

Gerência de Processador

Introdução

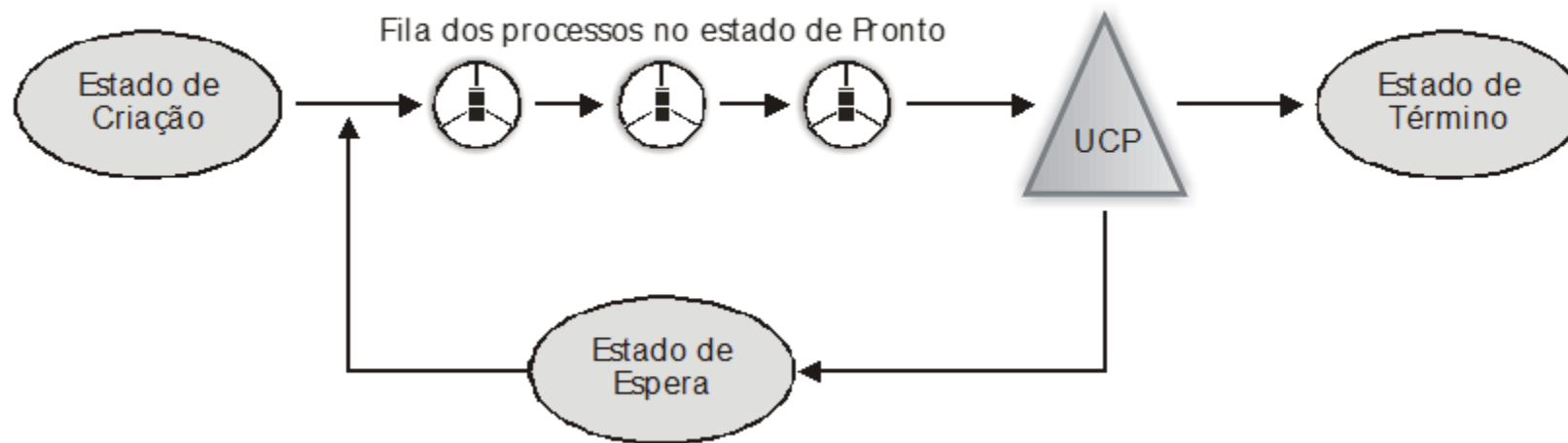
- Sistema multiprogramáveis: uso compartilhado do processador.
- **Política de escalonamento:** Critérios utilizados para determinar o processo a utilizar o processador.
- Objetivos da política de escalonamento:
 - Manter o processador ocupado a maior parte do tempo;
 - Balancear o uso do processador entre os processos;
 - Privilegiar a execução de aplicações críticas;
 - Maximizar o **número de processos executados** em um período (Throughput);
 - Minimizar o **tempo médio de espera e de turnaround** dos processos;
 - Oferecer **tempos de respostas** aceitáveis.
- Esses objetivos podem ser conflitantes...

Tipos de Escalonamento quanto a Preempção

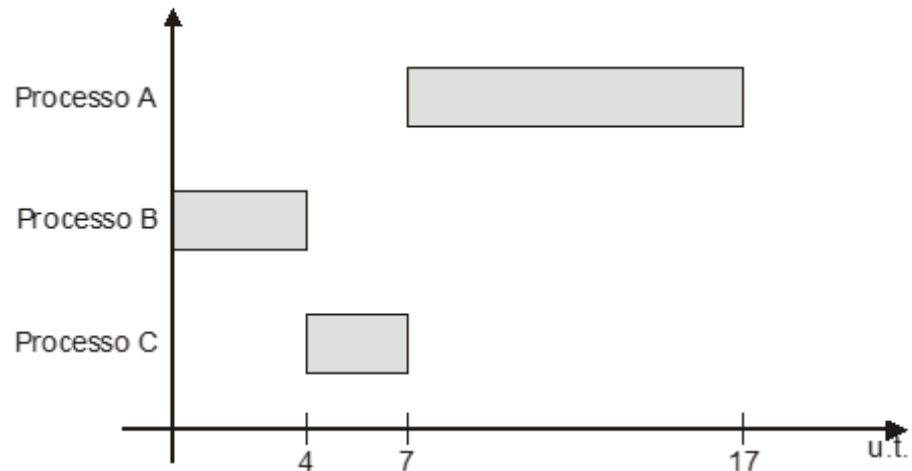
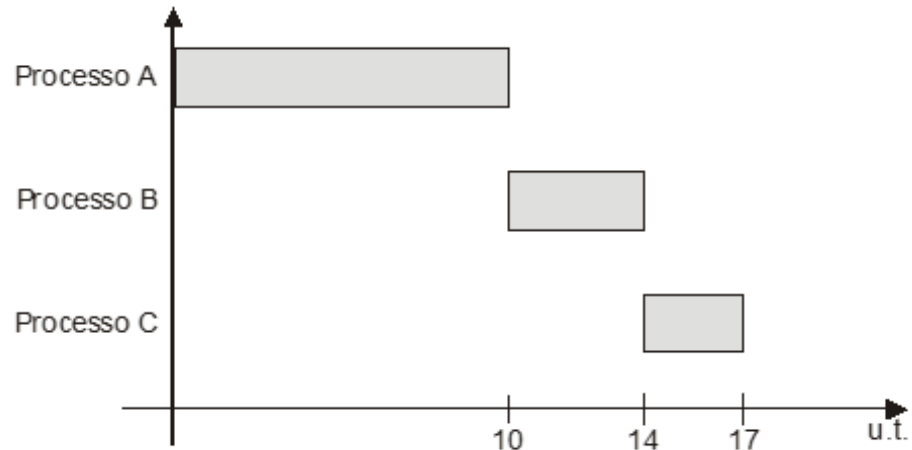
- Escalonamento não-preemptivo
 - Quando um processo está em execução nenhum evento externo pode interromper sua execução;
 - Processo só deixará a CPU caso sua execução acabe, sua fatia de tempo acabe ou execute um código que ocasione mudança para estado de espera (por ex. solicitação de um recurso não disponível);
 - Primeiro tipo de escalonamento implementado.
- Escalonamento Preemptivo:
 - O SO pode, a qualquer momento, interromper a execução de um processo e passa-lo para o estado de pronto a fim de alocar outro processo.
 - Necessário para aplicações de tempo real;
 - Permite distribuir de forma mais balanceada o uso do processador;
 - Apesar de mais complexo, é utilizado na maioria dos sistemas operacionais.

Escalonamento First-In-First-Out (FIFO)

- Política não-preemptiva bem simples;
- Baseado apenas no conceito de fila;
- Processos que passam ao estado de pronto entram no fim de um fila da qual o primeiro elemento é escalonado.



Escalonamento First-In-First-Out (FIFO)



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

Consideremos que os 3 processos foram CRIADOS ao mesmo tempo.

Então, para cada caso:

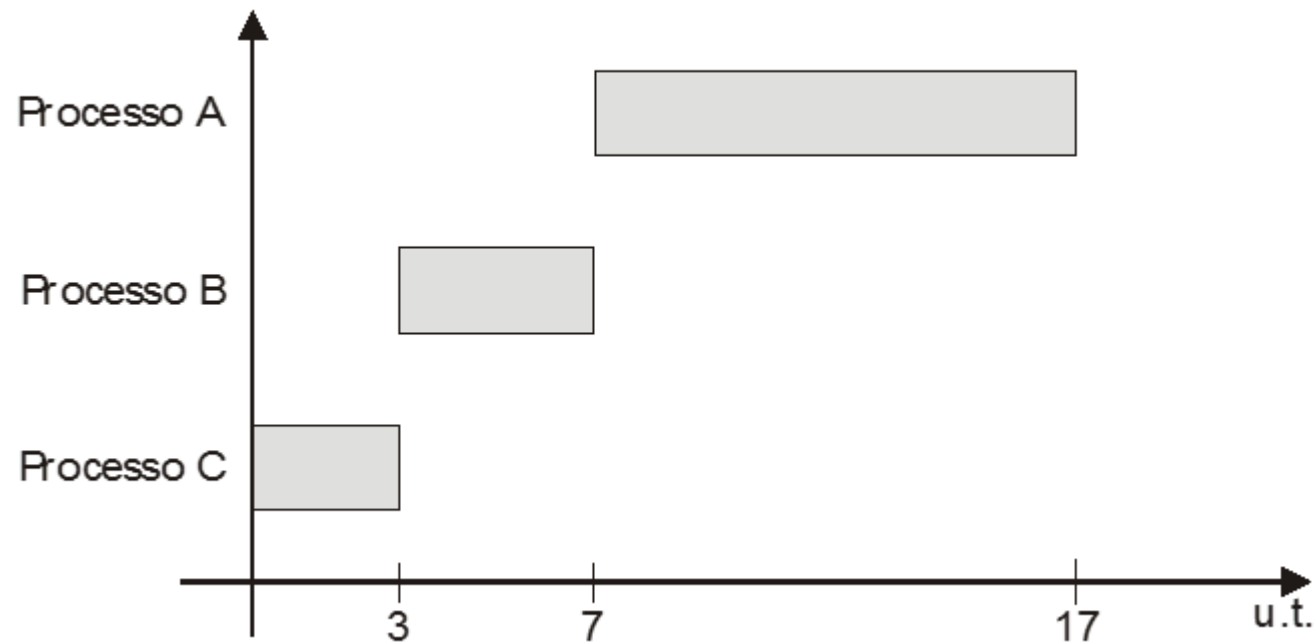
1. Qual o Tempo Médio de Espera que os processos experimentam?
2. Qual o Tempo de Turnaround Médio?

Escalonamento First-In-First-Out (FIFO)

- Principais limitações:
 - Impossível prever quando um processo terá sua execução iniciada (depende do tempo de execução de todos a sua frente na fila).
 - Situação se complica para processos I/O-bound (quando vão para a fila de espera, na volta tem de esperar a execução dos processos que “passaram na frente”).

Escalonamento Shortest-Job-First (SJF)

- Processo em estado de pronto que necessitar de menos tempo de processamento para terminar é selecionado.
- Vamos calcular o tempo de espera...



Escalonamento Shortest-Job-First (SJF)

- Uma estimativa do tempo de processador necessário é atribuída a cada processo com base em análises de execuções passadas;
- Se o tempo estimado for muito inferior ao tempo real, o sistema pode interromper a execução.
- Problema: estimar tempo de execução de processos interativos (entrada de dado é uma ação de tempo imprevisível).
- Possível solução: utilizar como base uma estimativa de tempo da próxima utilização da CPU ao invés da estimativa de tempo total de execução

Escalonamento Shortest-Job-First (SJF)

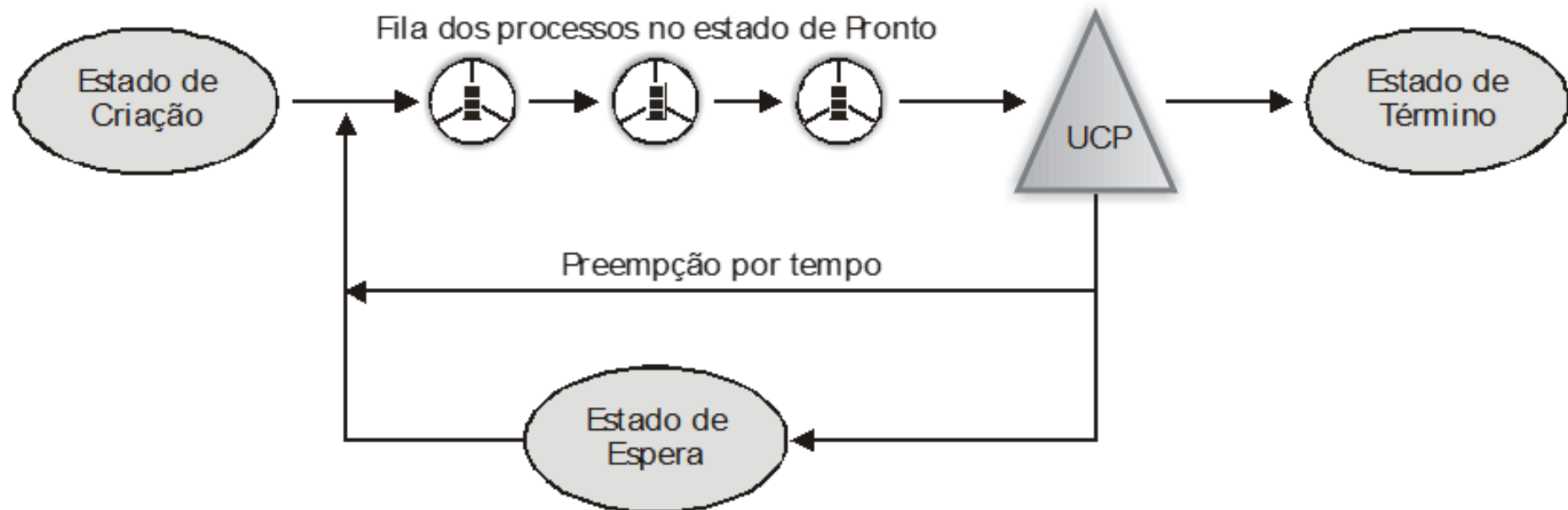
- Risco de starvation: processo “nunca” ser executado por sempre ter algum com estimativa menor.
- Melhor tempo médio de espera que o FIFO.
- Originalmente não-preemptivo.
- Versão preemptiva conhecida como ***shortest-remaining-time (SRT)***: quando há em estado de pronto um processo com tempo de processador estimado menor que o processo em execução uma preempção é executada.

Escalonamento Cooperativo

- Utilizado nos primeiros Windows (conhecido como multitarefa cooperativa);
- O sistema operacional não realiza preempção mas cada programa é responsável por monitorar a fila de mensagens para saber se deve deixar o processador;
- Se os programas não verificarem as mensagens como esperado, funciona como um não-preemptivo comum.

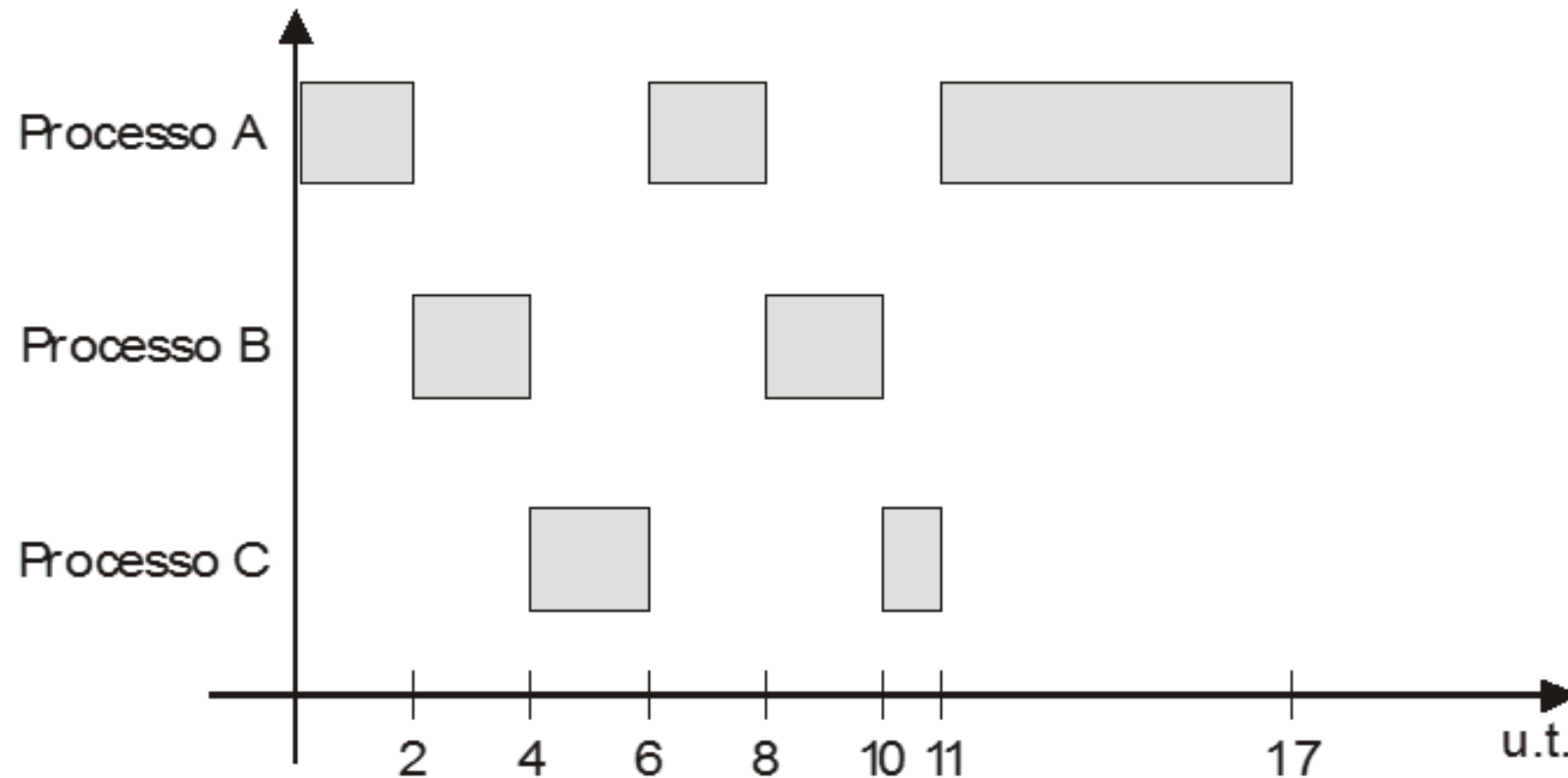
Escalonamento Circular

- Preemptivo: semelhante ao FIFO, mas define uma fatia de tempo de CPU (time slice) tal que passado esse tempo é gerada uma preempção por tempo e é feito um escalonamento.



Escalonamento Circular

- Exemplo de escalonamento circular, desconsiderando o tempo de troca de contexto.



Escalonamento Circular

- O valor da fatia de tempo afeta consideravelmente o desempenho:
 - Valor pequeno: excessivas mudanças de contexto;
 - Valor alto: tende a ter comportamento semelhante ao FIFO.
- Principal vantagem: define um tempo máximo alocado continuamente a um processo;
- Processos CPU-bound tendem a usar toda a fatia de tempo enquanto os I/O-bound tendem a deixar a CPU antes de sofrerem preempção por tempo.

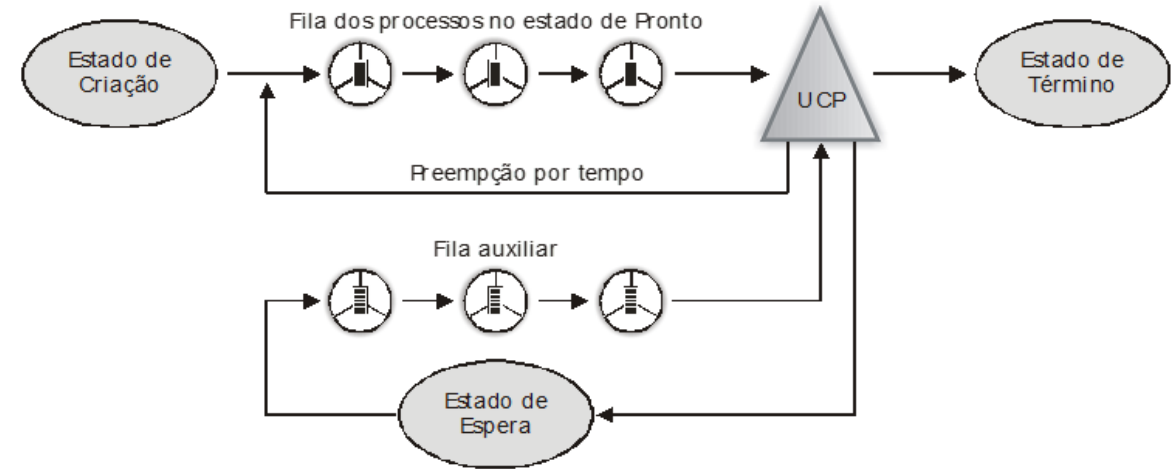
Exercícios

- Para a realização desta atividade, desconsidere o tempo gasto pelo SO nas trocas de contextos;
- Considere que todos os processos deste exercício foram CRIADOS ao mesmo tempo;
- Para o exercício abaixo, monte os gráficos e calcule o TEMPO MÉDIO DE ESPERA e o TEMPO MÉDIO DE TURNAROUND utilizando Escalonamento FIFO e Escalonamento SJF

Processo	Tempo de CPU
A	10
B	2
C	8
D	3

Escalonamento Circular Virtual

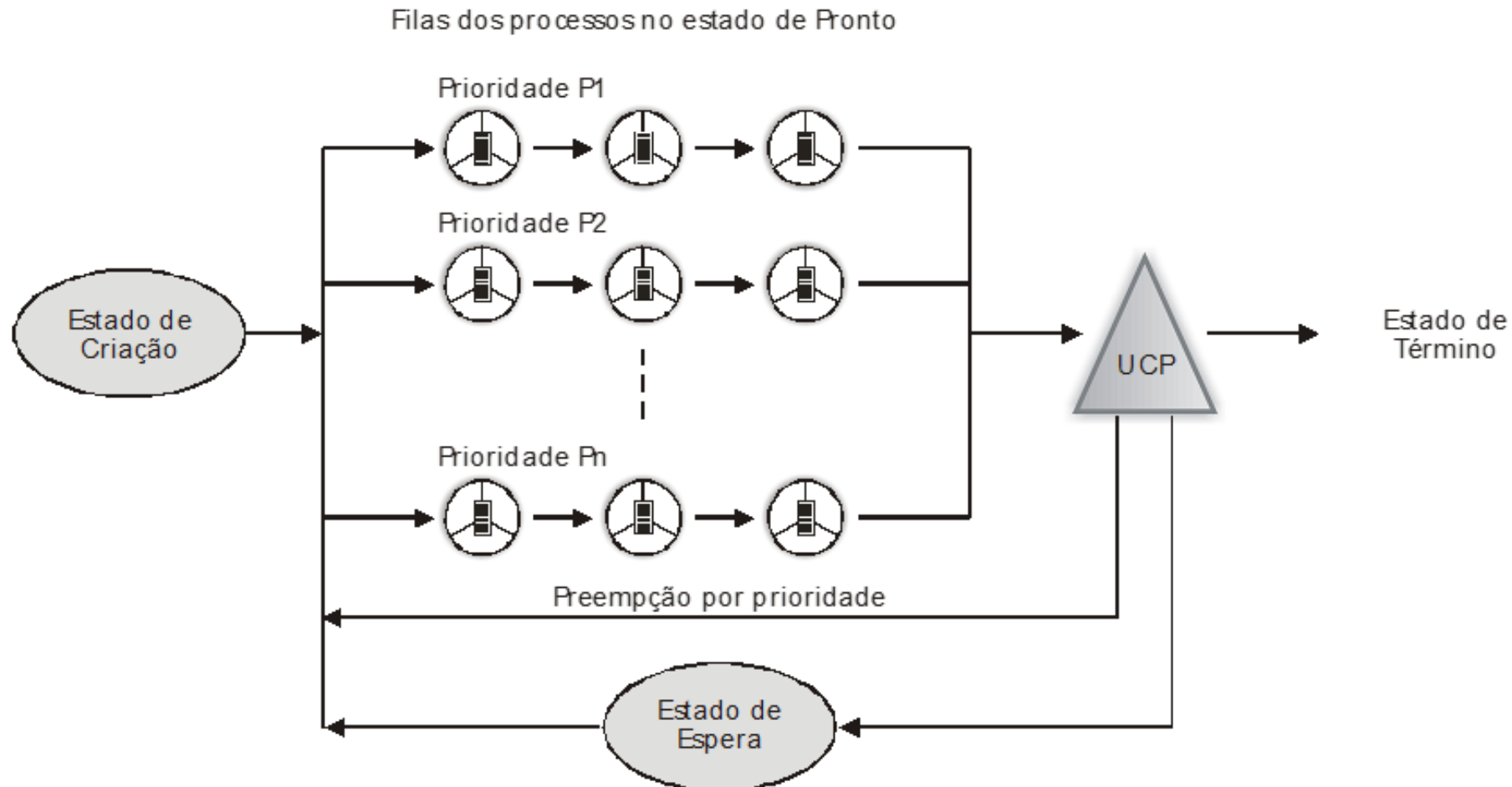
- Visa a resolver o problema dos I/O-bound:
 - Ao sair do estado de espera os processos vão para uma fila de pronto auxiliar que tem prioridade sobre a fila de pronto normal.
 - A fatia de tempo dada a um processo da fila auxiliar é o valor da fatia de tempo padrão do sistema menos o tempo de processador que o processo utilizou no último escalonamento;
 - Mais complexo porém gera melhores resultados.



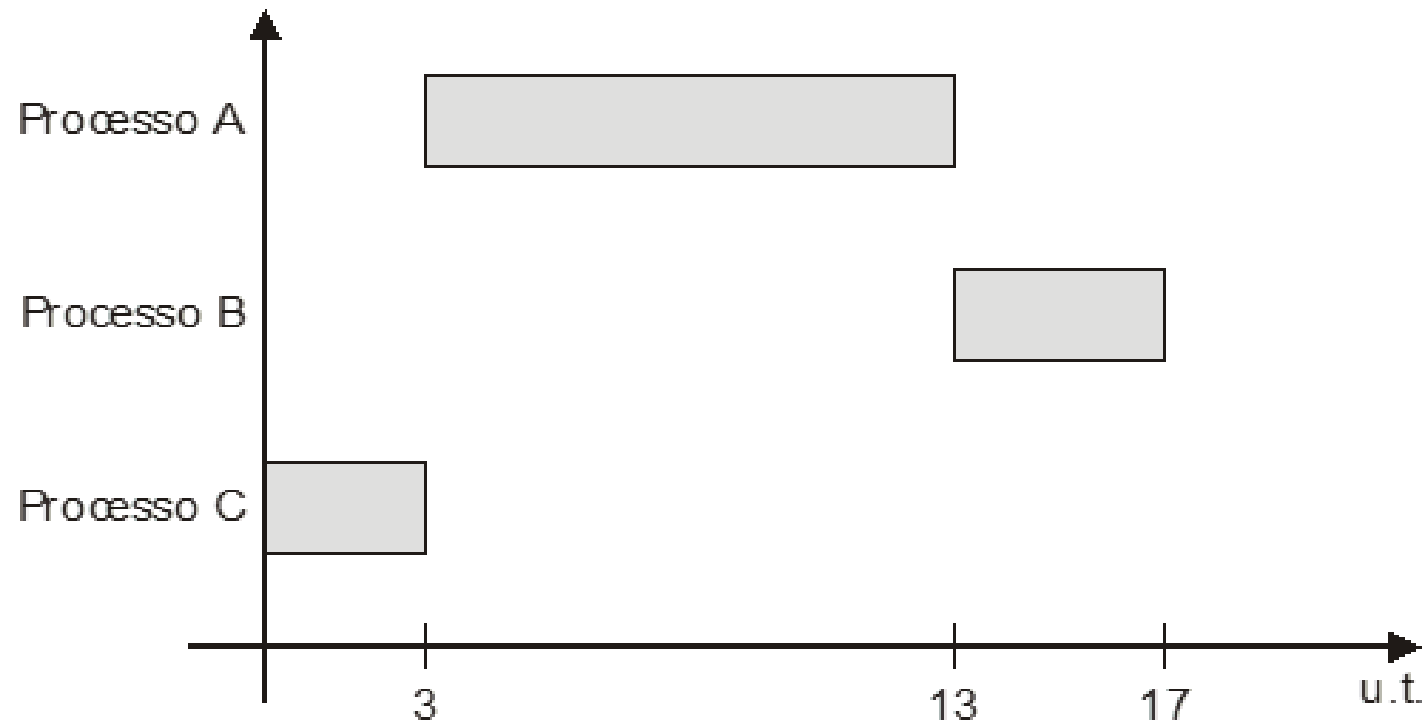
Escalonamento por Prioridades

- Escalonamento preemptivo realizado com base na prioridade de execução do processo:
 - Processo com maior prioridade na fila de pronto entra em execução;
 - Se prioridades são iguais, FIFO.
- Não há preempção por tempo.
- A cada intervalo de tempo definido as filas de pronto são avaliadas e ocorre preempção quando um processo com maior prioridade chega a uma fila de pronto (preempção por prioridade).
- Também pode ser implementado de maneira não-preemptiva: a chegada de um processo com maior prioridade não gera preempção.

Escalonamento por Prioridades



Escalonamento por Prioridades



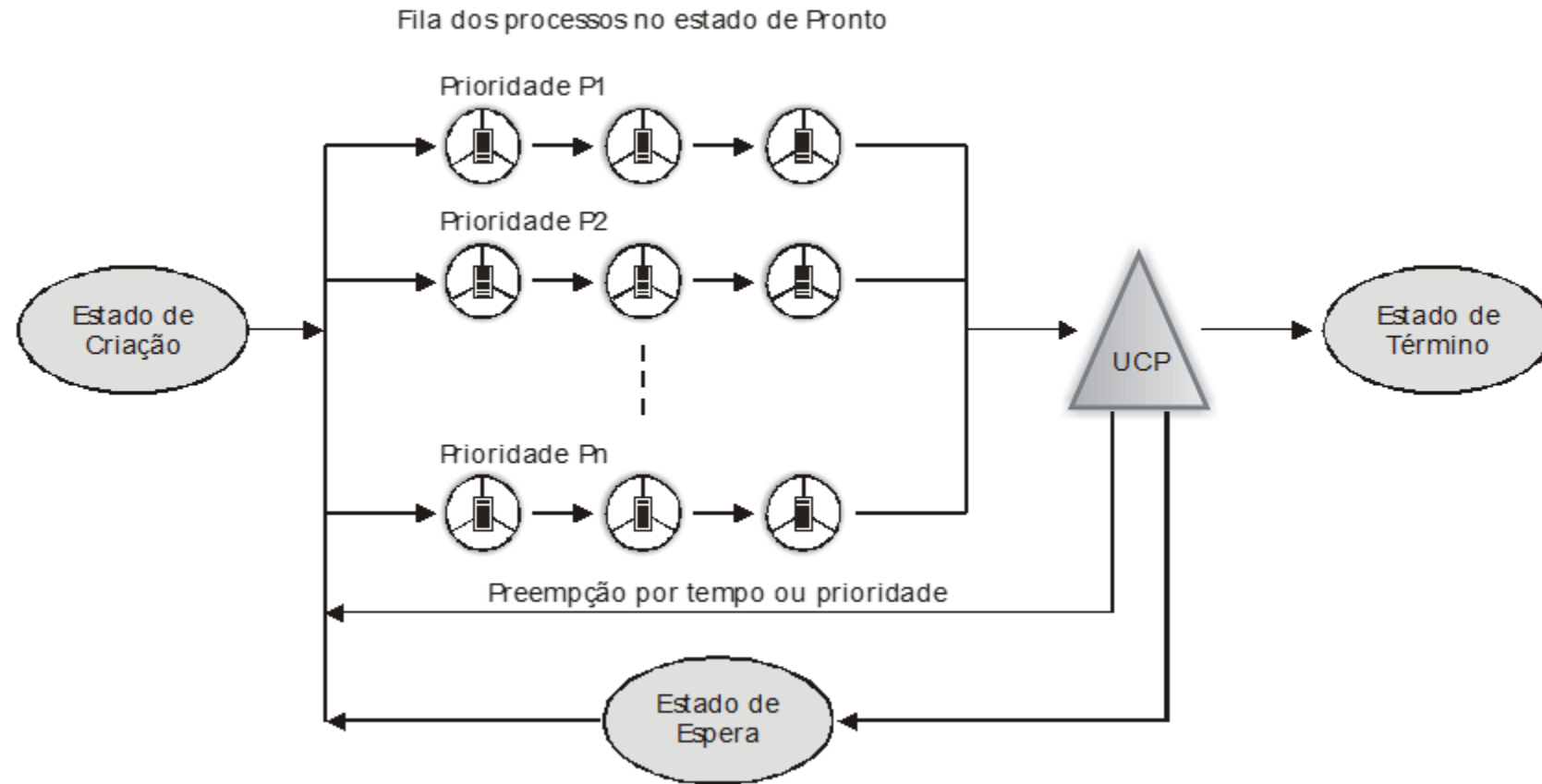
Processo	Tempo de processador (u.t.)	Prioridade
A	10	2
B	4	1
C	3	3

Escalonamento por Prioridades

- Um dos maiores problemas é o risco de *starvation* para processos de baixa prioridade.
- Técnicas de prioridade dinâmica podem ser utilizadas para evitar *starvation*.
 - Por exemplo, a técnica de *aging* que incrementa a prioridade de processos que ficam muito tempo na fila de pronto.
- O uso de prioridades é especialmente útil em sistemas de tempo real e aplicações de tempo compartilhado.

Escalonamento Circular com Prioridades

- Processo permanece em execução até que:
 - Termine seu processamento;
 - Passe para o estado de espera;
 - Sofra uma preempção por tempo;
 - Ou sofra preempção por prioridade.

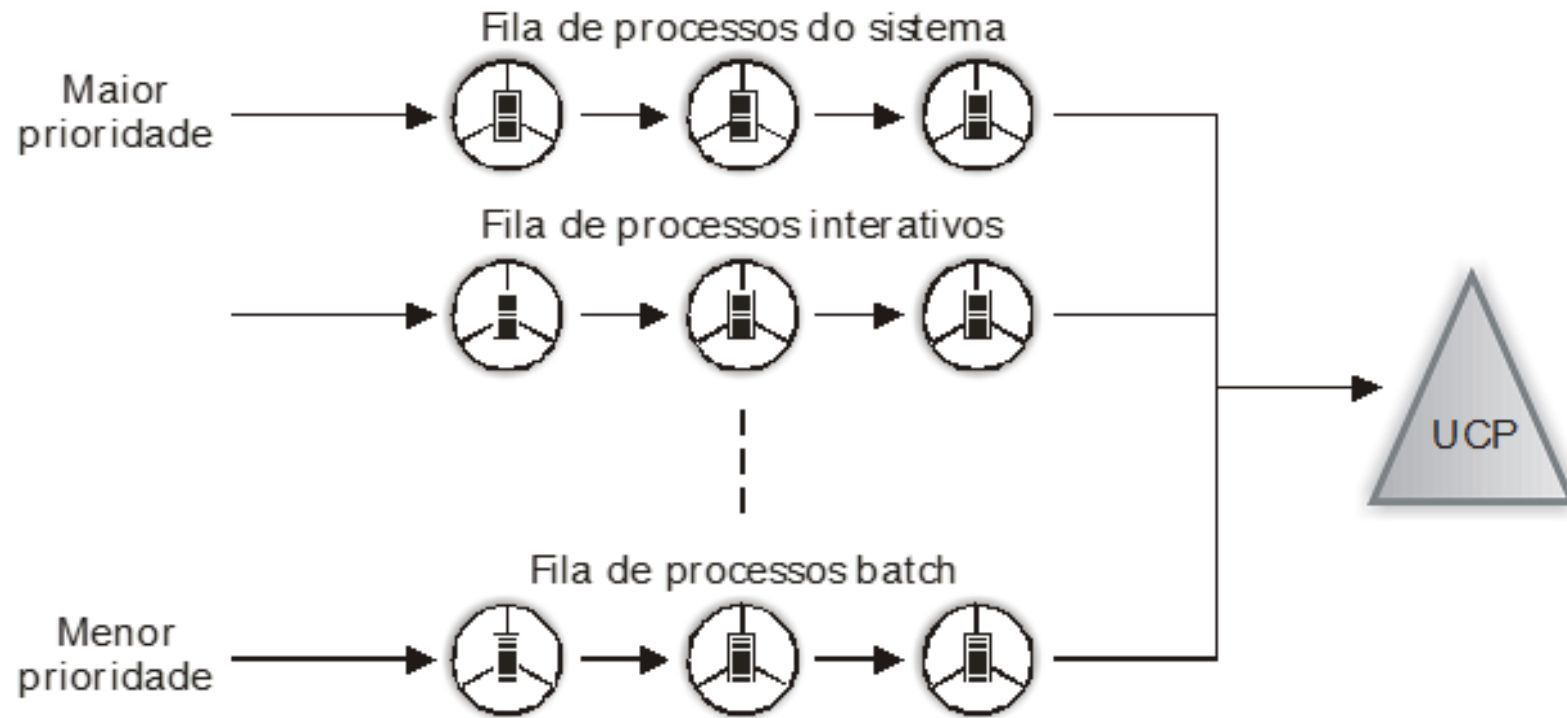


Escalonamento Circular com Prioridades

- Processos I/O-bound devem receber prioridades maiores que os CPU-bound de forma a compensar o menor probabilidade de usar toda sua fatia de tempo.
- Duas variações:
 - Escalonamento Circular com Prioridades Estáticas
 - Escalonamento Circular com Prioridades Dinâmicas

Escalonamento por Múltiplas Filas

- Utiliza várias filas, de acordo com o tipo do processo, tal que cada fila utiliza o mecanismo de escalonamento mais adequado.
- O processo não possui prioridade, ficando essa característica associada a fila.



Bibliografia

- Capítulo 8 do livro “Arquitetura de Sistemas Operacionais”, 4ª edição, de Francis Berenger Machado e Luiz Paulo Maia.