

SISTEMAS OPERACIONAIS

EAD - Escalonamento de Processos



Análise e Desenvolvimento de Sistemas

Prof. Ms. Renato Leite
renato@sectras.edu.br

Com o surgimento dos sistemas multiprogramáveis, onde múltiplos processos poderiam permanecer na memória principal compartilhando o uso da CPU, a gerência do processador tornou-se uma das atividades mais importantes em um sistema operacional. A partir do momento em que diversos processos podem estar no estado de pronto, devem ser estabelecidos critérios para determinar qual processo será escolhido para fazer uso do processador. Os critérios utilizados para esta seleção compõem a chamada política de escalonamento, que é a base da gerência do processador e da multiprogramação em um sistema operacional.

OBJETIVOS

- Entender como se gerencia a entrada de processos na fila de pronto;
- Compreender como o escalonador funciona;
- Conhecer o *dispatcher*;
- Estudar os algoritmos de escalonamento.



ESCALONAMENTO

- A política de escalonamento varia de acordo com o propósito do sistema operacional (Sistemas de Tempo Real x Sistemas de Tempo Compartilhado x Sistemas Batch);
- O escalonador (*scheduler*) implementa a política de escalonamento;
- *Dispatcher*: rotina que realiza a troca do processo em execução após a decisão do escalonador. Tempo gasto para a troca de processos é chamado latência do *dispatcher*.
- Em ambientes multithread, a unidade escalonada é a thread (somente as que estiverem no estado “pronto”);
- Para simplificar o entendimento, chamamos sempre a unidade escalonada de processo.

CRITÉRIOS DE ESCALONAMENTO

- **Throughput:** representa o número de processos executados em um determinado intervalo de tempo. Quanto maior o throughput, maior o número de tarefas executadas em função do tempo. A maximização do throughput é desejada na maioria dos sistemas;
- **Tempo de processador:** é o tempo que um processo leva no estado de execução durante seu processamento. As políticas de escalonamento não influenciam o tempo de processador de um processo, sendo este tempo função apenas do código da aplicação e da entrada de dados;
- **Tempo de espera:** é o tempo total que um processo permanece na fila de pronto durante seu processamento, aguardando para ser executado.

CRITÉRIOS DE ESCALONAMENTO

- **Tempo de turnaround:** é o tempo que um processo leva desde a sua criação até ao seu término, levando em consideração todo o tempo gasto na espera para alocação de memória, espera na fila de pronto, processamento na CPU e na fila de espera, como nas operações de E/S;
- **Tempo de resposta:** é o tempo decorrido entre uma requisição ao sistema ou à aplicação e o instante em que a resposta é exibida. Em sistemas interativos, podemos entender tempo de resposta como o tempo decorrido entre a última tecla digitada pelo usuário e o início da exibição do resultado no monitor.

CRITÉRIOS DE ESCALONAMENTO

- Vemos então que escalonamento é um compromisso entre a disponibilização de uso do processador pelo tempo de espera por esse uso;
- Os algoritmos de escalonamento enfrentam o desafio de evitar principalmente o *starvation*, ou seja, que um processo espere indefinidamente para entrar em execução;

PREEMPÇÃO

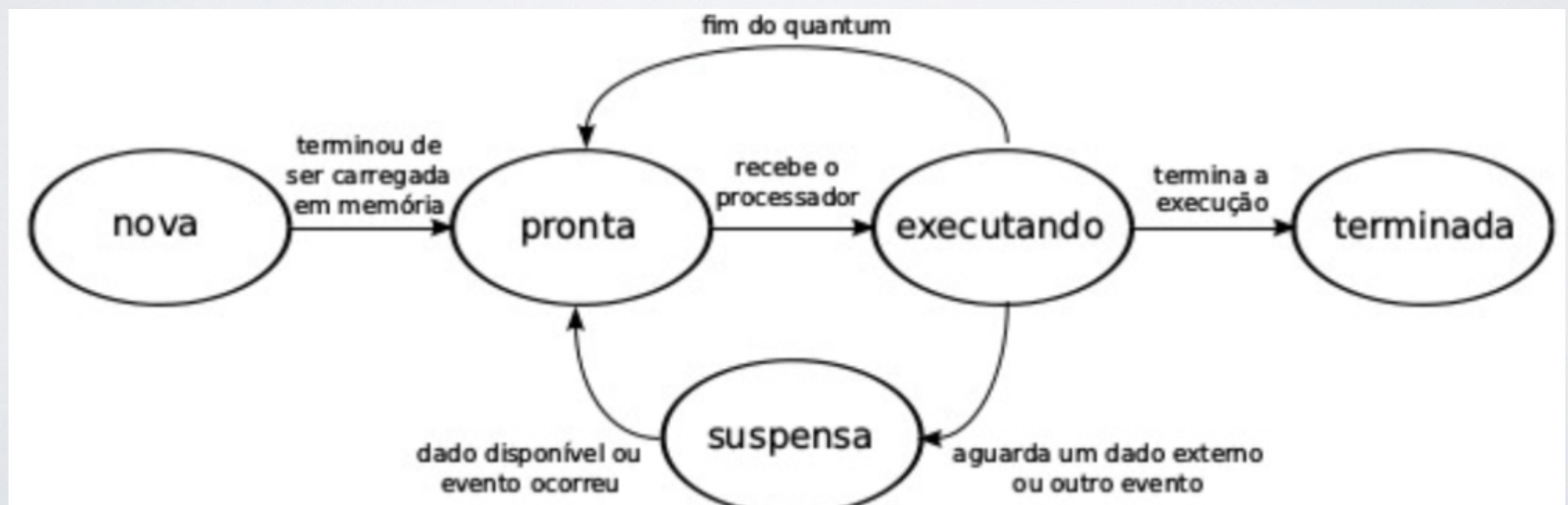
- Interrupção de um processo em execução para ser substituído por outro;
- O escalonamento de processos pode ser:
 - Não preemptivo: processamento batch;
 - Preemptivo: possibilidade de interrupção de um processo em execução sob um determinado critério de preferência (baseado no algoritmo escolhido).

ESTADOS DO PROCESSO E PREEMPÇÃO

Quais os estados possíveis vistos em sala de aula para processos em tempo compartilhado?

ESTADOS DO PROCESSO E PREEMPÇÃO

Quais os estados possíveis vistos em sala de aula para processos em tempo compartilhado?



PREEMPÇÃO

- A preempção permite que os critérios que estudamos a pouco sejam trabalhados para atender da melhor forma possível aos programas em execução em sistemas multitarefas de tempo real ou de tempo compartilhado;
- Atualmente a imensa maioria dos sistemas utiliza políticas de escalonamento preemptivas.

ALGORITMOS DE ESCALONAMENTO

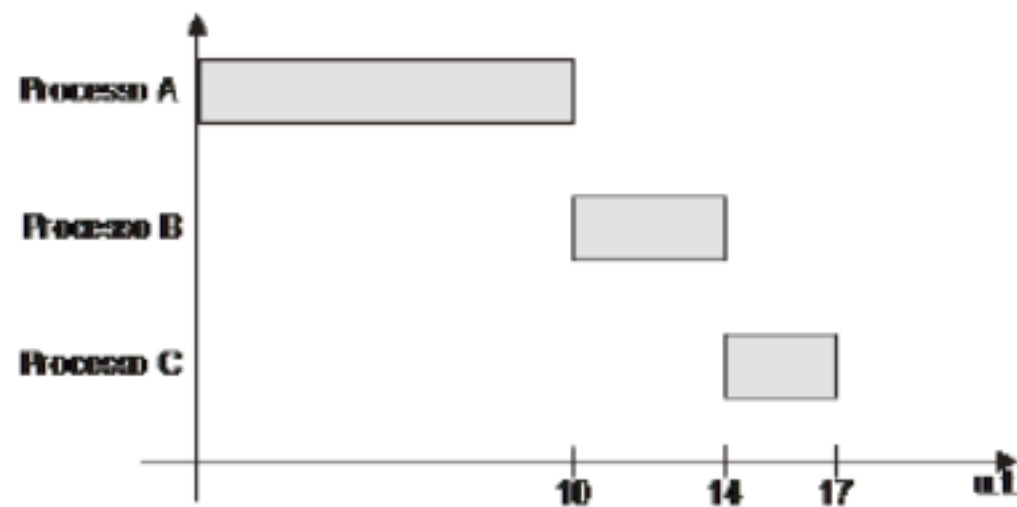
- Não preemptivo:
 - FIFO;
 - Shorted-job-first;
 - Cooperativo.
- Preemptivo:
 - Circular (Round-robin);
 - Escalonamento com prioridade;
 - Circular com prioridade;
 - Multiplas filas;

FIFO

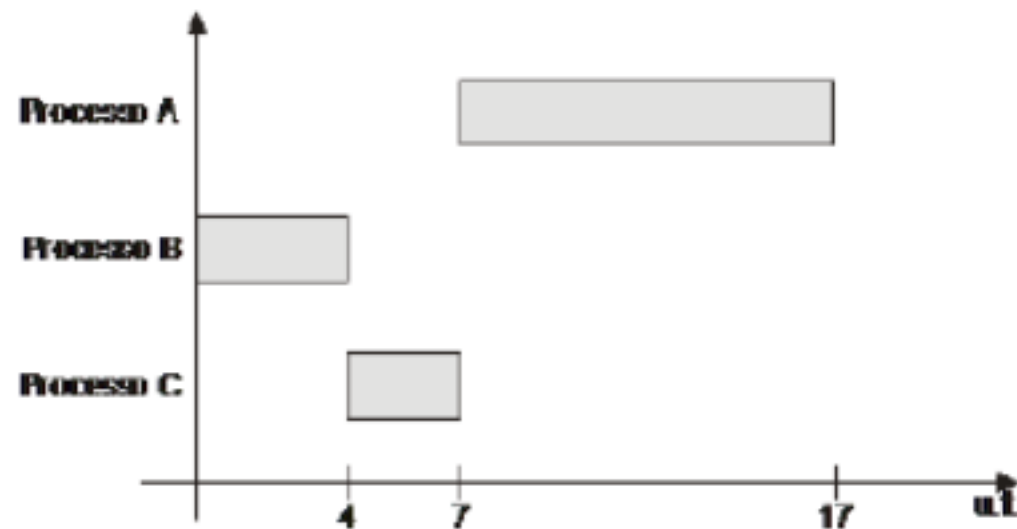
- First-in-first-out;
- Uma fila;
- Nesse algoritmo os processos tendem a ter uma espera maior quando entram programas longos em execução. Não otimiza o tempo total de vida do processo (*turnaround*);
- Vamos analisar o turnaround e tempo de espera dos processos nos dois exemplos a seguir.



FIFO



Processo	Tempo de processador (u.t.)
A	10
B	4
C	3



- Caso 1:

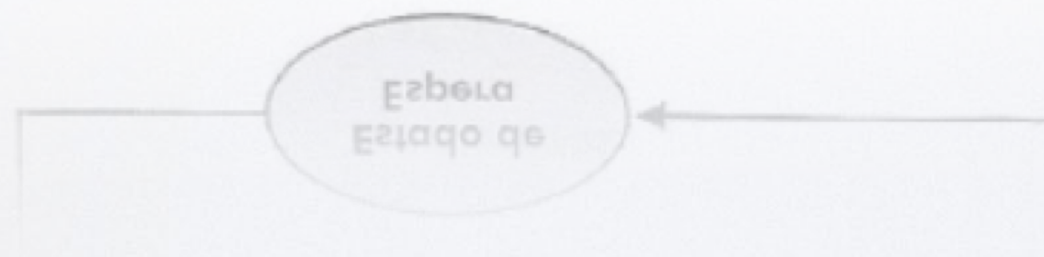
$$(0+10+14)/3 = 8 \text{ u.t.}$$

- Caso 2:

$$(0+4+7)/3 = 3.7 \text{ u.t.}$$



FIFO



SHORT-JOB-FIRST : SJF

- Seleciona o processo em estado “pronto” que necessitar de menos tempo de execução;
- O tempo de execução atribuído ao processo é uma estimativa baseada em execuções passadas;
- Reduz o tempo médio de espera dos processos;
- Problema: estimar tempo gasto por processos interativos (dependem da ação do usuário);
- Utilizado nos primeiros sistemas multi-tarefa batch;
- Como seria uma versão com preempção?

SHORT-JOB-FIRST

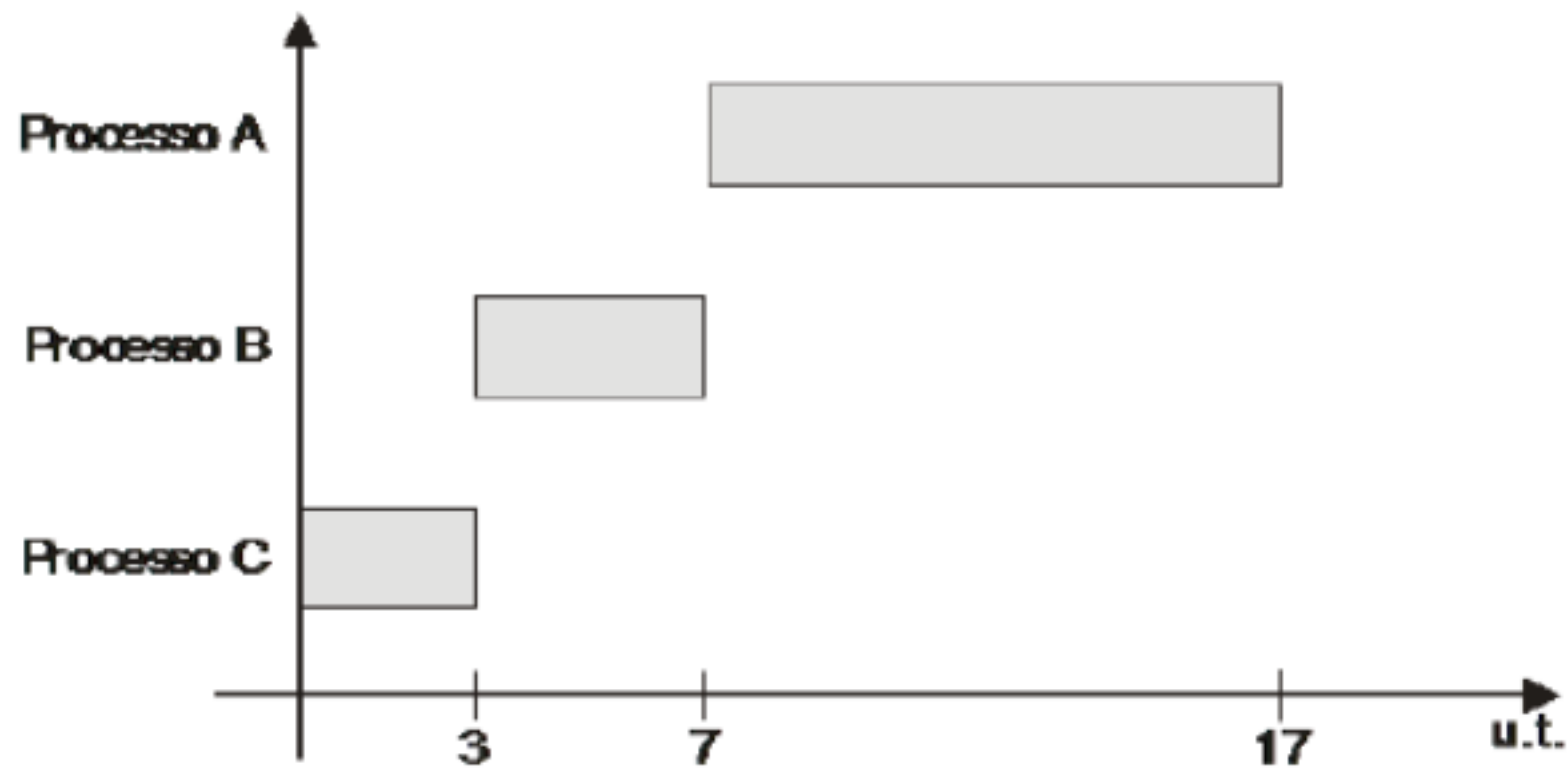
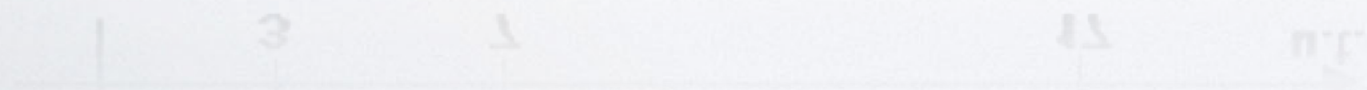


Figura 8.4: Exemplo de escalonamento SJF

- Caso único:
 $(0+3+7)/3$
 $=3.3 \text{ u.t.}$

Figura 8.4: Exemplo de escalonamento SJF



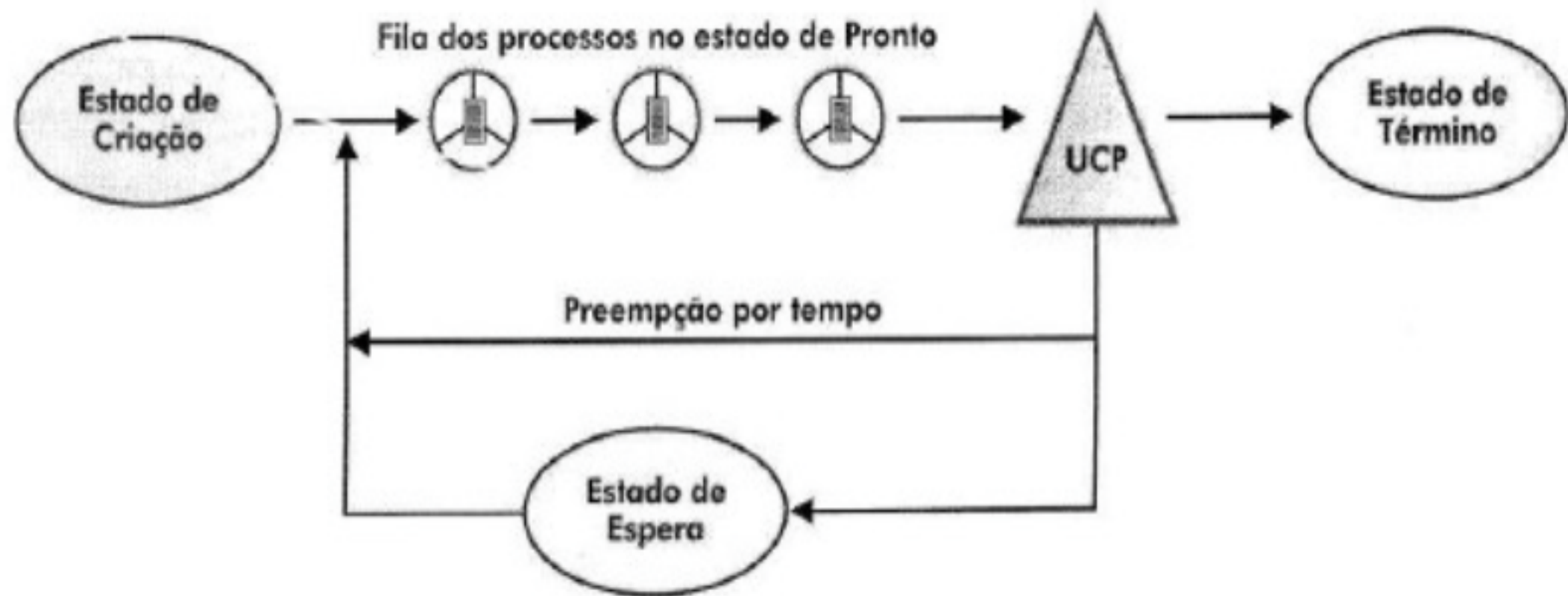
ESCALONAMENTO COLABORATIVO

- O próprio processo em execução determina se poderá ou não haver preempção, através da consulta a uma fila de mensagens enviadas por outros processos;
- Não pode ser utilizado em Sistemas de Tempo Real. Por quê?
- Problema: um processo pode dominar o uso do processador caso não esteja apto a verificar a fila de mensagens e provocar a preempção;
- Primeiros sistemas operacionais da família Windows (conhecido como multitarefa cooperativo).

ROUND ROBIN

- FIFO com preempção;
- Time Slice:
 - Grande: mesmos problemas do FIFO (tempo médio de espera alto);
 - Pequeno: alto custo de preempção, perde-se muito tempo com troca de contexto (10 a 100 milissegundos);
- Utilizado em sistemas que executam processos interativos.
- Tempo de CPU normalmente varia entre 10 e 100 milissegundos;

ROUND ROBIN

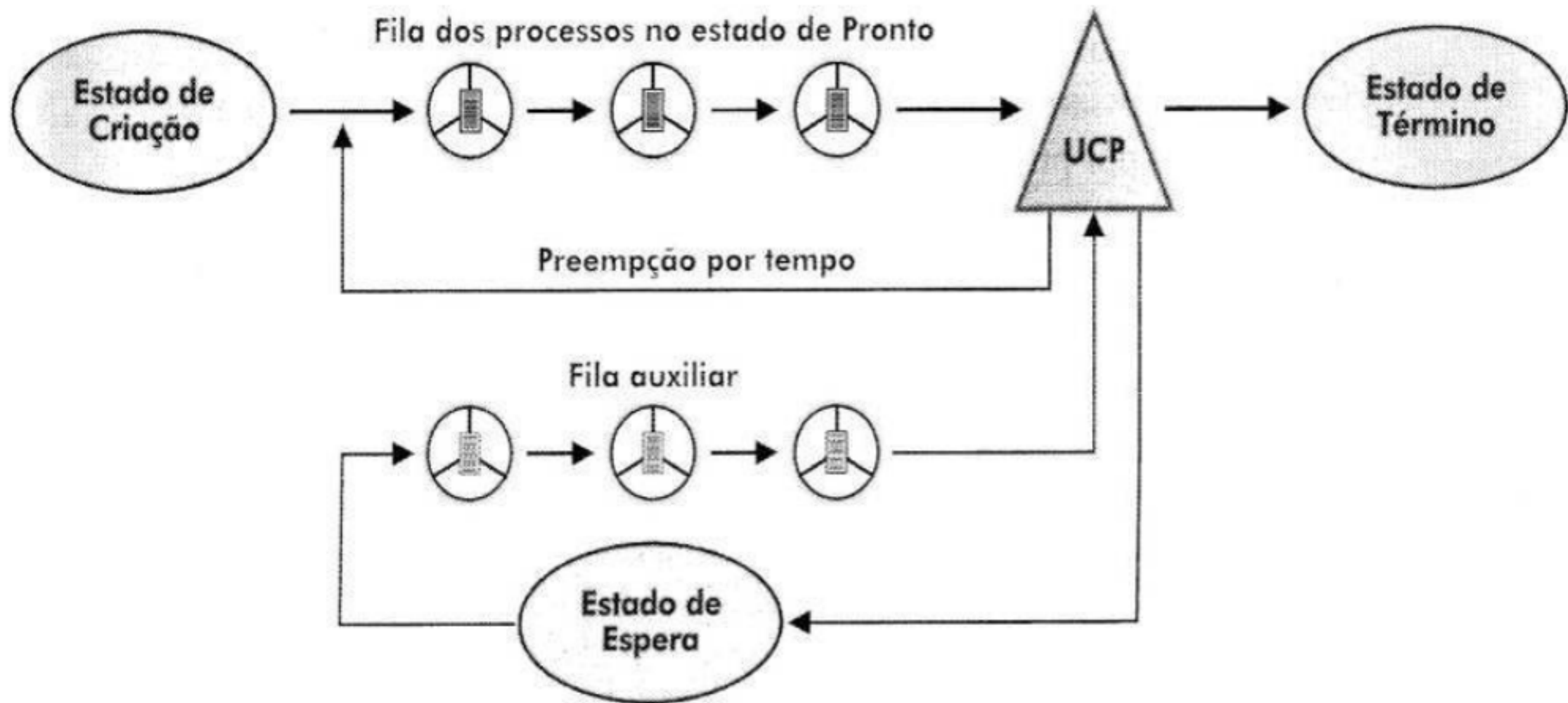


**Vamos fazer um gráfico com
quantum = 2u.t e A = 8u.t, B =
4u.t. e C = 3u.t**

ROUND ROBIN MELHORADO

- Qual o problema com o escalonamento circular simples?
 - Resposta: processos que não fazem tanto I/O acabam não utilizando bem seu time-slice;
 - Solução: permitir que processos que fazem I/O ao retornar de seu bloqueio, sejam direcionados para uma fila auxiliar com prioridade sobre a fila circular principal.

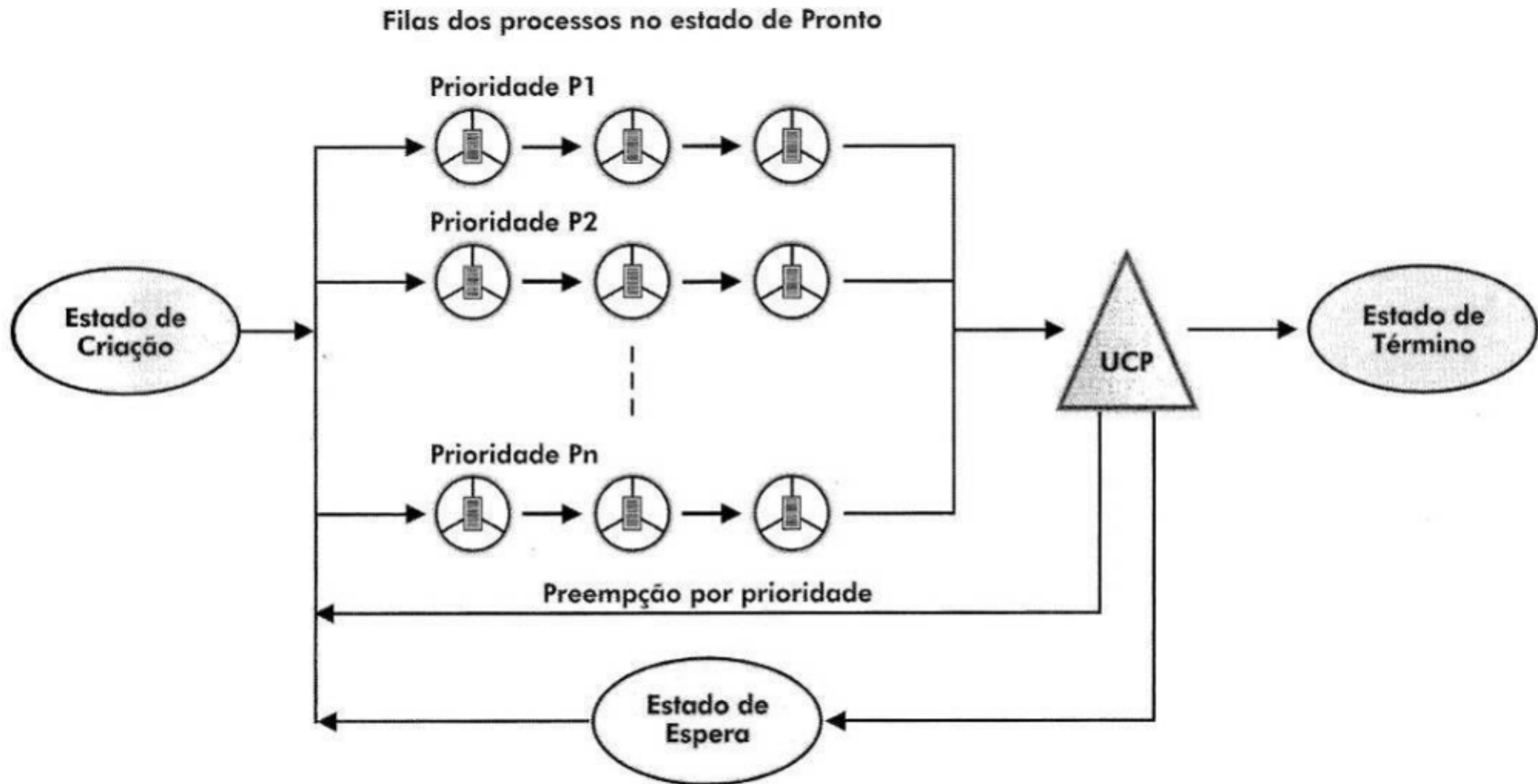
ROUND ROBIN MELHORADO



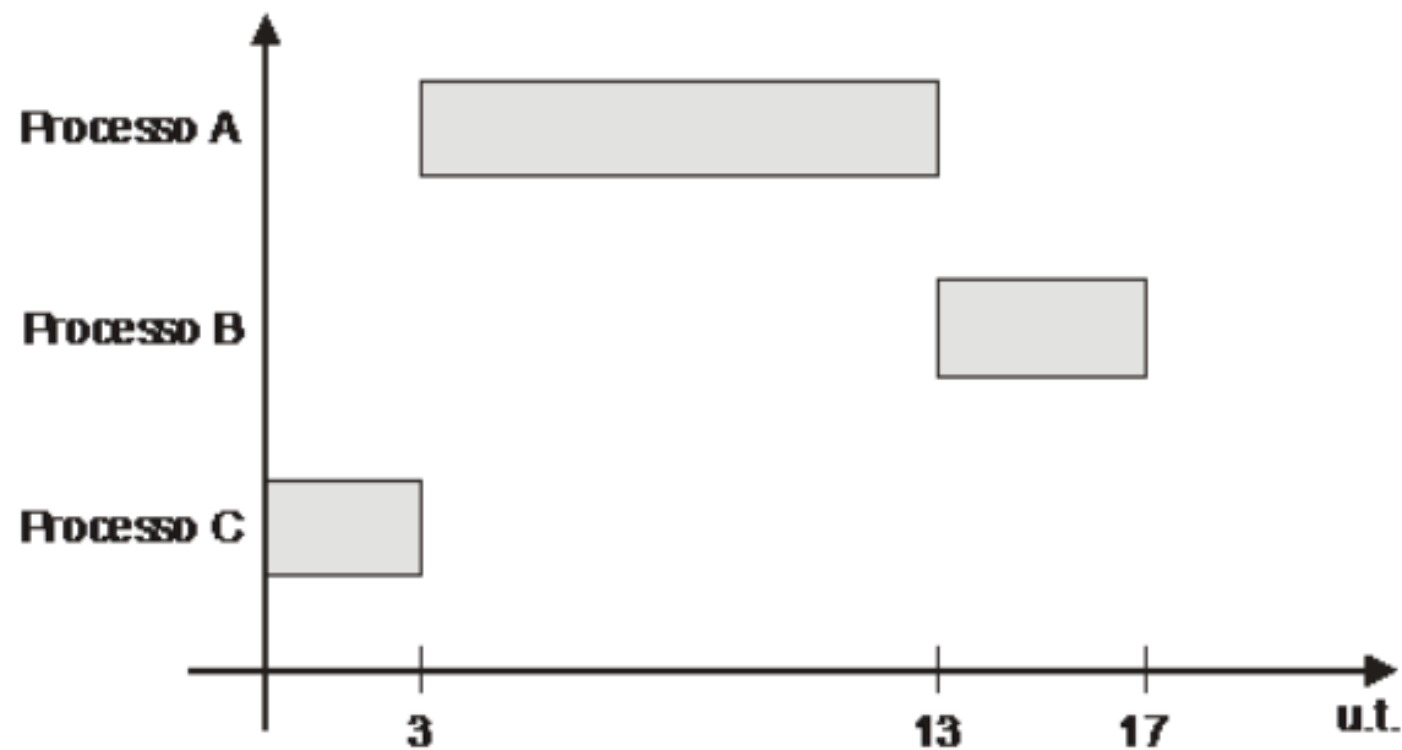
ESCALONAMENTO COM PRIORIDADE

- Prioridade de execução: valor associado ao processo (contexto de software) e escalonador sempre escolhe o de maior prioridade (FIFO para desempate);
- De tempos em tempos a rotina de escalonamento reavalia as prioridades dos processos em “pronto”. Se for detectado algum com prioridade de execução maior do que o que está sendo executado, pode ocorrer a preempção, ou simplesmente o rearranjo da fila de entrada (não-preemptivo);
- Para que processos de menor prioridade não deixem de ser executados, alguns sistemas podem incrementar a prioridade dos processos que estão a mais tempo na fila;
- Não há tempo compartilhado nesse algoritmo, apenas cooperativo ou por interrupção.

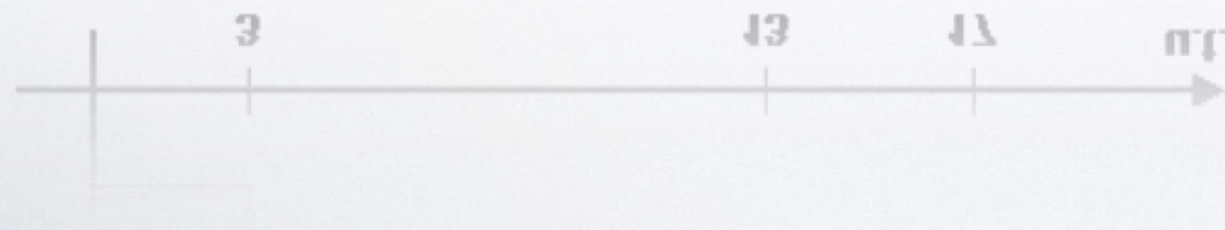
ESCALONAMENTO COM PRIORIDADE



ESCALONAMENTO COM PRIORIDADE



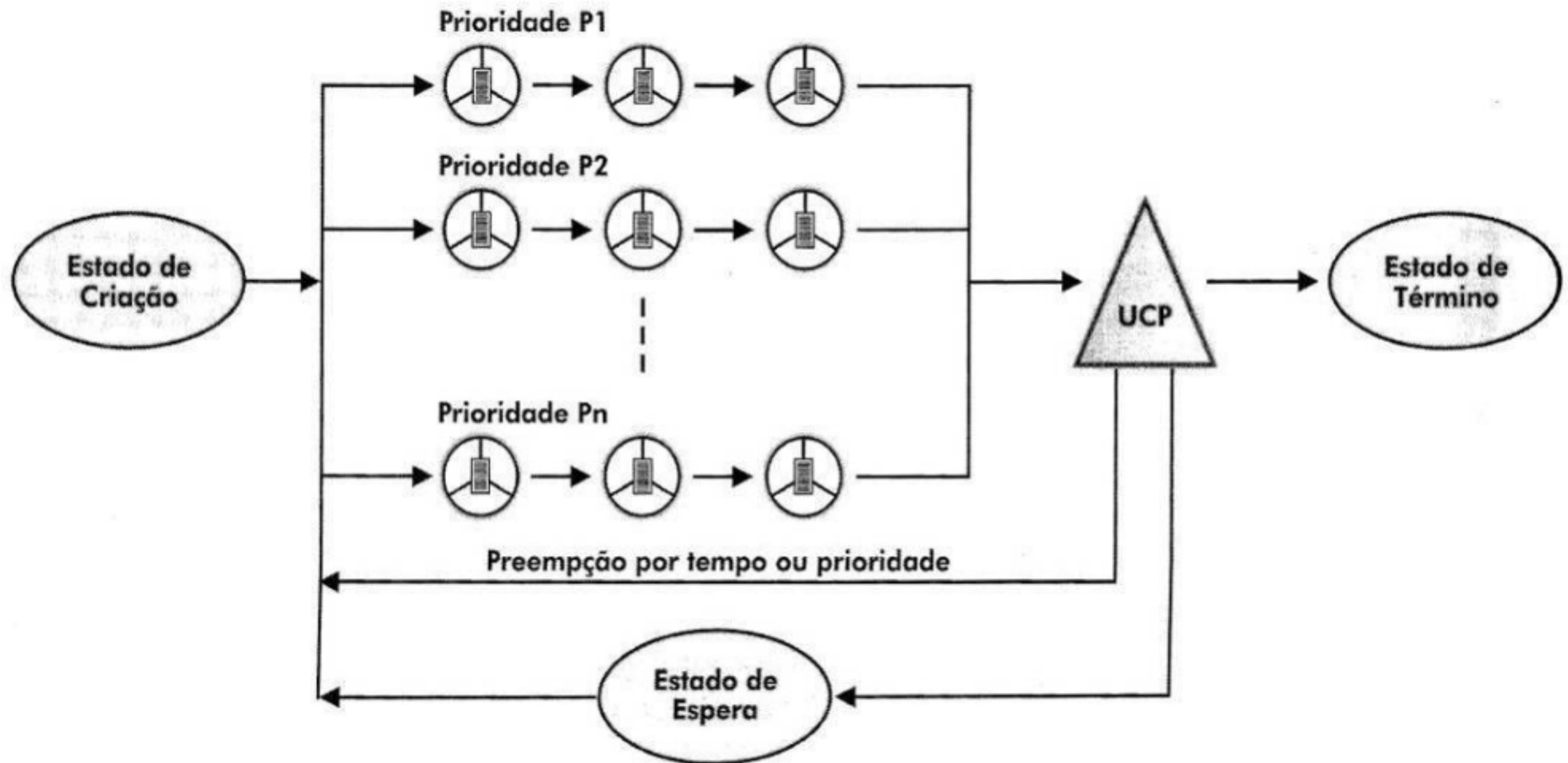
Processo	Tempo de processador (u.t.)	Prioridade
A	10	2
B	4	1
C	3	3



ESCALONAMENTO CIRCULAR COM PRIORIDADE

- Escalonamento circular com prioridade:
- Fatia de tempo + prioridade de execução;
- Utilizado na maioria dos sistemas de tempo compartilhado;
- O Windows 2000 utiliza 32 níveis de prioridade.

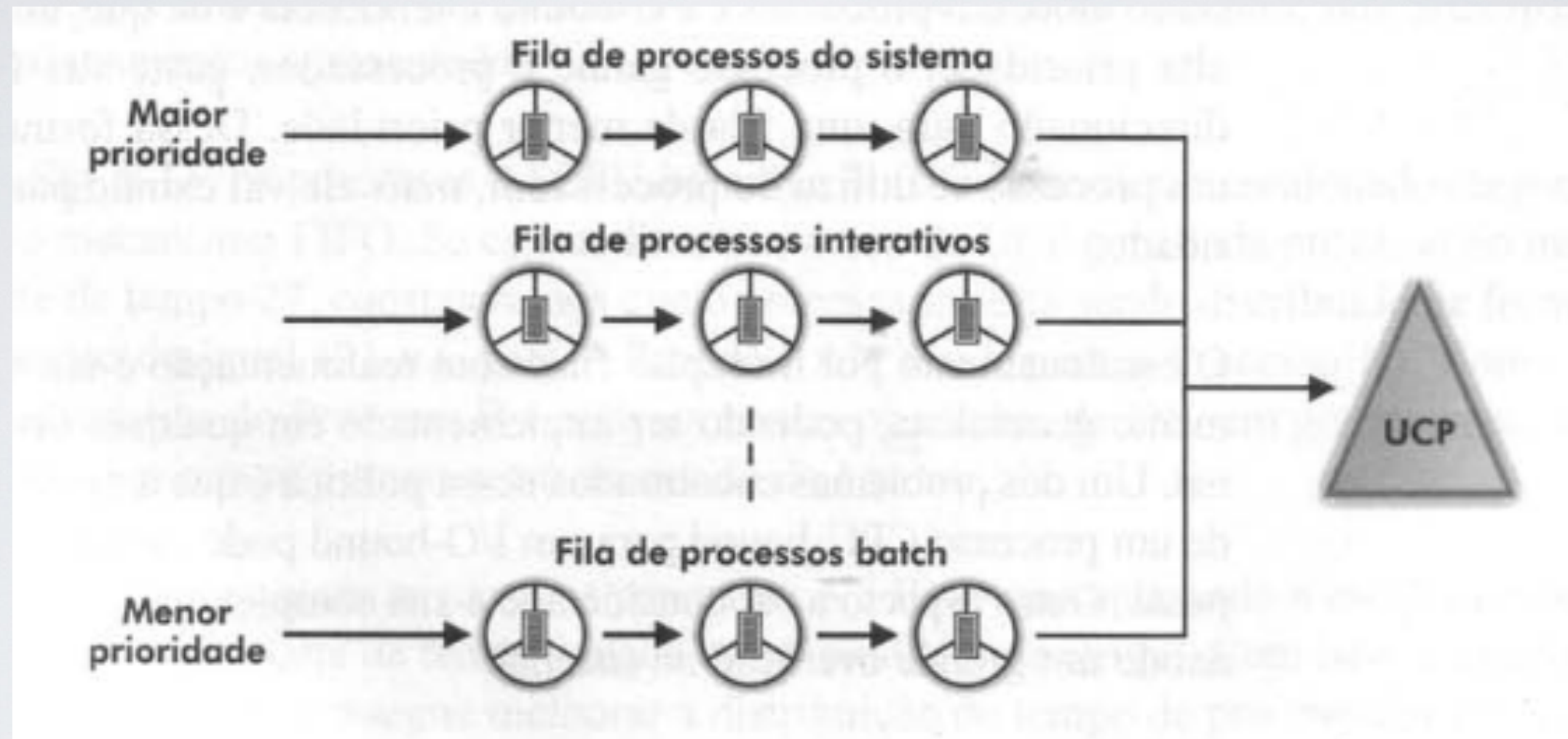
ESCALONAMENTO CIRCULAR COM PRIORIDADE



ESCALONAMENTO POR MULTIPLAS FILAS

- Existem diversas filas de processos no estado de pronto, cada qual com uma prioridade específica;
- Processos são associados as filas em função de características próprias: importância, tipo de processamento ou área de memória necessária;
- A principal vantagem de múltiplas filas é a possibilidade da convivência de mecanismos de escalonamento distintos em um mesmo sistema operacional;
- Uma fila de menor prioridade só pode ser escalonada se as de maior prioridade estiverem vazias;
- Associação é feita na criação do processo.

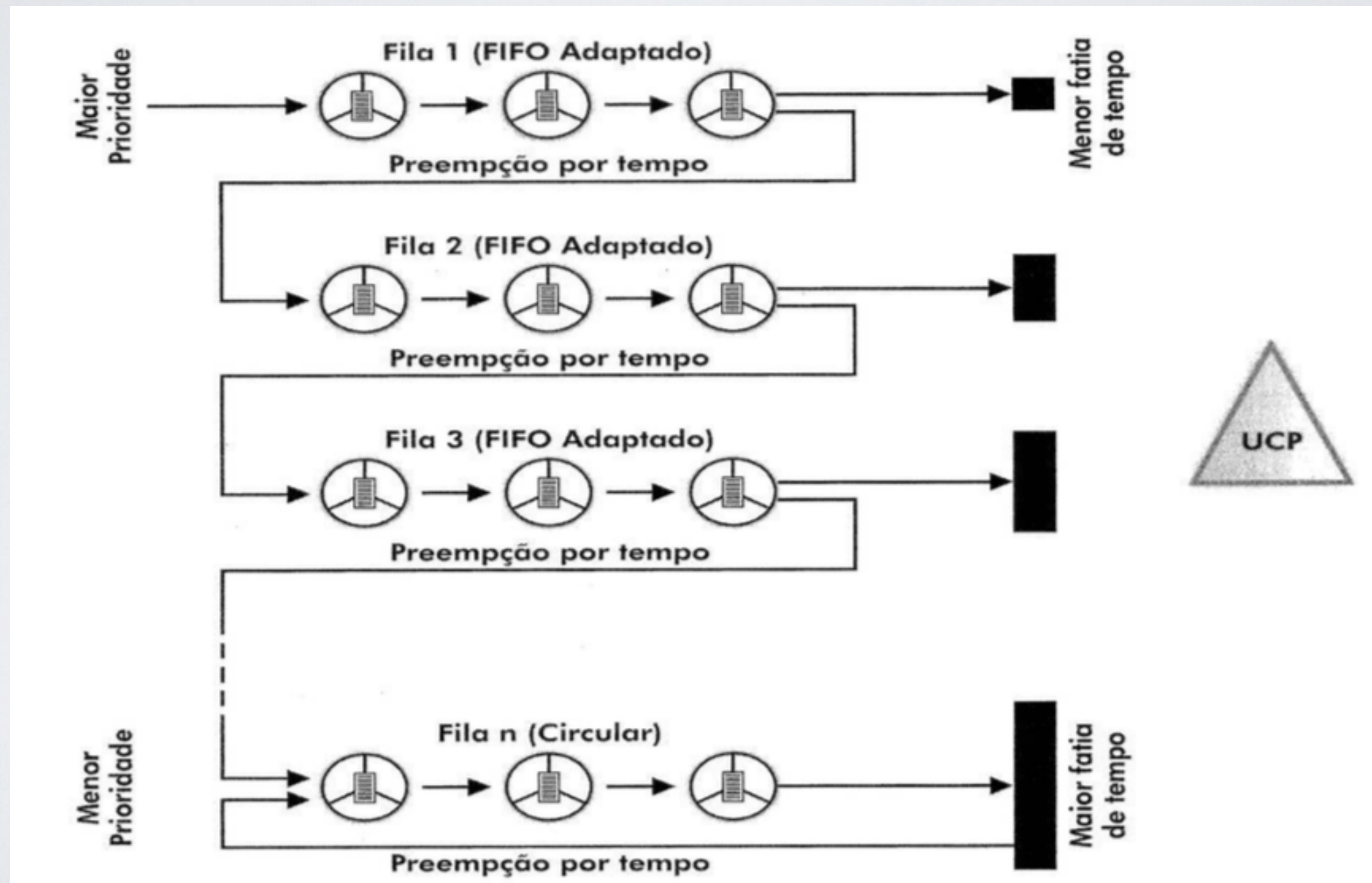
ESCALONAMENTO POR MULTIPLAS FILAS



MÚLTIPLAS FILAS COM REALIMENTAÇÃO

- Semelhante ao escalonamento por múltiplas filas, porém os processos podem trocar de fila durante seu processamento;
- Realimentado pois utiliza time-sharing;
- Vantagem: permite ao sistema operacional identificar dinamicamente o comportamento de cada processo, direcionando-o para fila com prioridade de execução e mecanismo de escalonamento mais adequado ao longo de seu processamento;
- Fila de menor prioridade é circular;
- Demais filas FIFO;

MULTIPLAS FILAS COM REALIMENTAÇÃO



EXERCÍCIOS

- Página 152 - 8.15
- MACHADO, Francis B.; MAIA, Luiz Paulo. Arquitetura de Sistemas Operacionais. 5a ed. Rio de Janeiro : LTC.

REFERÊNCIAS

- MACHADO, Francis B.; MAIA, Luiz Paulo. Arquitetura de Sistemas Operacionais. 3a ed. Rio de Janeiro : LTC, 2002.

DÚVIDAS?

