INSTITUTO FEDERAL
Rio Grande do Sul

Sistemas Operacionais

Maria Helena Schneid Vasconcelos
maria.vasconcelos@sertao.ifrs.edu.br

16/11/2021

Conteúdo Programático

1 - Histórico de Sistemas Operacionais
2 - Tipos de Sistemas Operacionais e suas características.
3 - Gerência de processador.

1ª Etapa

4 - Gerência de memória.
5- Gerência de arquivos.
6 - Gerência de entrada e saída.
7 – Estudo de Caso.

2ª Etapa

Algoritmos de escalonamento

Alguns algoritmos de escalonamento:

- Algoritmos **não preemptivos:**

1. **FIFO**
2. **SJF**
3. **Cooperativo**

Algoritmo escalonamento Cooperativo

Nos algoritmos anteriores, os processos não podiam ser preemptados por tempo máximo de processamento (timesharing).

Com a evolução, os computadores passaram a ser multi-usuário ou multitarefa. Neste caso, uma maior interação com o usuário e uma melhor utilização da CPU pelos programas tornou-se necessária.

O método cooperativo é uma forma de tornar isso possível.

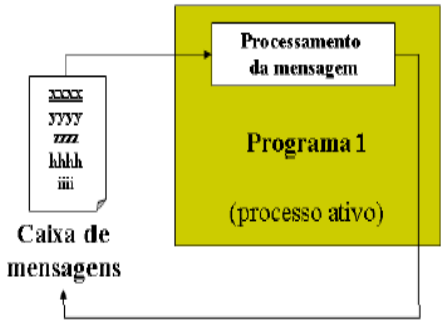
Neste caso, os processos cooperam uns com os outros, passando a CPU a um outro caso este seja necessário.

Algoritmo escalonamento Cooperativo

Uma forma de implementar este modelo é criando uma "fila ou caixa de mensagens", onde informações sobre os eventos ocorridos são postadas.

Os processos verificam continuamente a caixa de mensagens, retirando e processando as que são endereçadas a eles.

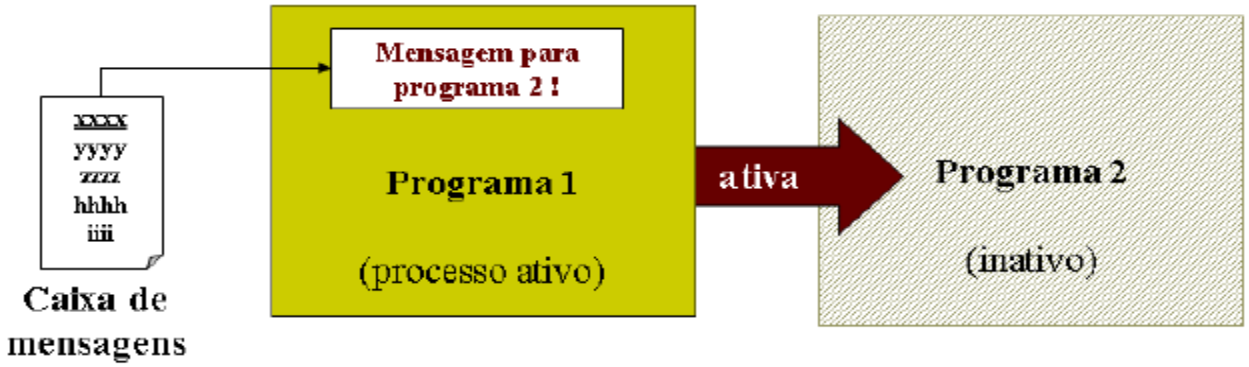
Os processos ou eventos do sistema geram novas mensagens, que são colocadas no final da fila.



Algoritmo escalonamento Cooperativo

Se o processo perceber uma mensagem que não seja para ele, chama o programa adequado, passando assim a CPU ao próximo.

Nas primeiras versões do Windows este sistema estava presente, e se chamava Multitarefa cooperativa!



Algoritmos preemptivos

1. **Round robin (circular)**

2. **Múltiplas filas (e suas variações)**

Algoritmos preemptivos

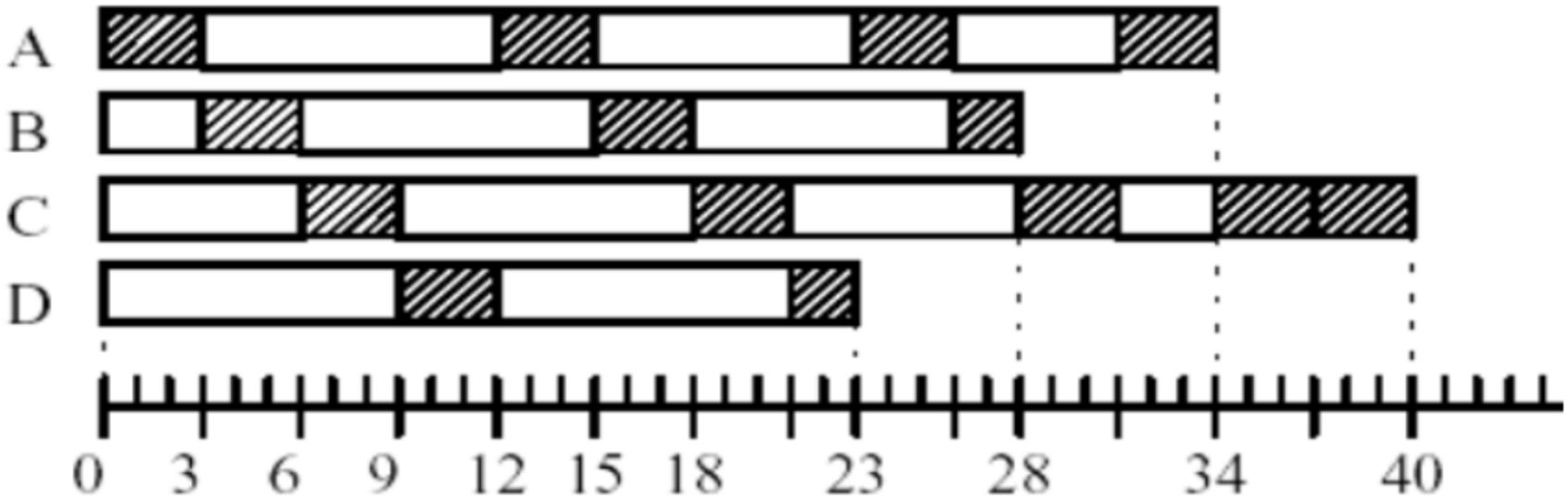
1. **Round robin (circular)**

- Algoritmo escalonamento RR

O algoritmo anterior, apesar de cooperativo, não garantia a execução de todos os programas, já que eles não poderiam ser obrigatoriamente parados (não preemptivos).

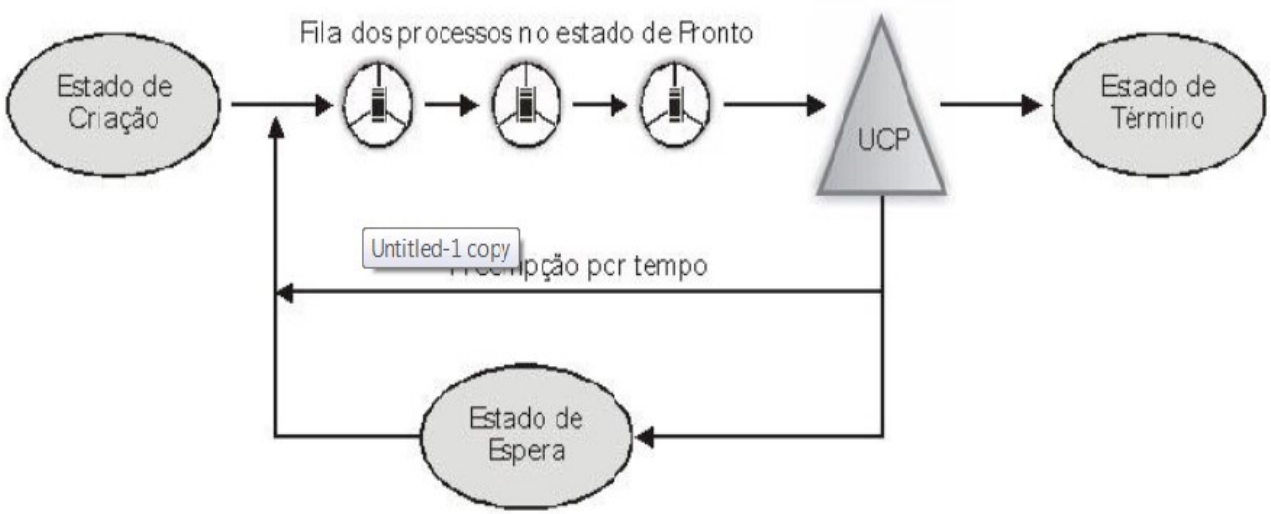
- Em sistemas de timesharing o tempo de CPU deve ser compartilhado, sendo dada uma parcela de tempo a cada programa (time-slice).

- Denominamos este intervalo de tempo de quantum.



Algoritmo escalonamento RR

- Além disso, este algoritmo é similar ao algoritmo FIFO, pois também mantemos uma fila, agora circular, para armazenar os processos.



- Para a execução existe a necessidade de um relógio para delimitar as fatias de tempo.

O processo perde o processador quando:

1. Libera explicitamente o processador;

2. Realize uma chamada de sistema (bloqueado);

3. Termina sua execução;

4.Quando sua fatia de tempo é esgotada.

Algoritmo escalonamento RR

Problema 1:

Dimensionar o quantum:

1. Compromisso entre overhead e tempo de resposta em função do número de usuários.

2. Compromisso entre tempo de chaveamento e tempo do ciclo de processador (quantum).

- Imaginamos que a cada 20 ms de processamento útil seja necessário gastar 5 ms com tarefas de troca de contexto para rodar processo de de outro usuário, por exemplo. Com isso, 20% do tempo de processador será gasto com estes overheads.

Problema 2:

Processos I/O bound são prejudicados

1. Esperam da mesma forma que processos CPU bound porém muito provavelmente não utilizam todo o seu quantum;

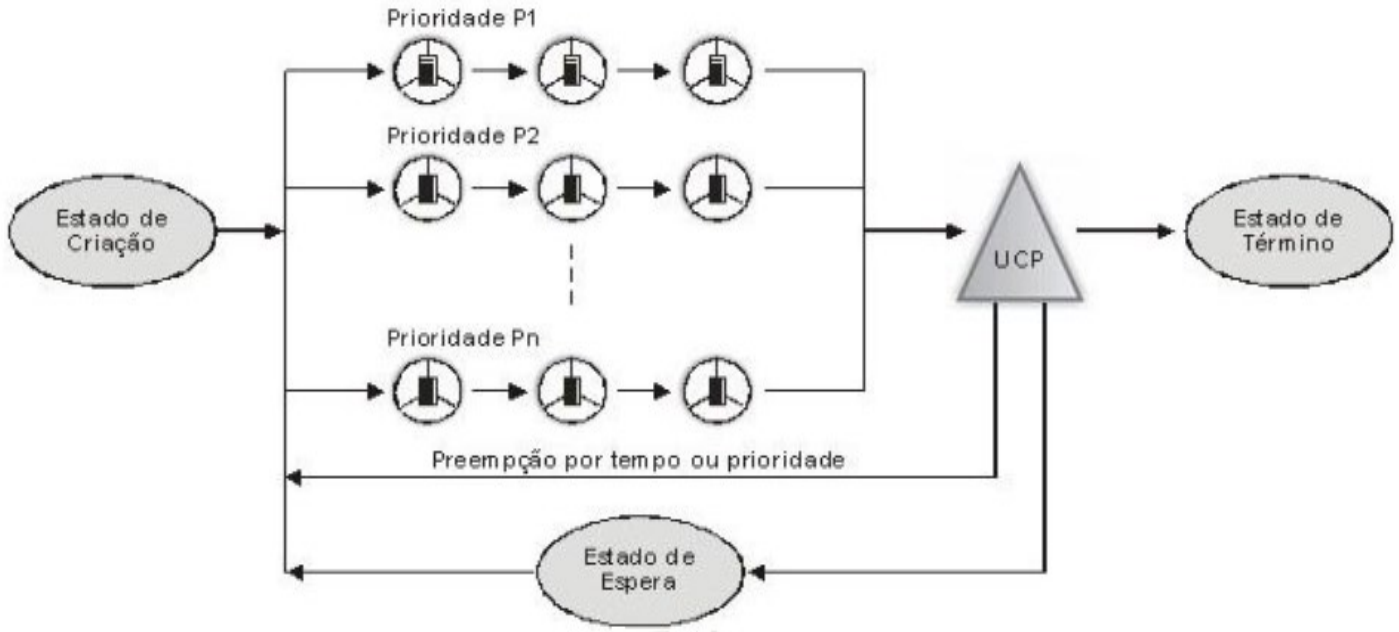
2. **Solução:** Estabelecer prioridades para os processos.

Escalonamento por Prioridades

- Neste caso, cada processo tem uma prioridade associada (**FIFO por prioridade**).

- Quando um processo de maior prioridade entra na fila, ele passa na frente, como num banco, onde os idosos tem preferência.

Escalonamento por Prioridades



- Este algoritmo é recomendado para sistemas de processamento em tempo real, onde determinadas atividades ou processos tem prioridade sobre os demais e devem ser tratados no momento em que ocorrem.

- Um problema que surge é o dos processos com menor prioridade não serem nunca processados se existirem processos de maior prioridade. Neste caso, o processo "morre de fome" (***starvation***).

- Em alguns casos o Sistema Operacional pode mudar a prioridade dos processos, minimizando este problema.

Escalonamento por Prioridades

1. Associar prioridades a processos I/O bound para compensar o tempo gasto em estado de espera (apto);

2. Sempre que um processo de maior prioridade que o processo atualmente em execução entrar no estado apto deve ocorrer uma preempção;

3. Escalonamento com prioridades é inerente a preempção;

4. É possível haver prioridade não-preemptiva;

5. Escalonador deve sempre selecionar o processo de mais alta prioridade.

Escalonamento por Prioridades

Prioridade **estática** versus **dinâmica**

- Prioridade **estática**: Um processo é criado com uma determinada prioridade e esta prioridade é mantida durante todo o tempo de vida do processo;

- Prioridade **dinâmica**: A prioridade do processo é ajustada de acordo com o estado de execução do processo e/ou do sistema.

- Ex.: ajustar a prioridade em função da fração do quantum que foi realmente utilizada pelo processo q = 100 ms.

- Processo A utilizou 2ms !nova prioridade = 1/0.02 = 50

- Processo B utilizou 50ms !nova prioridade = 1/0.5 = 2

Escalonamento por Prioridades

- Problemas com prioridades

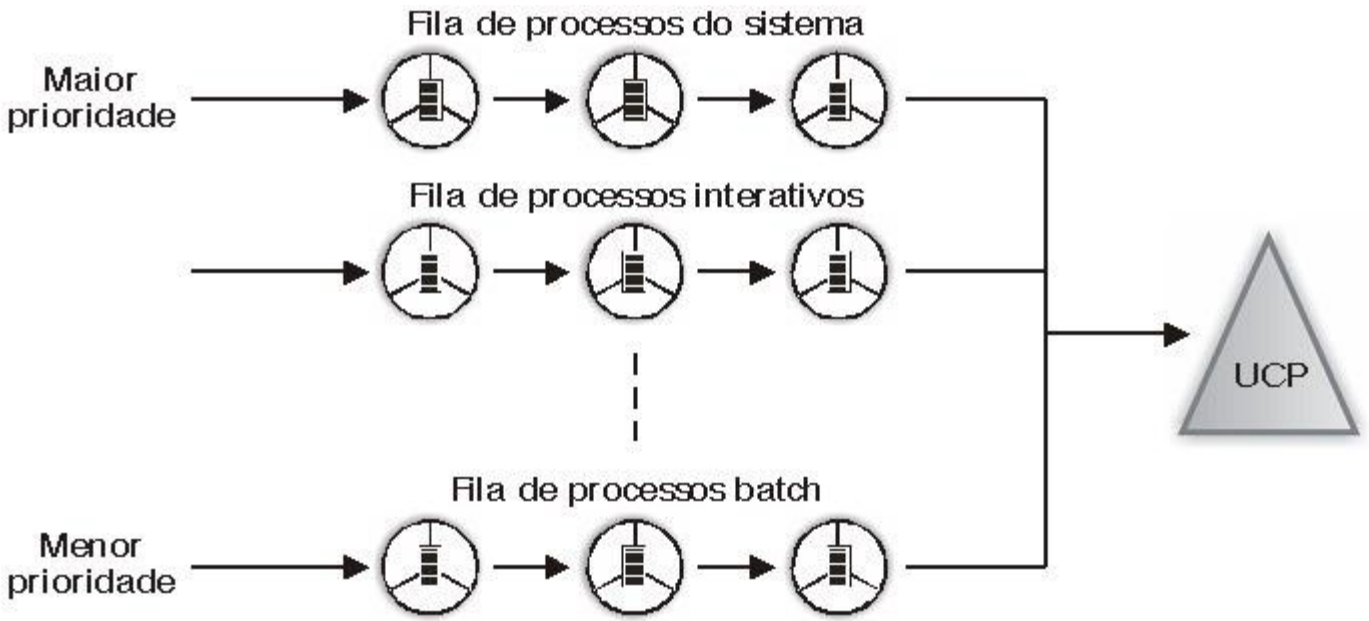
- 1. Com prioridades um processo de baixa prioridade pode não executar (starvation);

- 2. Um processo que durante sua execução troca de comportamento (CPU bound a I/O bound) pode ficar mal classificado a nível de prioridades e ser penalizado;

3. Solução: Múltiplas filas de prioridades.

- Escalonamento por Múltiplas Filas de Prioridades
- Neste modelo temos diversas filas de processos no estado pronto, cada qual com prioridades específicas e com sua própria política de escalonamento (FIFO, SJF, RR)

Escalonamento por Múltiplas Filas de Prioridades



Escalonamento por Múltiplas Filas de Prioridades com Realimentação

- Este modelo é similar ao modelo de múltiplas Filas de Prioridades, porém os processos podem trocar de fila durante seu processamento.

- Um mecanismo adaptativo é implementado de forma a avaliar o comportamento do processo e alterar sua fila.

- Por exemplo, toda vez que um processo é “preemptado” por tempo ele muda de fila, diminuindo sua prioridade e aumentando sua fatia de tempo (*time-slice*).

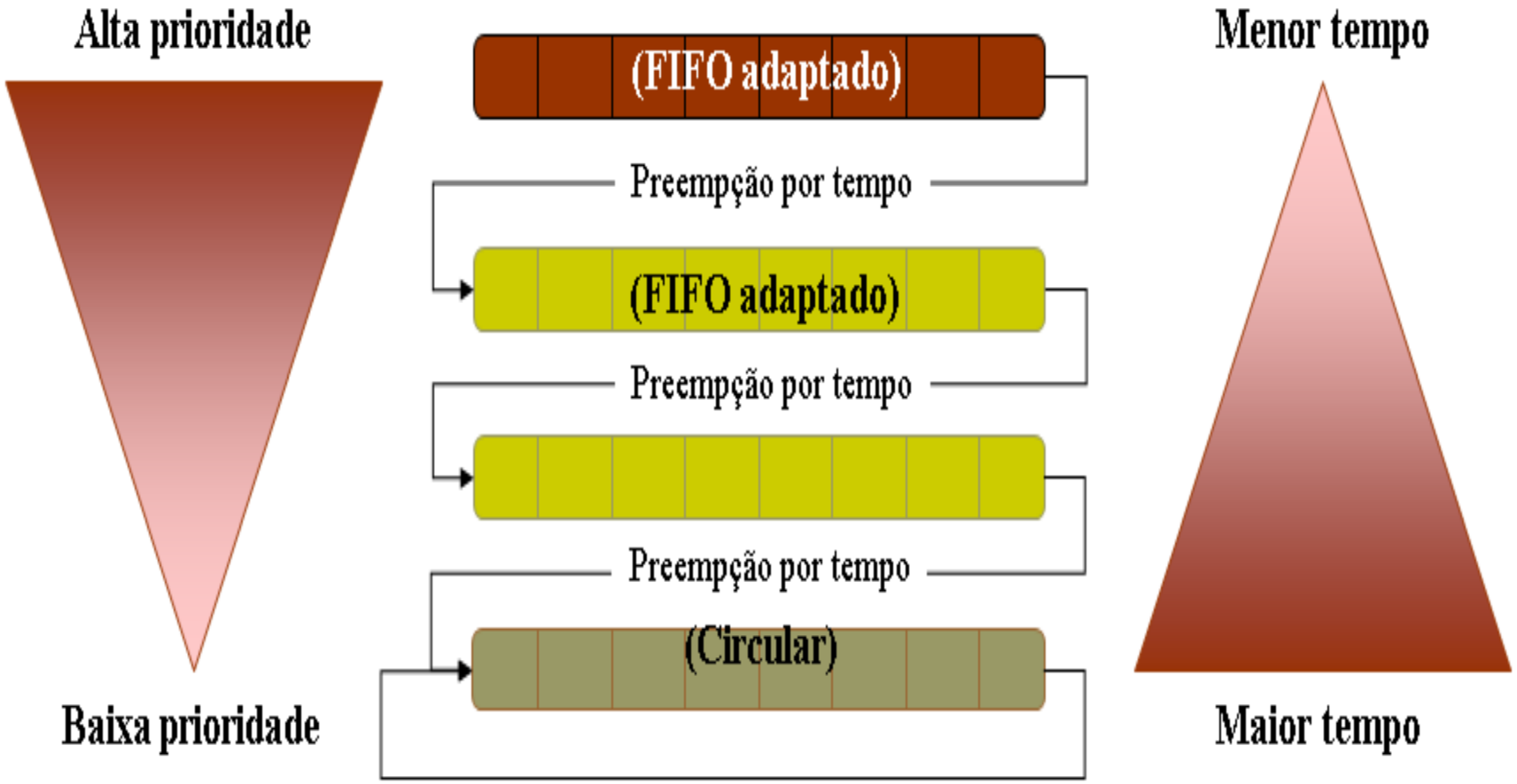
Características:

1. Baseado em prioridades dinâmicas;

2. Em função do tempo de uso da CPU a prioridade do processo aumenta e diminui;

3. Sistema de envelhecimento (aging) evita postergação indefinida.

Escalonamento por Múltiplas Filas de Prioridades com Realimentação



BIBLIOGRAFIA

MACHADO, F. B. & MAIA, L. P., **Arquitetura de Sistemas Operacionais, 4 Edição, São Paulo, LTC, 2007.**

TANENBAUM, A. S. **Sistemas Operacionais Modernos: 2ª edição, São Paulo, editora Prentice Hall, 2003.**

SILBERSCHATZ, A**. Sistemas Operacionais – Conceitos: São Paulo, editora LTC, 2004.**

Atividade

1- O que são processos em um Sistema Operacional?

2- Quais são os componentes de um processo?

3- Qual é a importância de um Gerenciador de Tarefas?