

Estratégias de Escalonamento de Processos

Rossano Pablo Pinto, MSc.
FATEC - Americana
15/02/2008

Escalonamento de Processos

- O que é um sistema multiprogramável?
 - compartilha a CPU entre diversos processos
 - é desejável que os processos obtenham tempos justos de processamento
 - é desejável que os processos obtenham serviço similar ao de sistemas mono-programáveis

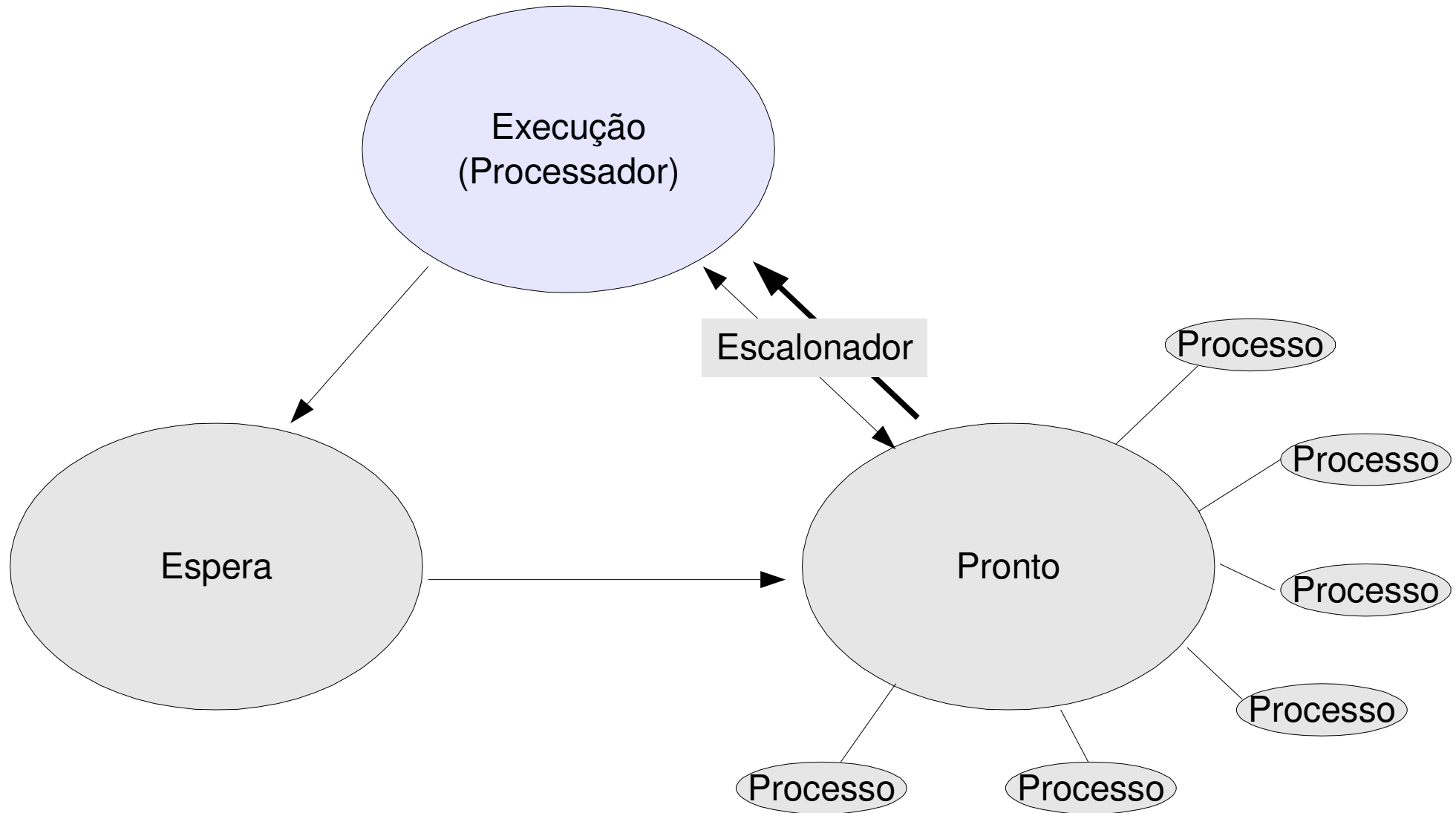
Escalonamento de Processos

- Sistemas multiprogramáveis devem ser eficientes em:
 - criar processos (ex.: Fork)
 - matar (finalizar) processos
 - bloquear processos (execução -> bloqueado ou pronto)
 - acordar processos
 - Eventualmente ESCALONAR o processo

Escalonamento de Processos

- Procedimento de escolha do próximo processo
 - fundamental
 - tenta escolher o processo mais adequado para ser executado em um dado momento
 - ESCALONADOR (scheduler): parte do código do SO que escolhe o próximo processo a ser executado

Escalonamento de Processos



Escalonamento de Processos

- Objetivos do ESCALONADOR
 - manter CPU perto de 100% de uso
 - balanceamento da CPU p/ vários processos
 - maximizar vazão (throughput)
 - resposta rápida a usuários interativos
 - evitar “starvation”

Escalonamento de Processos

- Objetivos conflitantes - Resolução de conflito
 - Determinar o tipo do sistema (tipo da maioria dos processos)
 - lote (batch)
 - interativo
 - CPU-bound
 - I/O-bound
 - tempo-real
 - tempo-compartilhado (vários usuários)

Para cada tipo de sistema
deve haver uma
POLÍTICA adequada

Escalonamento de Processos

- Critérios de escalonamento

- depende do tipo de processamento:

- | | |
|--|--|
| • Fairness | <u>[Sistemas Multiprogramáveis]</u> |
| • UCPU (utilização de CPU perto de 100%) | |
| • Vazão (Throughput) | |

- | | |
|--------------|--------------------------------|
| • Turnaround | <u>[Sistemas Batch]</u> |
|--------------|--------------------------------|

- | | |
|---------------------|--|
| • Tempo de resposta | <u>[Sistemas on-line - interativos]</u> |
|---------------------|--|

- | | |
|------------------|--|
| • Meet deadlines | <u>[Sistemas de tempo real]</u> |
|------------------|--|

Escalonamento de Processos

- Fairness
 - Todos os processos devem ter o mesmo tipo de tratamento
- UCPU
 - manter CPU perto de 100%
 - Ex.: 30% indica sistema com carga proc. baixa
 - Ex.: 90% carga processamento alta

Escalonamento de Processos

- Vazão
 - Representa número de processos finalizados em um determinado intervalo
 - 50 processos por hora é melhor do que 40
- Tempo de turnaround
 - tempo da admissão ao término de um processo
 - contabiliza todos os tempos administrativos (alocação de memória, espera em filas, E/S, etc..)

Escalonamento de Processos

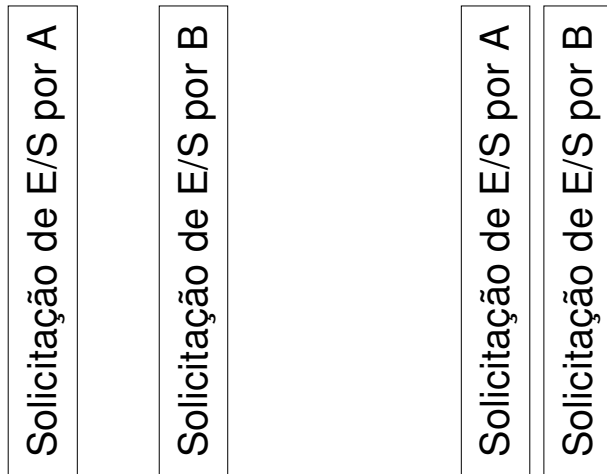
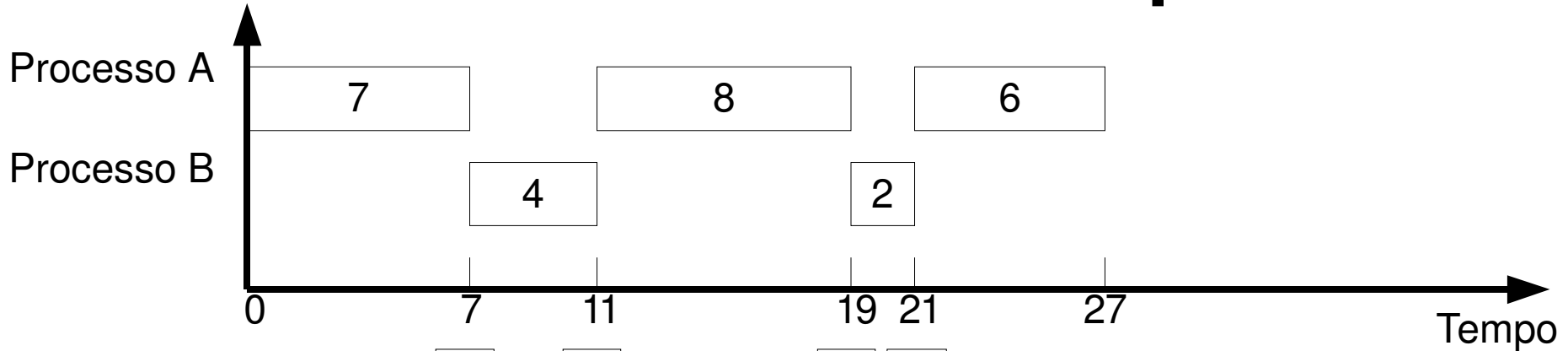
- Tempo de resposta
 - tempo de resposta decorrido do momento da submissão de um pedido até a primeira resposta
 - sistemas interativos exigem que seja baixo

Escalonamento de Processos

- Escalonamento não-preemptivo
 - primeiros sistemas multiprogramáveis
 - quando processo era escolhido, este era executado até que o “próprio” escolhesse abrir mão da CPU
 - Exemplos de escalonamentos nestes sistemas:
 - FIFO
 - Shortest job first
 - cooperativo (Ex. win 3.1x, MAC OS 9 e anteriores)

Escalonamento de Processos

FIFO - Não-Preemptivo



- Utilizado em sistemas batch
- Sistemas Multiprog. utiliza variações deste

	Tempo de CPU (u.t.)	Característica do processo
Processo A	21	CPU-bound
Processo B	6	I/O-bound

Escalonamento de Processos

Shortest Job First - Não-Preemptivo

- Associa processo ao tempo de execução
- Escolhe os de tempo menor primeiro
- Prioriza **vazão** e **turnaround**, e reduz tempo médio de espera. Ex. turnaround (p,t):
 - A8, B3, C5, D2, E1 -> A8, B11, C16, D18, E19
 - E1, D2, B3, C5, A8 -> E1, D3, B6, C11, A19
- Problema: (pré) determinar os tempos

Escalonamento de Processos

Cooperativo - Não-Preemptivo

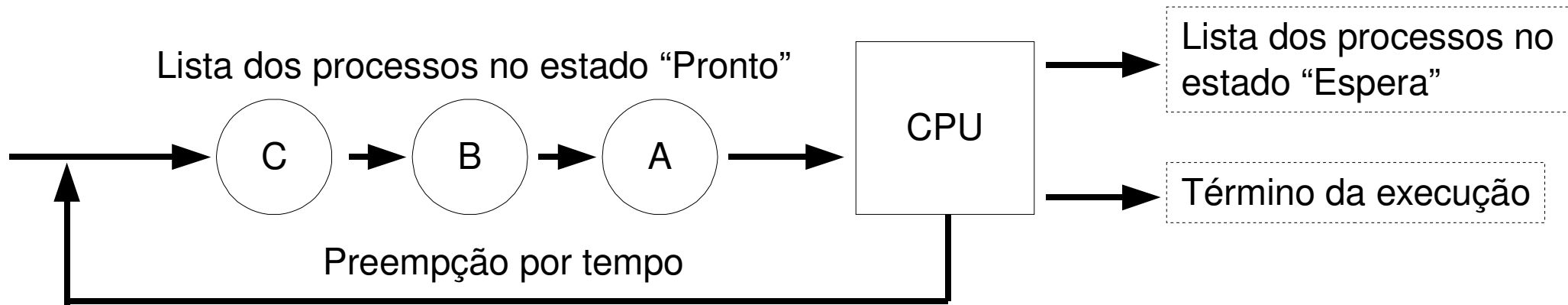
- Processo voluntariamente libera CPU (volta p/ fila de pronto)
- Não existe intervenção do SO
- Win 3.x - multitarefa cooperativa
 - processo verifica periodicamente fila de interesse
- Problemas óbvios: loops infinitos, erros diversos **TRAVAM TODO O SISTEMA**

Escalonamento de Processos

- Escalonamento Preemptivo
 - Sistema interrompe processo (tempo ou prioridade)
 - Compartilhamento mais uniforme da CPU
 - Troca de um processo pelo outro
 - Troca de contexto
 - Trocas de contexto não devem ocorrer em excesso (carga administrativa do sistema deve ser baixa)

Escalonamento de Processos

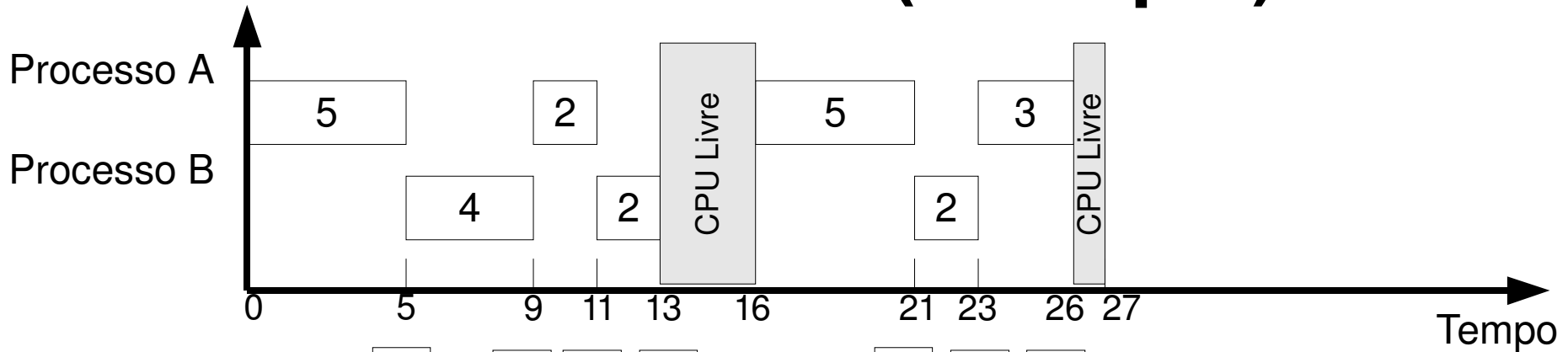
CIRCULAR (Round Robin)



Conceito de timeslice (fatia de tempo) / quantum
Entre 5 e 800 ms no LINUX (100ms default)

Escalonamento de Processos

CIRCULAR (Exemplo)



Fim timeslice de A

Solicitação de E/S por B

Solicitação de E/S por A

Solicitação de E/S por B

Fim timeslice de A

Solicitação de E/S por B

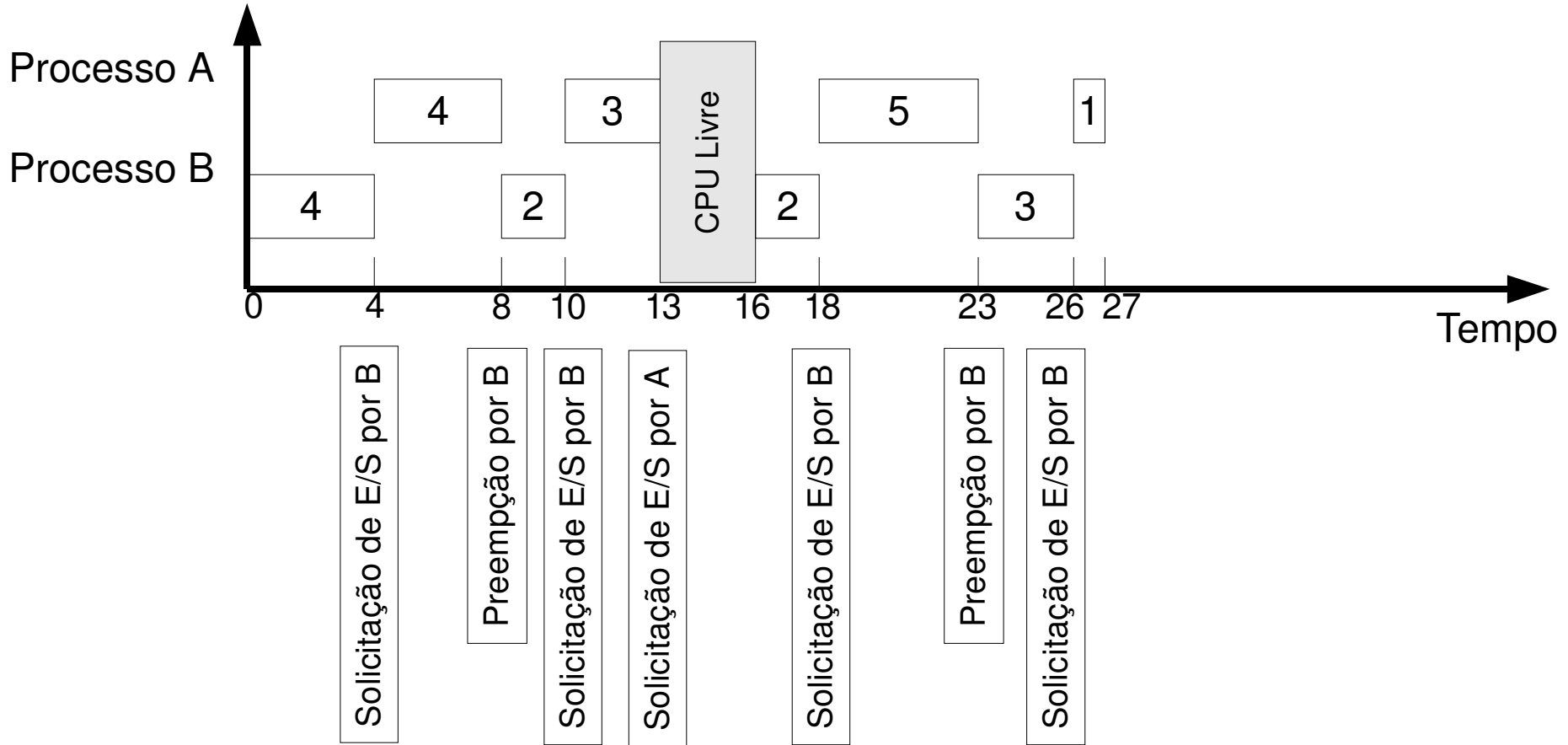
Solicitação de E/S por A

timeslice = 5 u.t.

	Tempo de CPU (u.t.)	Característica do processo
Processo A	15	CPU-bound
Processo B	8	I/O-bound

Escalonamento de Processos

PRIORIDADES



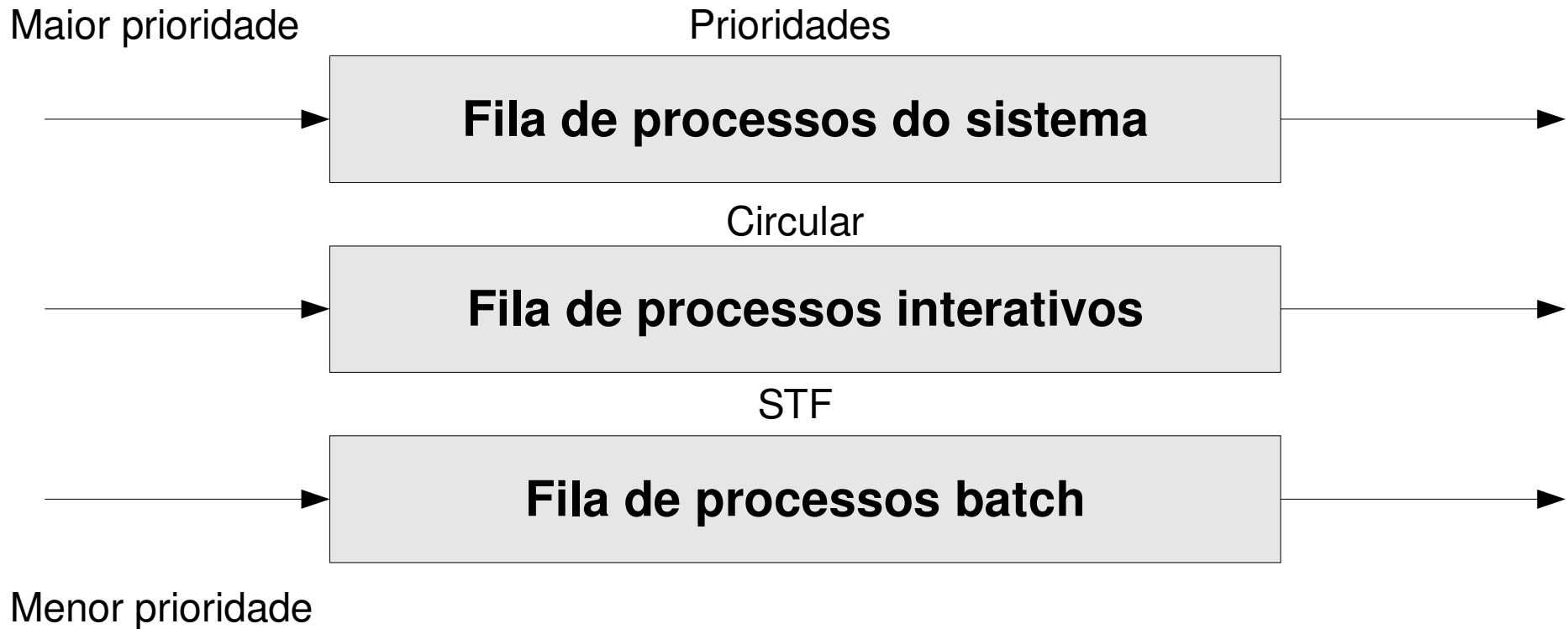
	Tempo de CPU (u.t.)	Característica do processo	Prioridade
Processo A	13	CPU-bound	4
Processo B	11	I/O-bound	7

Escalonamento de Processos

- Prioridade
 - Estática
 - Dinâmica
 - processos I/O bound recebem + prioridade a cada saída da fila de processos bloqueados

Escalonamento de Processos

MÚLTIPLAS FILAS



- Ex.: 3 filas (níveis) com prioridades diferentes
- Cada file adota uma política de escalonamento diferente.
- Necessidade de pré-classificação dos processos
- Não prevê mudança de comportamento ao longo da execução

Escalonamento de Processos

MÚLTIPLAS FILAS COM REALIMENTAÇÃO

Maior prioridade

quantum <



Menor prioridade

quantum >

Detecção de mudança de comportamento em tempo de execução

Escalonamento de Processos

- Escalonamento de tempo-real
 - Prioridades estáticas (IMPORTANTE)
 - NÃO existe timeslice!!!
 - Processo executa até o final (RUN-TO-COMPLETITION)

Escalonamento de Processos

- Escalonamento com múltiplos processadores
 - Sistemas Fracamente Acoplados
 - cada processador faz seu escalonamento local
 - Sistemas Fortemente Acoplados
 - Exclusão mútua no código do escalonador
 - 2+ processos não podem ser escalonados p/ um mesmo processador ao mesmo tempo
 - 1 processo não pode ser escalonado p/ execução em 2+ processadores (salvo threads!)

Escalonamento de Processos

- Estratégias adotados no Linux
 - O(1) scheduler (big o scheduler)
 - O tempo de escolha do próximo processo independe do número de processos existentes no sistema
 - implementa 2 faixas separadas de prioridade
 - valores NICE (-20 0 +19): maior núm, menor prioridade
 - determina o tempo do timeslice (-19 recebe o valor máximo de timeslice)
 - prioridades de tempo-real (0 à 99)

Escalonamento de Processos

- Estratégias adotados no Linux (man sched_setscheduler)
 - #include <sched.h>
 - int sched_setscheduler(pid_t pid, int policy,
 - const struct sched_param *param);
 - int sched_getscheduler(pid_t pid);
 - struct sched_param {
 - ...
 - int sched_priority;
 - ...
 - };