



Sistemas Operacionais
Machado/Maia
Prof. Dr. Ricardo Ramos

Capítulo 08

Gerência do Processador

8.1 Introdução

Com o surgimento dos sistemas multiprogramáveis a gerência do processador tornou-se uma das atividades mais importantes em um SO.

A política de escalonamento é a base da gerência do processador e da multiprogramação em um SO.

8.1 Introdução

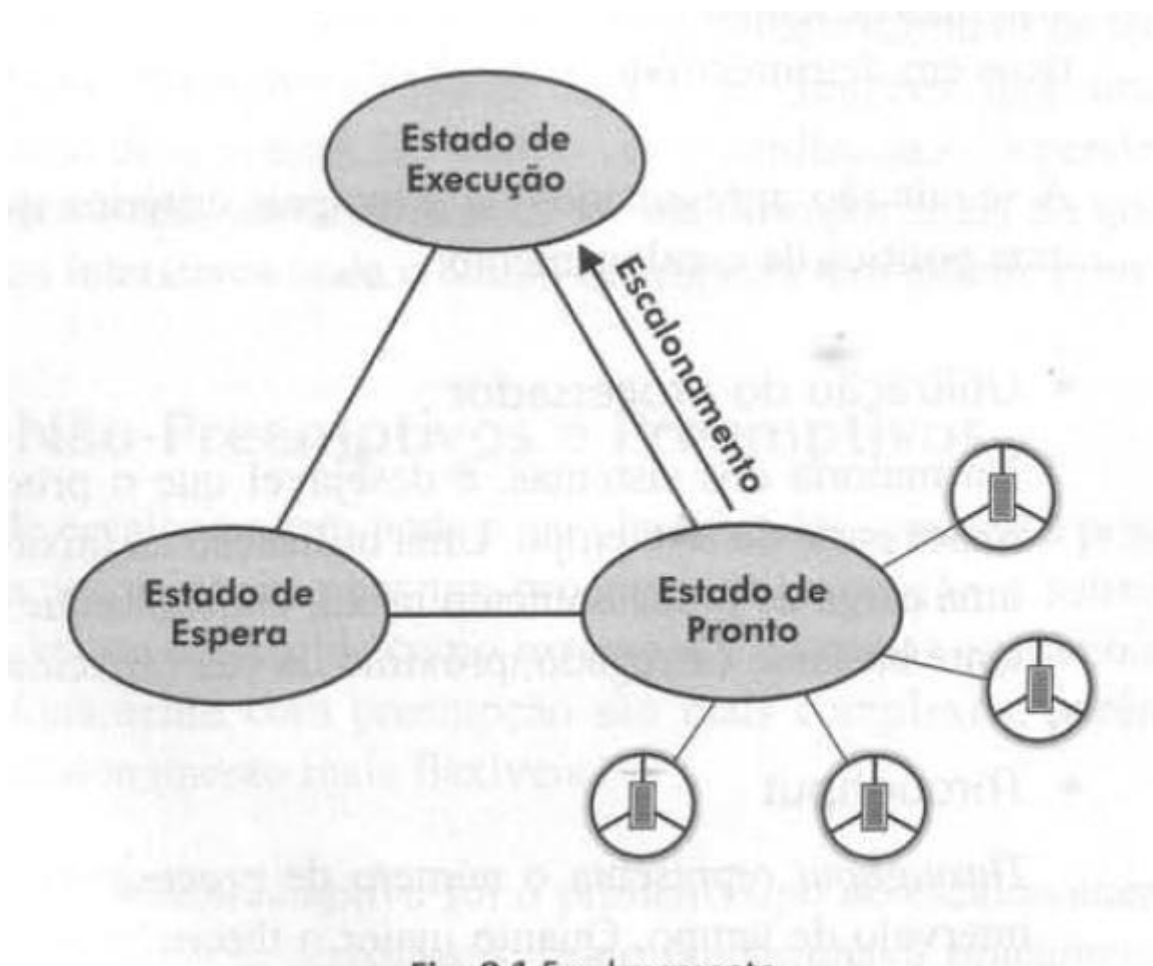


Fig. 8.1 Escalonamento.

8.2 Funções básicas

Funções básicas da política de escalonamento:

- manter o processador ocupado a maior parte do tempo;
- balancear o uso da CPU entre processos;
- privilegiar a execução de aplicações críticas;
- maximizar o *throughput* do sistema;
- oferecer tempos de respostas razoáveis para usuários interativos.

8.2 Funções básicas

A rotina do SO que tem como principal função implementar os critérios da *política de escalonamento* é denominada escalonador (*scheduler*).

Outra rotina importante na gerência do processador é conhecida como *dispatcher*, responsável pela troca de contexto dos processos. O período de tempo gasto na substituição de um processo em execução por outro é denominado latência do *dispatcher*.

8.3 Critérios de Escalonamento

Sistemas de tempo compartilhado - exigem que o escalonamento trate todos os processos de forma igual, evitando o *starvation*.

Sistemas de tempo real - devem priorizar a execução de processos críticos em detrimento da execução de outros processos.

8.3 Critérios de Escalonamento

8.3.1 - Utilização do processador

Na maioria dos sistemas é desejável que o processador permaneça a maior parte do tempo ocupado.

8.3 Critérios de Escalonamento

8.3.2 - *Throughput*

Representa o número de processos executados em um determinado intervalo de tempo.

Deseja-se maximizar o *throughput*.

8.3 Critérios de Escalonamento

8.3.3 - Tempo de CPU

Tempo de processador ou tempo de CPU é o tempo que um processo leva no **estado de execução** durante seu processamento.

8.3 Critérios de Escalonamento

8.3.3 - Tempo de Espera

Tempo de espera é o tempo total que um processo permanece na fila **(estado) de pronto** durante seu processamento, aguardando para ser executado.

Deseja-se reduzir o tempo de espera.

8.3 Critérios de Escalonamento

8.3.4 - Tempo de *Turnaround* (Tempo de vida)

É o tempo que um processo leva desde a sua criação até o seu término, levando em consideração todo o tempo gasto na espera para alocação de memória, espera na fila de pronto (tempo de espera), processamento na CPU (tempo de processador) e na fila de espera, como nas operações de E/S.

Deseja-se minimizar o *turnaround*.

8.3 Critérios de Escalonamento

8.3.5 - Tempo de Resposta

É o tempo decorrido entre uma requisição ao sistema ou à aplicação e o instante em que a resposta é exibida.

Sistemas interativos.

8.3 Critérios de Escalonamento

Em qualquer política de escalonamento busca-se otimizar a utilização do processador e o *throughput*, enquanto tenta diminuir os tempos de *turnaround*, espera e resposta.

8.5 FIFO (*First-In-First-Out*) FCFS

O processo que chegar primeiro ao estado de pronto é o selecionado para execução.



Fig. 8.2 Escalonamento FIFO.

8.5 FIFO (*FCFS*)

$$Tm_espera = (0+10+14)/3 = 8u.t$$

$$Tm_espera = (7+0+4)/3 = 3,7u.t.$$

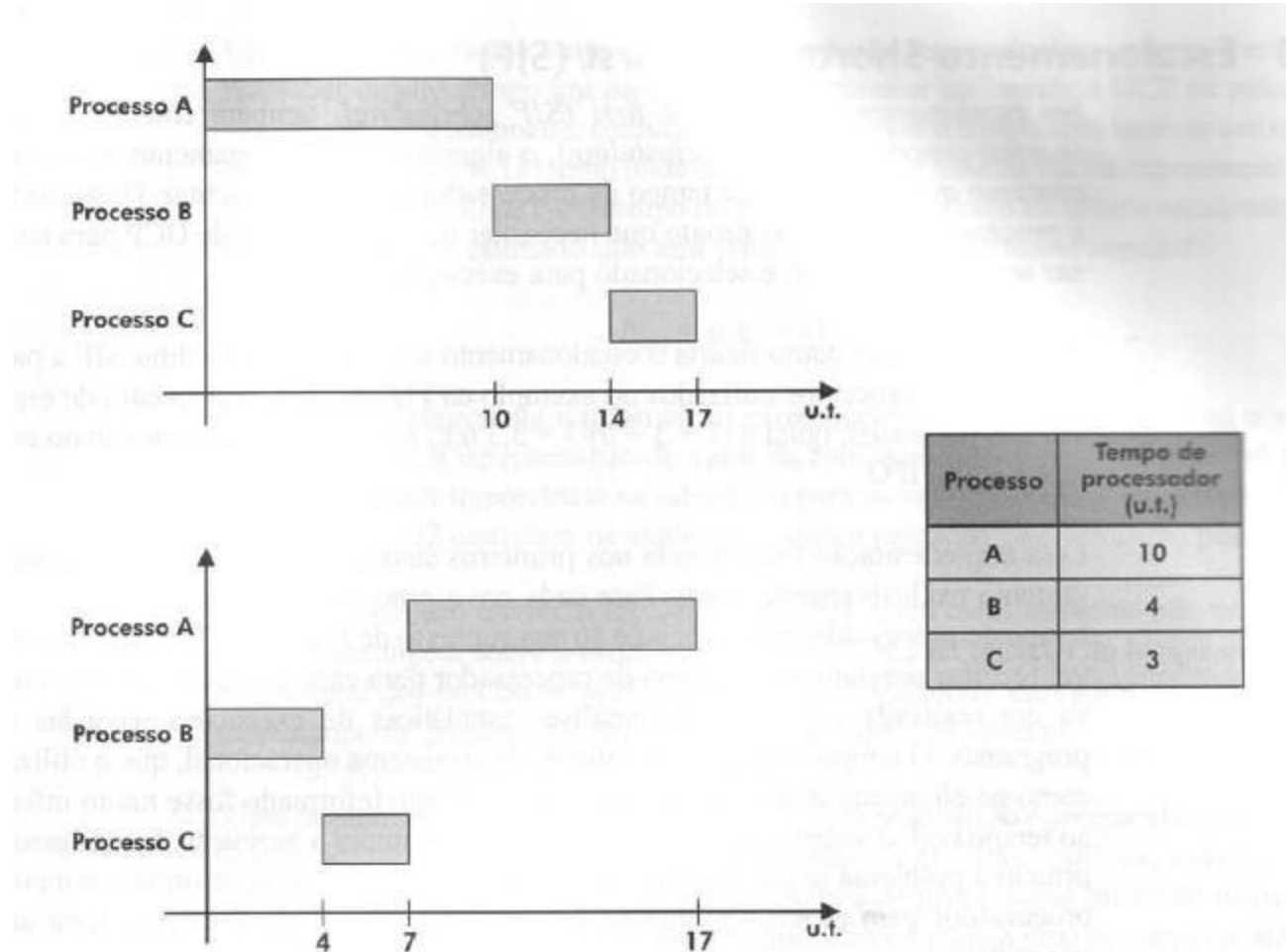


Fig. 8.3 Escalonamento FIFO (exemplo).

8.5 FIFO (*FCFS*)

Simples e cooperativo (**não-preemptivo**)

Problemas:

- não se preocupa em melhorar o T_{m_espera}
- trata diferente processos CPU-bound de I/O-bound

Inicialmente implementado em sistemas
monoprogramáveis com proc. batch

8.6 SJF (*Shortest-Job-First*)

O algoritmo de escalonamento seleciona o processo que tiver o menor tempo de processador ainda por executar.

Dessa forma, o processo em estado de pronto que necessitar de menos tempo de CPU para terminar seu processamento é selecionado para execução.

8.6 SJF (*Shortest-Job-First*)

$$\begin{aligned} Tm_espera &= \\ (7+3+0)/3 &= \\ 3,3 \text{ u.t.} \end{aligned}$$

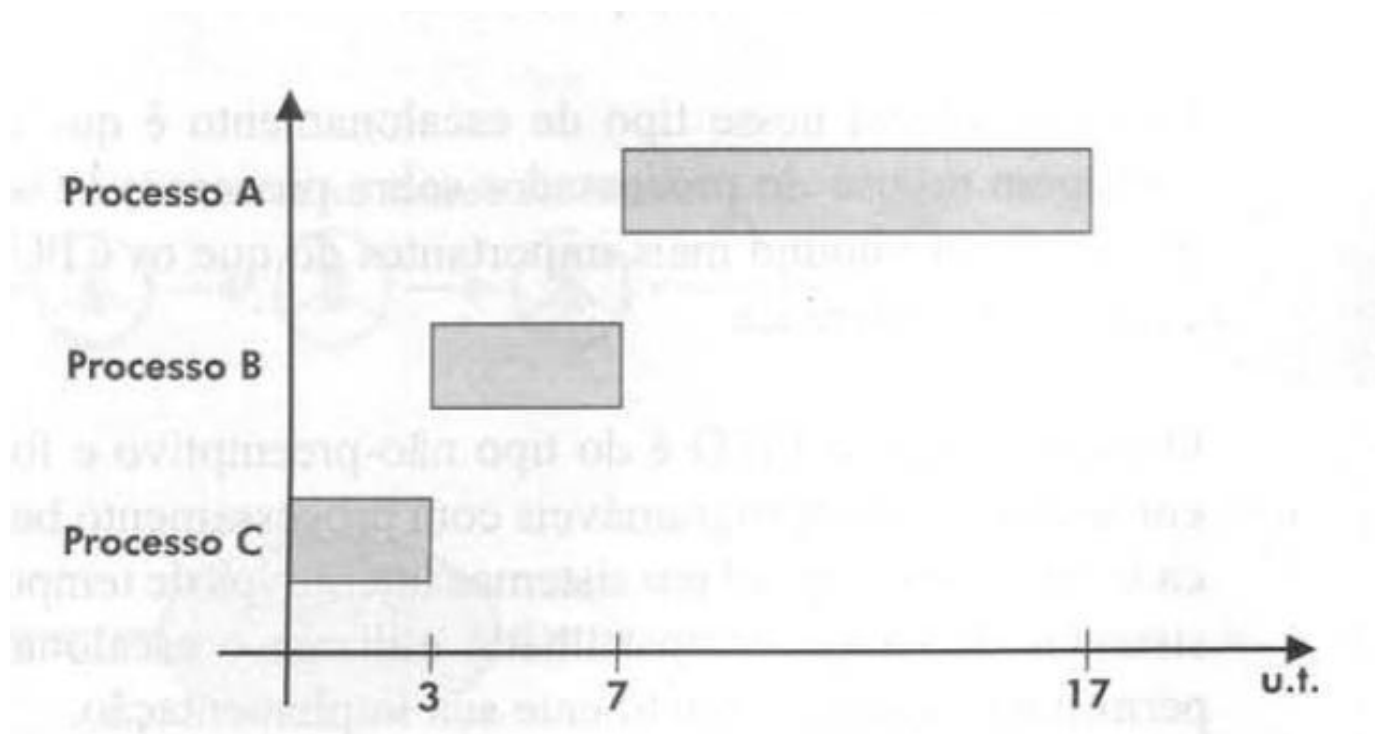


Fig. 8.4 Escalonamento SJF (exemplo).

8.6 SJF (*Shortest-Job-First*)

Problema:

- impossibilidade de estimar o tempo de processador para processos interativos, já que a entrada de dados é uma ação imprevisível

Não-preemptivo e possível de haver *starvation*.

8.8 Circular (*round robin scheduling* ou *RR*)

Preemptivo para sistemas de tempo compartilhado.

Semelhante ao FIFO porém quando um processo passa para o estado de execução existe um tempo-limite para o uso contínuo do processador denominado fatia de tempo (*time-slice*) ou *quantum*.

8.8 Circular (RR)

Preempção por tempo.

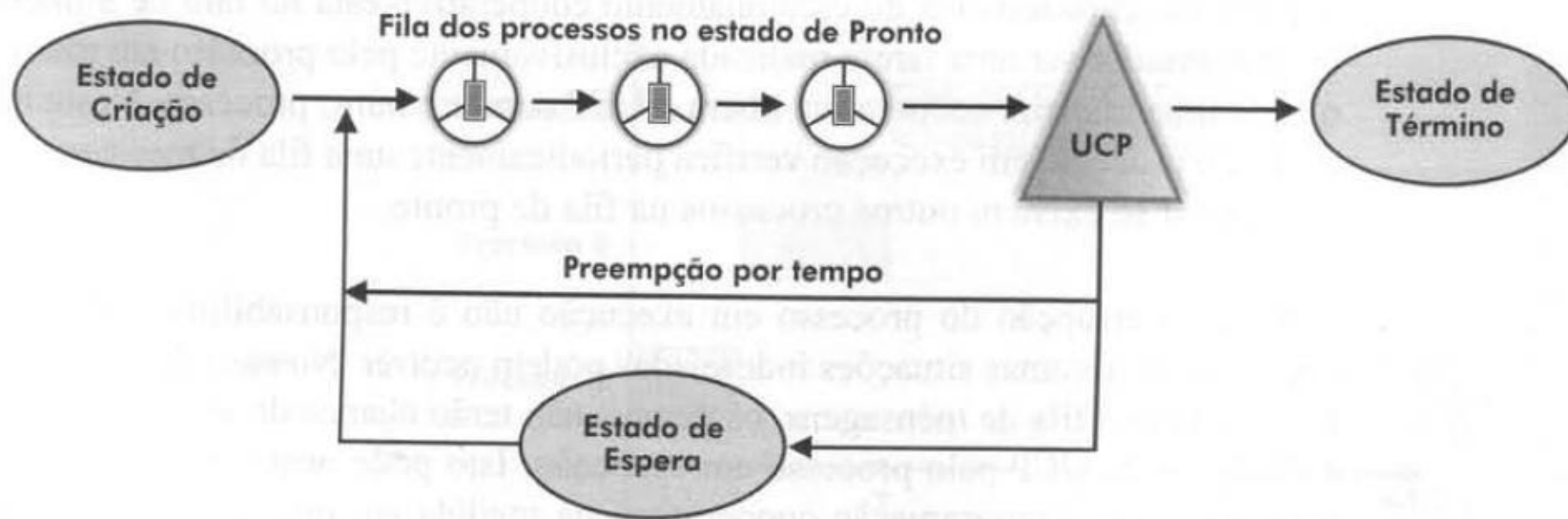


Fig. 8.5 Escalonamento circular.

8.8 Circular (*round robin scheduling*)

Quantum = 2 u.t.

Tempo de troca
de contexto
(TTC) = 0 u.t

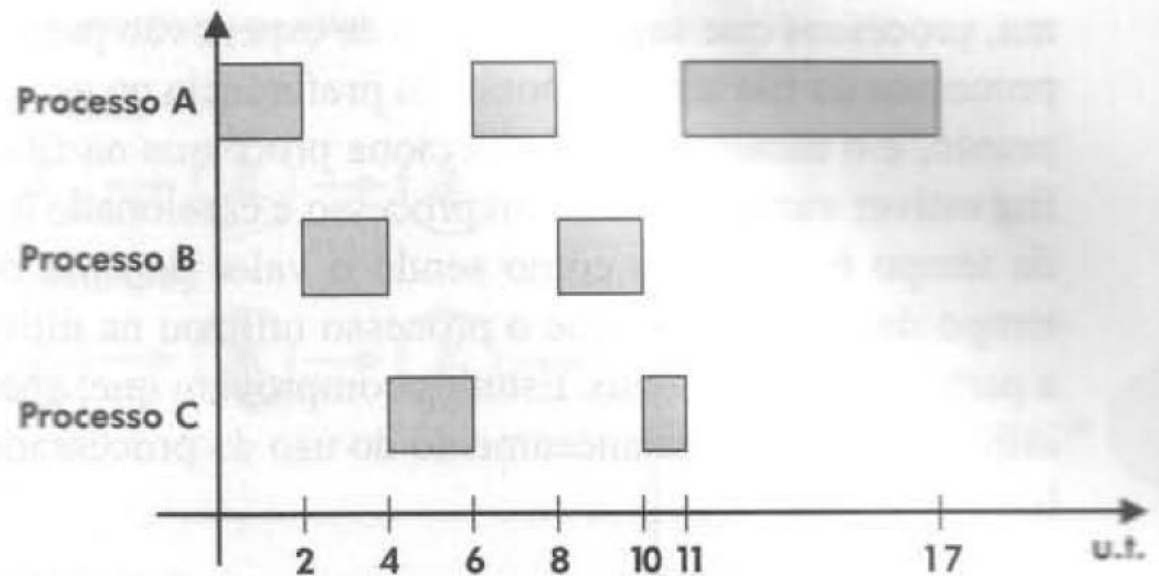


Fig. 8.6 Escalonamento circular (exemplo).

8.8 Circular (RR)

Quantum = valor alto (mesmo que o FIFO)

Quantum = valor pequeno (grande número de preempções)

Vantagem: não permite que um processo monopolize a CPU

8.9 Prioridades

Preemptivo ou **não-preemptivo** realizado com base em um valor associado a cada processo denominado prioridade de execução.

O processo com maior prioridade no estado de pronto é sempre escolhido para execução, e processos com prioridades iguais são escalonados segundo o critério FIFO (ou FCFS).

8.9 Prioridade preemptivo

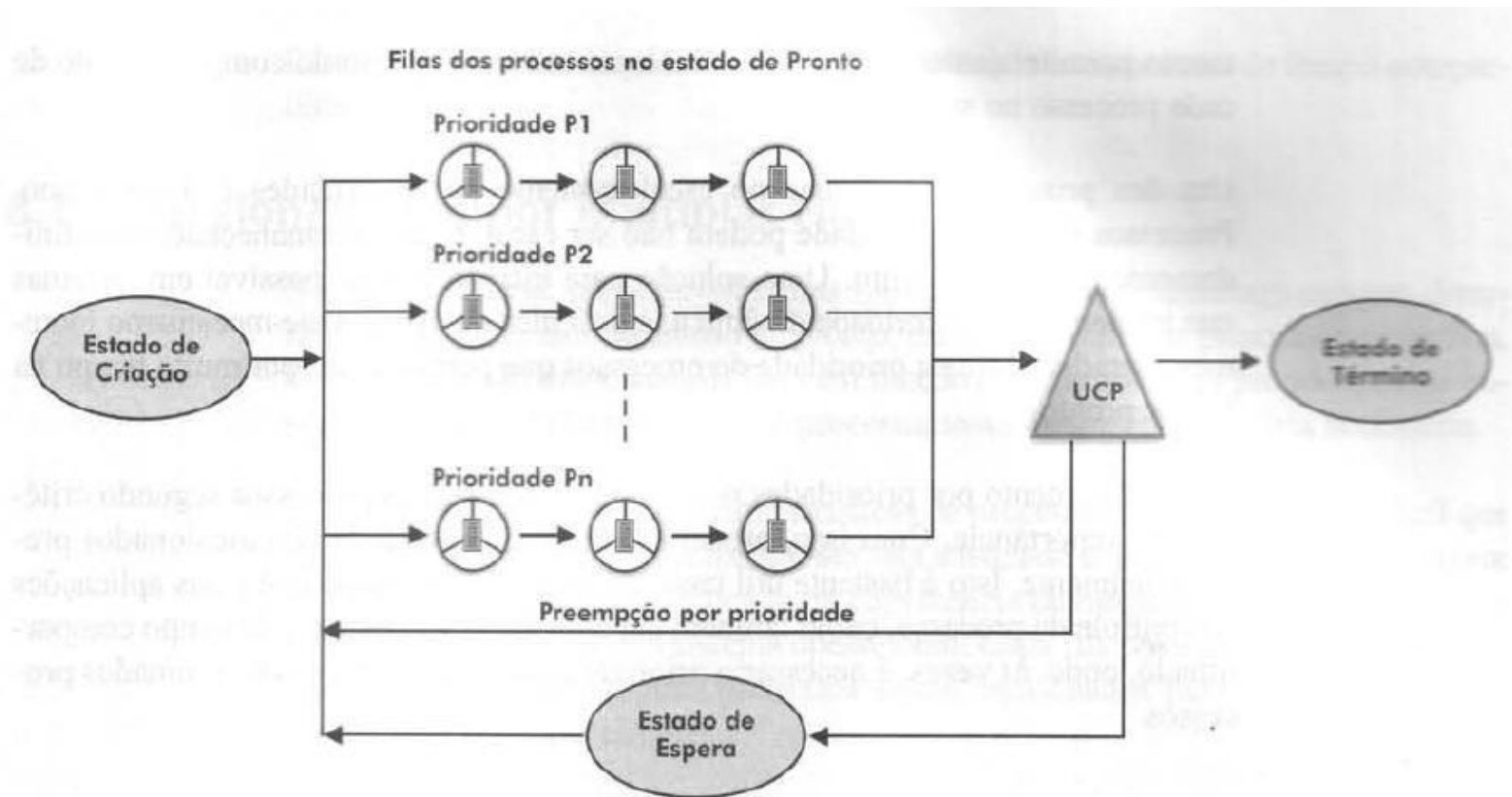


Fig. 8.8 Escalonamento por prioridades.

8.9 Prioridade

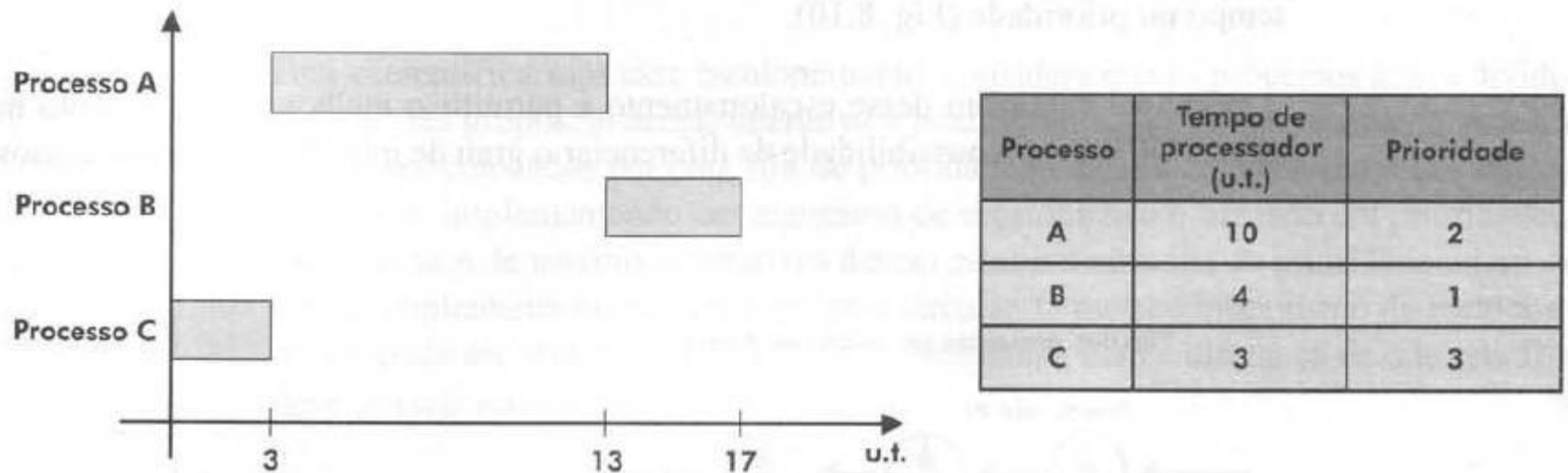


Fig. 8.9 Escalonamento por prioridades (exemplo).

$$Tm_espera = (3+13+0)/3 = 5,3u.t.$$

8.9 Prioridades

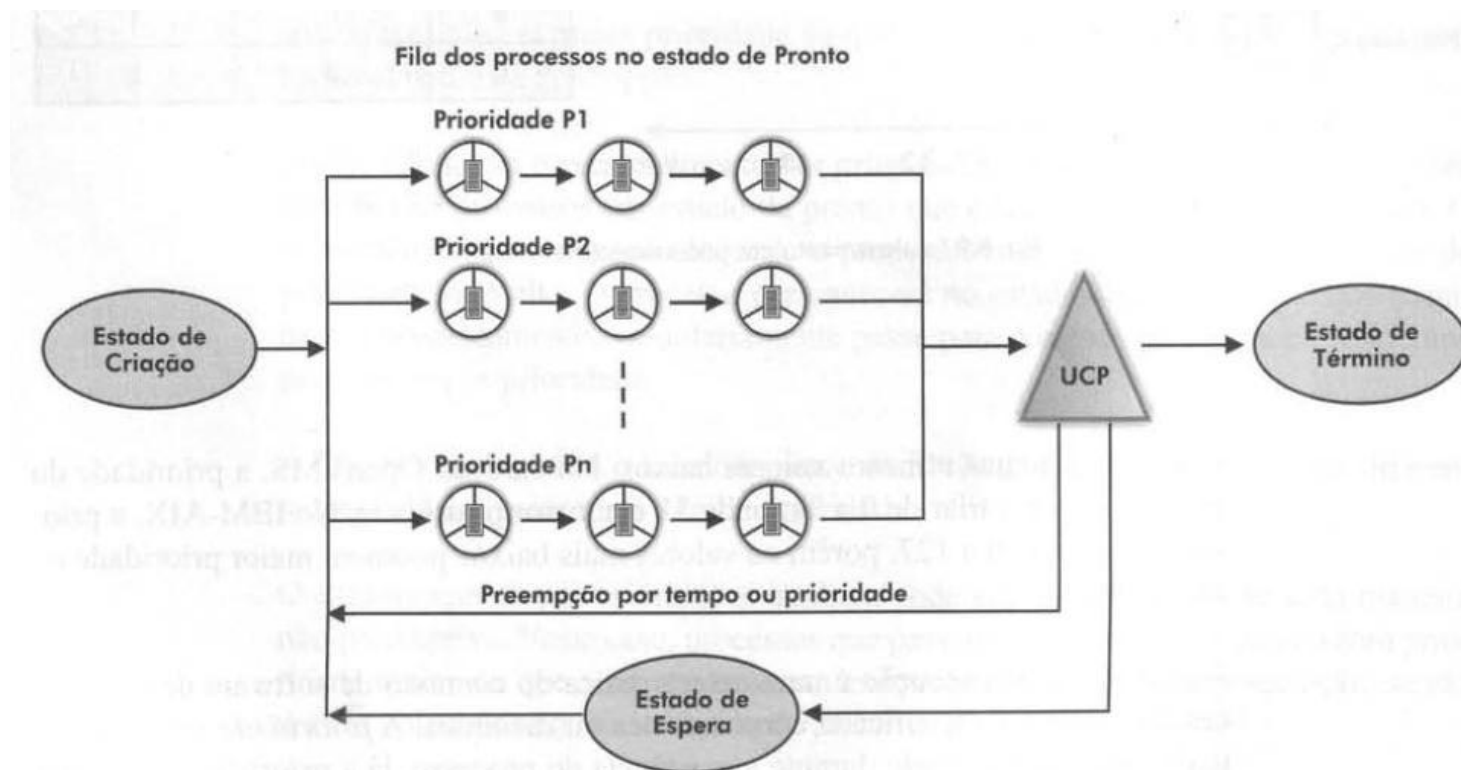
A prioridade de execução é uma característica do contexto de software de um processo e pode ser *estática* ou *dinâmica*.

Problema: *starvation* (resolve com a técnica do *aging*)

Sistemas de tempo real.

8.10 Circular com Prioridades

Implementa o conceito de fatia de tempo e de prioridade de execução associada a cada processo.



8.13 Sistemas de tempo compartilhado

Processamento interativo.

Atualmente a maioria dos SOs de tempo compartilhado utiliza o escalonamento circular (RR) com prioridades dinâmicas.

8.14 Sistemas de tempo real

Respostas imediatas.

Controle de processos como controle de tráfego aéreo são aplicações de tempo real.

Escalonamento por prioridade (estática).