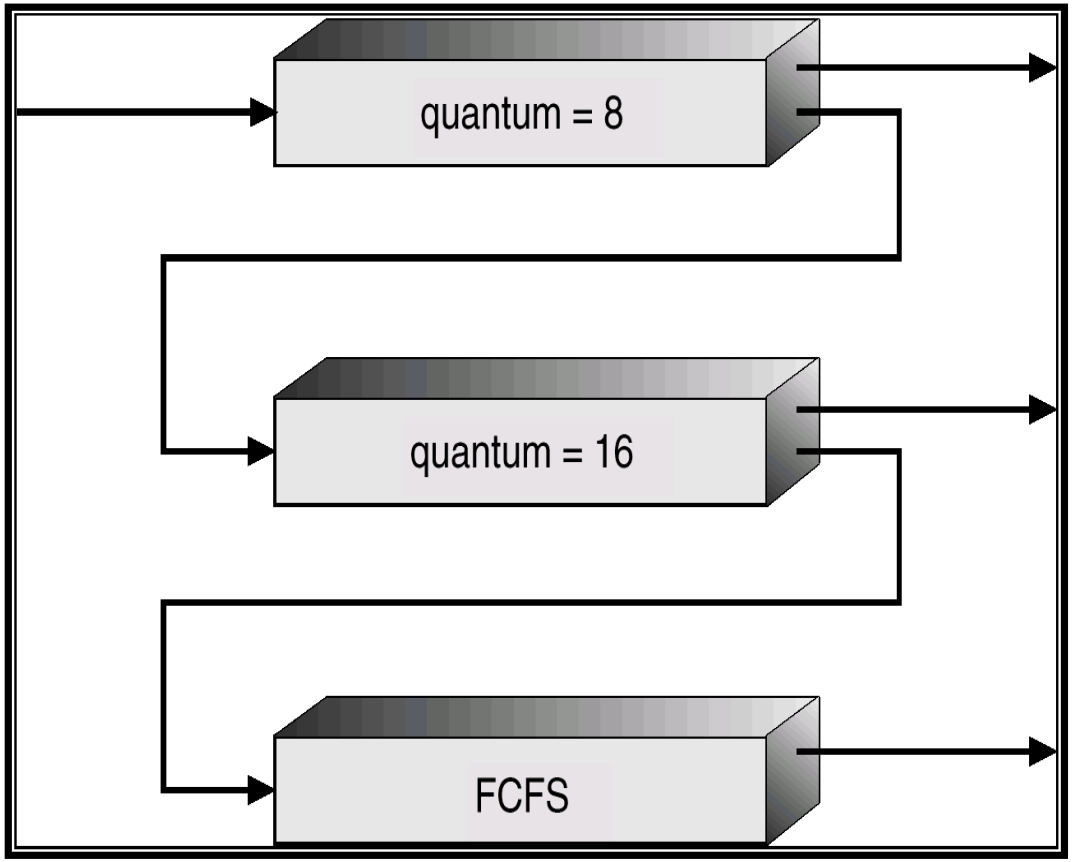
Exemplo (1)

Escalonamento de Processos

Escalonamento de Multinível

Escalonamento Multinível (6)

Exemplo (2)



Escalonamento de Processos

- **Referências Utilizadas:**
  - Livro do Tanenbaum
    - Sistemas Operacionais Modernos
    - **www.cs.vu.nl/~ast**
  - Livro do Silberschatz
    - Operating System Concepts
    - **www.bell-labs.com/topic/books/aos-book/**
  - Livro do Machado e Maia
    - Arquitetura de Sistemas Operacionais.