

Programação Concorrente

Odorico Machado Mendizabal

Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE

Processos – Parte 2

Objetivo da aula

- Identificar os componentes de um processo e ilustrar como eles são representados e gerenciados por um SO
- Descrever o ciclo de vida de um processo e trocas de contexto
- Desenvolvimento de programas usando chamadas ao sistema adequadas
- Descrever e comparar mecanismos de comunicação entre processos

Execução de processos

Programas



Emacs

Processos

P1

P2

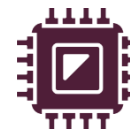
P3

P4

P5

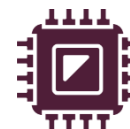
Recursos

CPU 1



...

CPU n

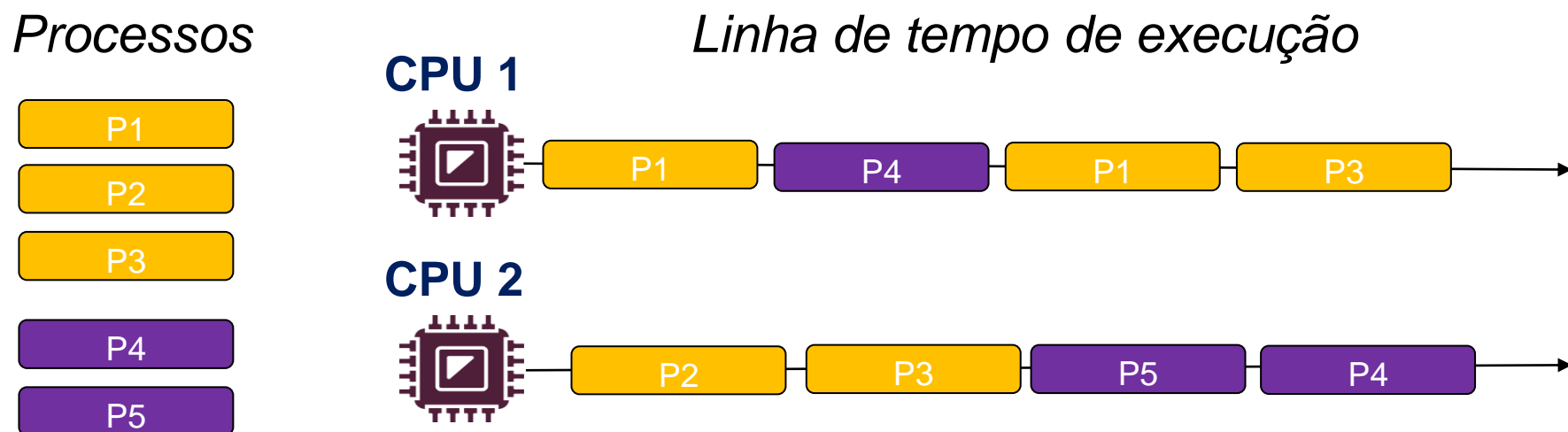


Memória
Principal

Dispositivos
de E/S

- Um núcleo de processamento executa apenas um processo a cada instante
- Um computador com n núcleos pode processar até n processos simultaneamente
- S.O. multitarefas alternam o uso de CPU entre processos. Cada processo executa por um quantum de tempo ou até que este seja interrompido

Execução de processos



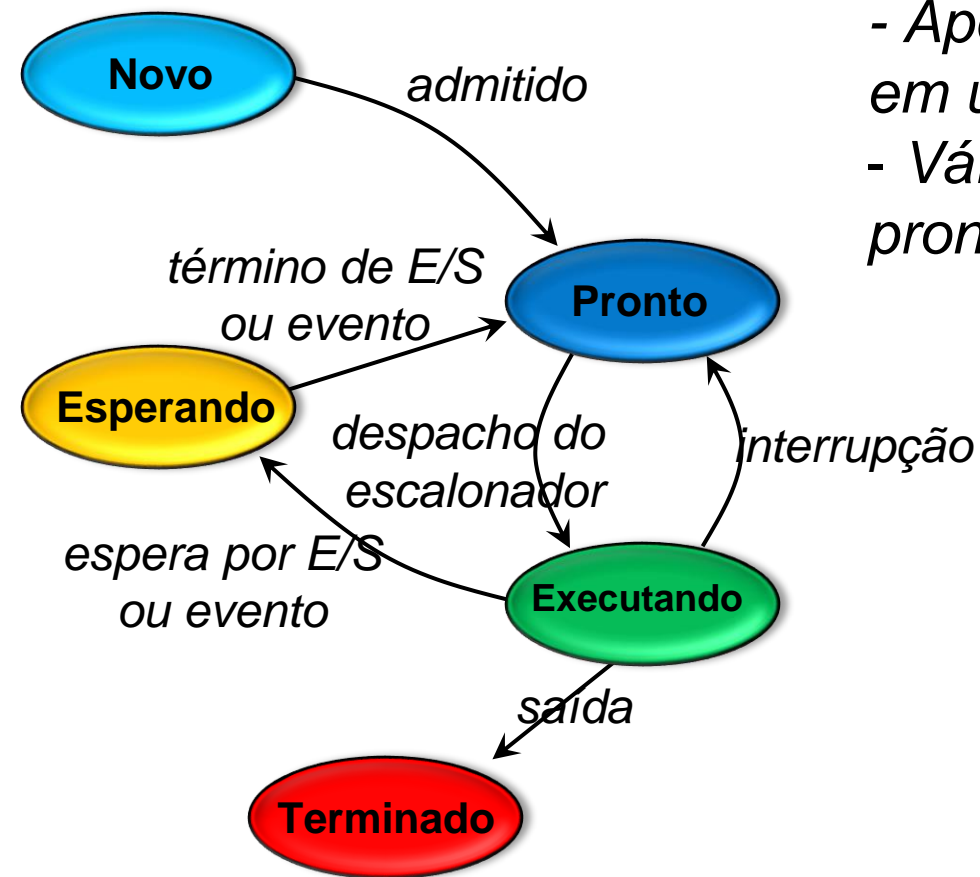
Escalonamento em sistemas operacionais multitarefa

- Em geral, adotam política de melhor esforço para atender todos os processos de maneira igualitária (*justiça*)
- Há concorrência entre processos
- Melhor uso de recursos (CPU, dispositivos de E/S)

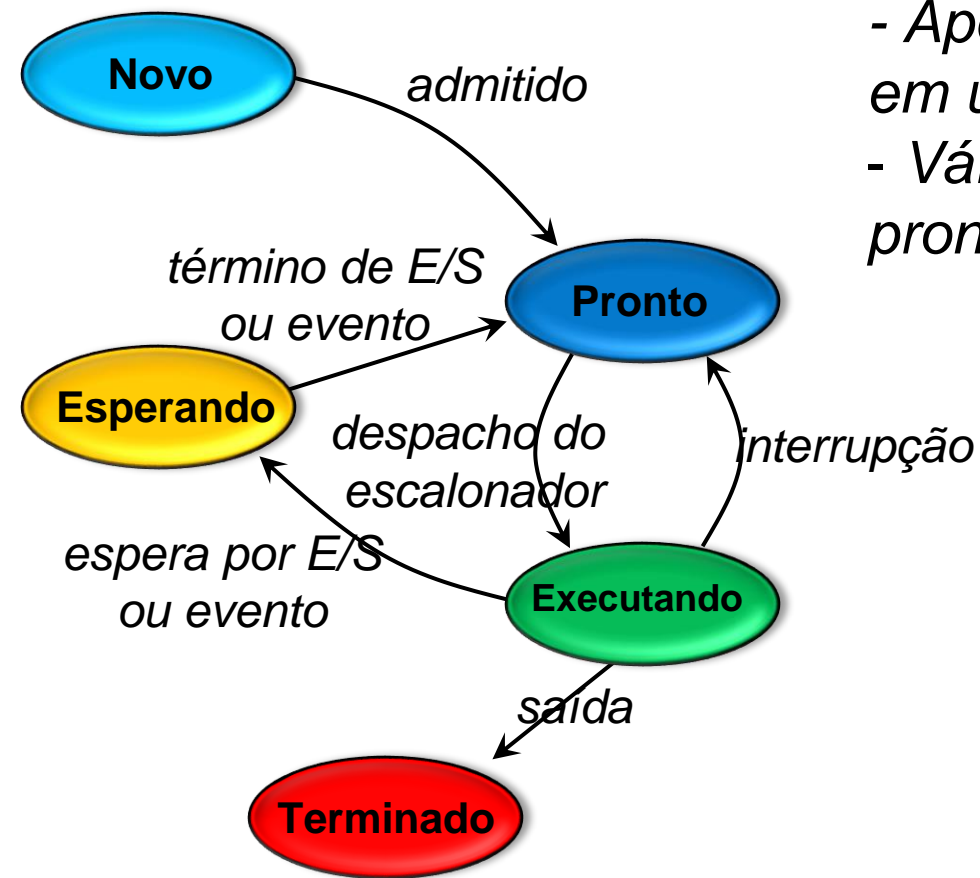
Estados do processo

Durante a execução, um processo muda de estado

- Apenas um processo pode estar executando em uma CPU
- Vários processos podem estar esperando ou prontos



Estados do processo



Durante a execução, um processo muda de estado

- *Apenas um processo pode estar executando em uma CPU*
- *Vários processos podem estar esperando ou prontos*

- **Novo:** processo foi criado

- **Pronto:** processo pronto para ser executado, mas sem o direito de usar a CPU

- **Executando:** sendo executado pela CPU

- **Esperando:** aguarda operação de E/S, liberação de algum recurso ou fim de tempo de espera

- **Terminado:** processo finalizou sua execução

Bloco de controle de processo

PCB – Process Control Block (Bloco de controle do processo)

► Contém informações associadas ao processo

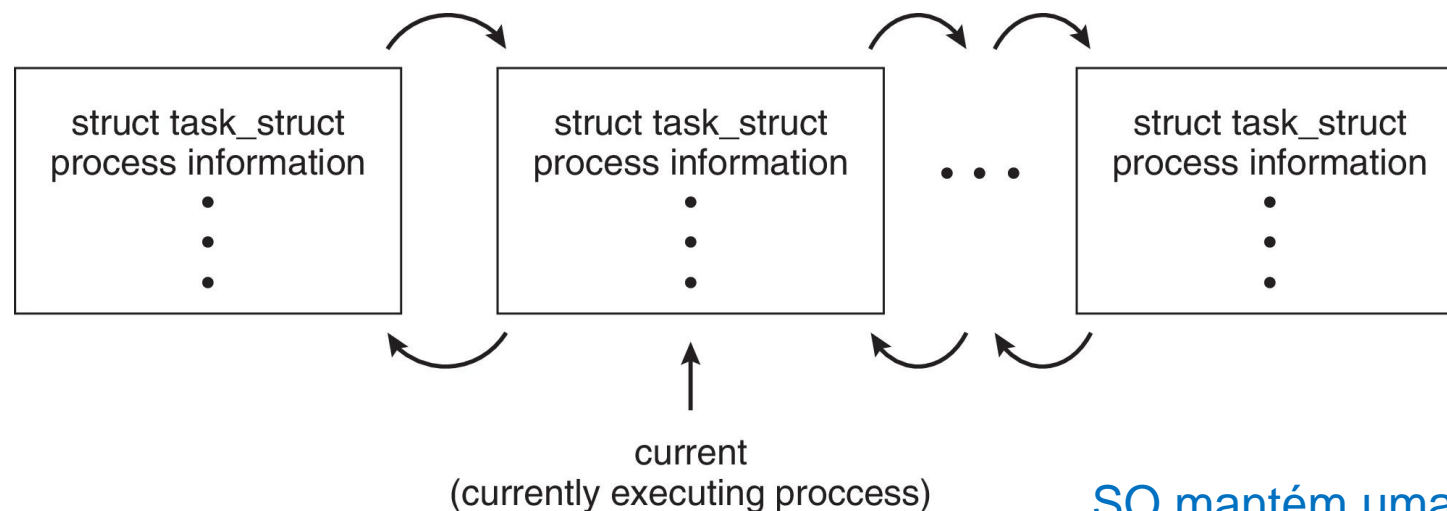
- Estado do processo
- Contador de programa (PC)
- Registradores da CPU
- Informações de escalonamento
- Informações de gerência de memória
- Informação contábil
- Informações de E/S

<i>estado do processo</i>
<i>número do processo</i>
<i>contador de programa</i>
<i>registradores</i>
<i>limites de memória</i>
<i>arquivos e descritores abertos</i>
<i>. . .</i>

Representação de um processo em Linux

Representado pela estrutura `task_struct`

```
pid t_pid;           /* identificador do processo */
long state;          /* estado do process */
unsigned int time_slice /* informação para escalonamento */
struct task_struct *parent; /* pai do processo */
struct list_head children; /* filhos do processo */
struct files_struct *files; /* lista de arquivos abertos */
struct mm_struct *mm; /* espaço de endereço do processo */
```

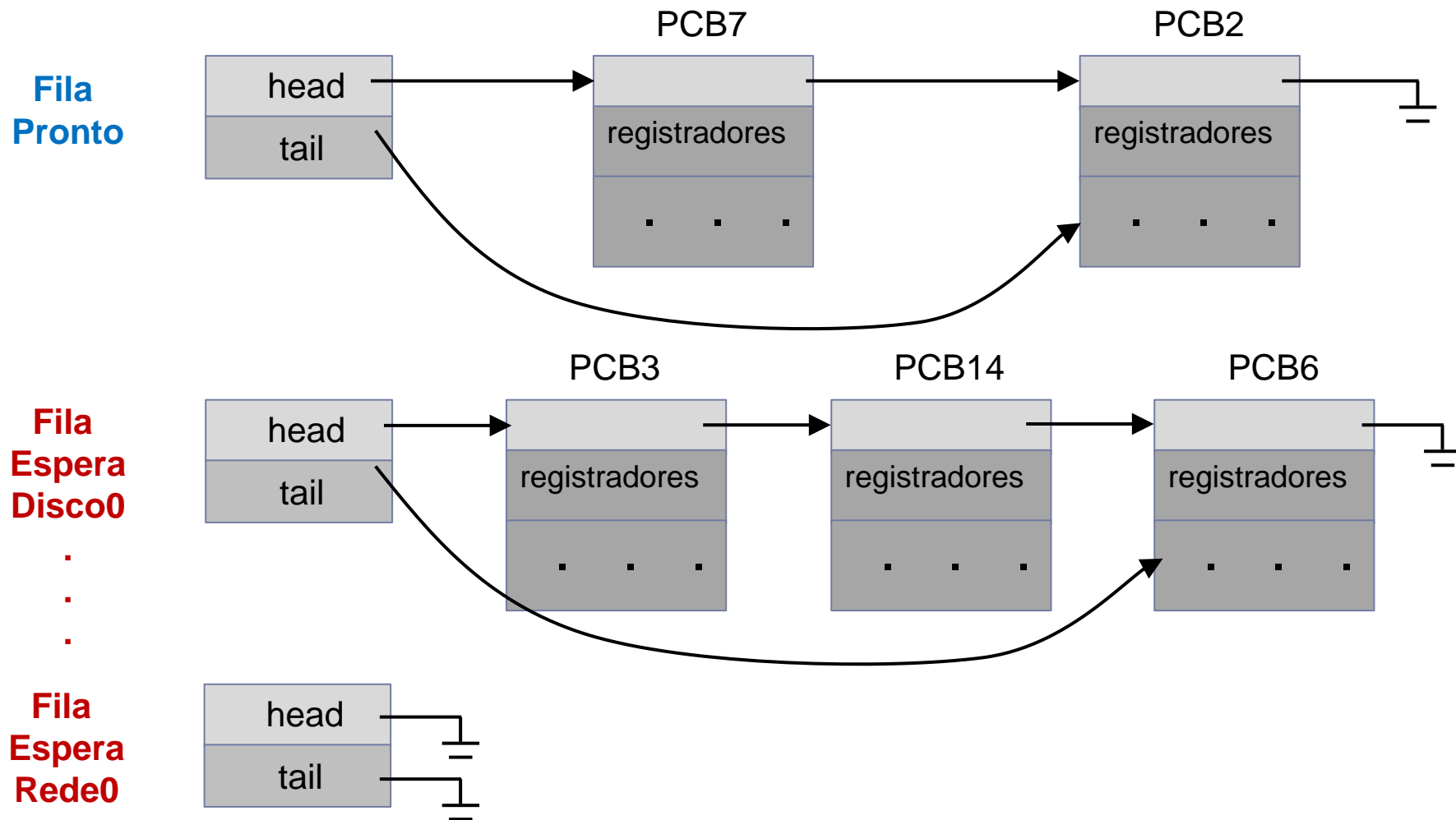


SO mantém uma lista de processos

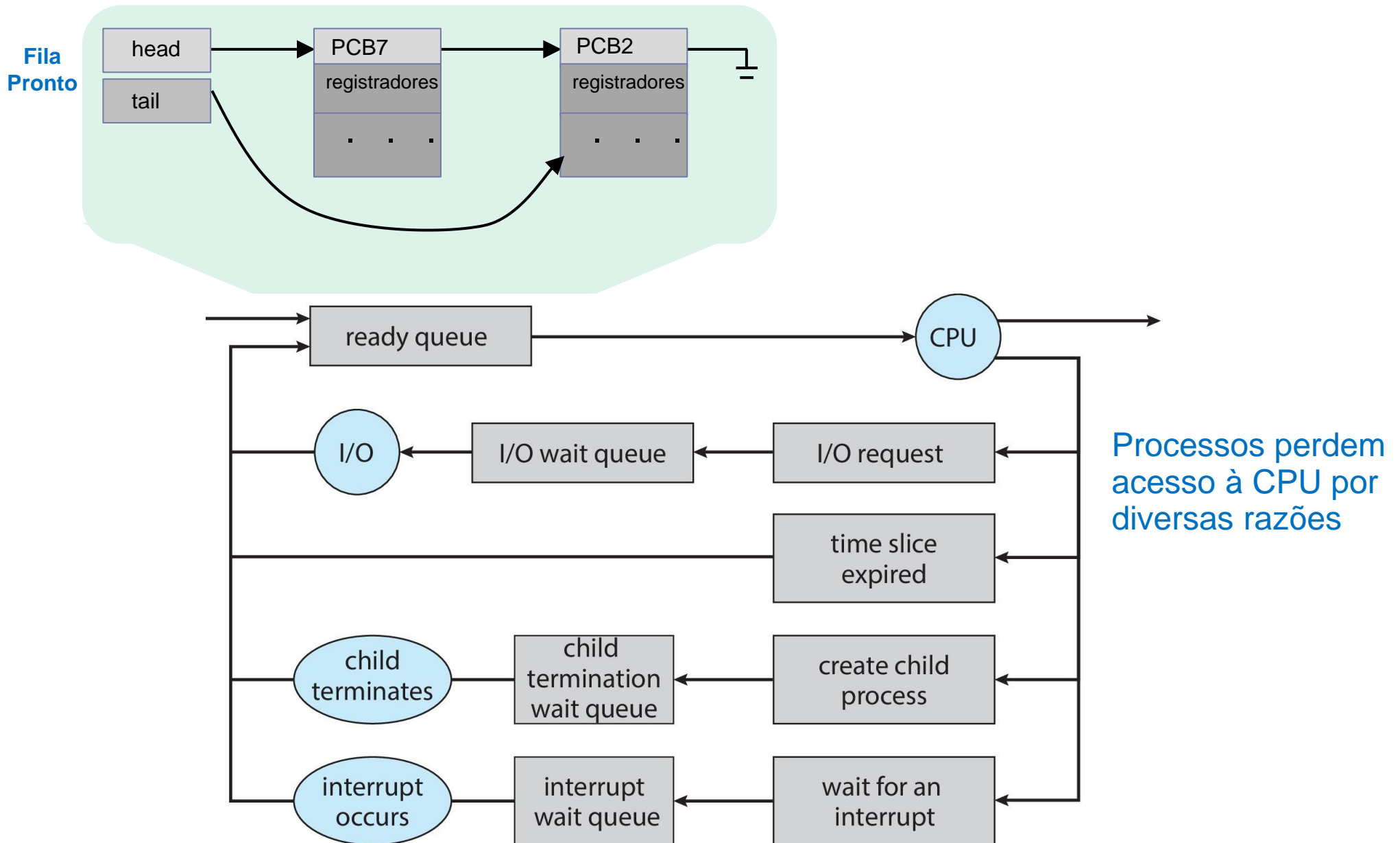
Escalonamento de processo

- Manter sempre que possível algum processo executando
 - Melhorar a utilização da CPU
- Alternar execução entre diferentes processos (justiça)
 - Processos **executando** podem ser interrompidos
 - Pelo sistema operacional (executou bastante tempo), devido a uma interrupção ou chamada de sistema
 - Processos **prontos** podem ser escolhidos para execução
 - Dado algum critério do escalonador (veremos algoritmos de escalonamento nas próximas aulas)

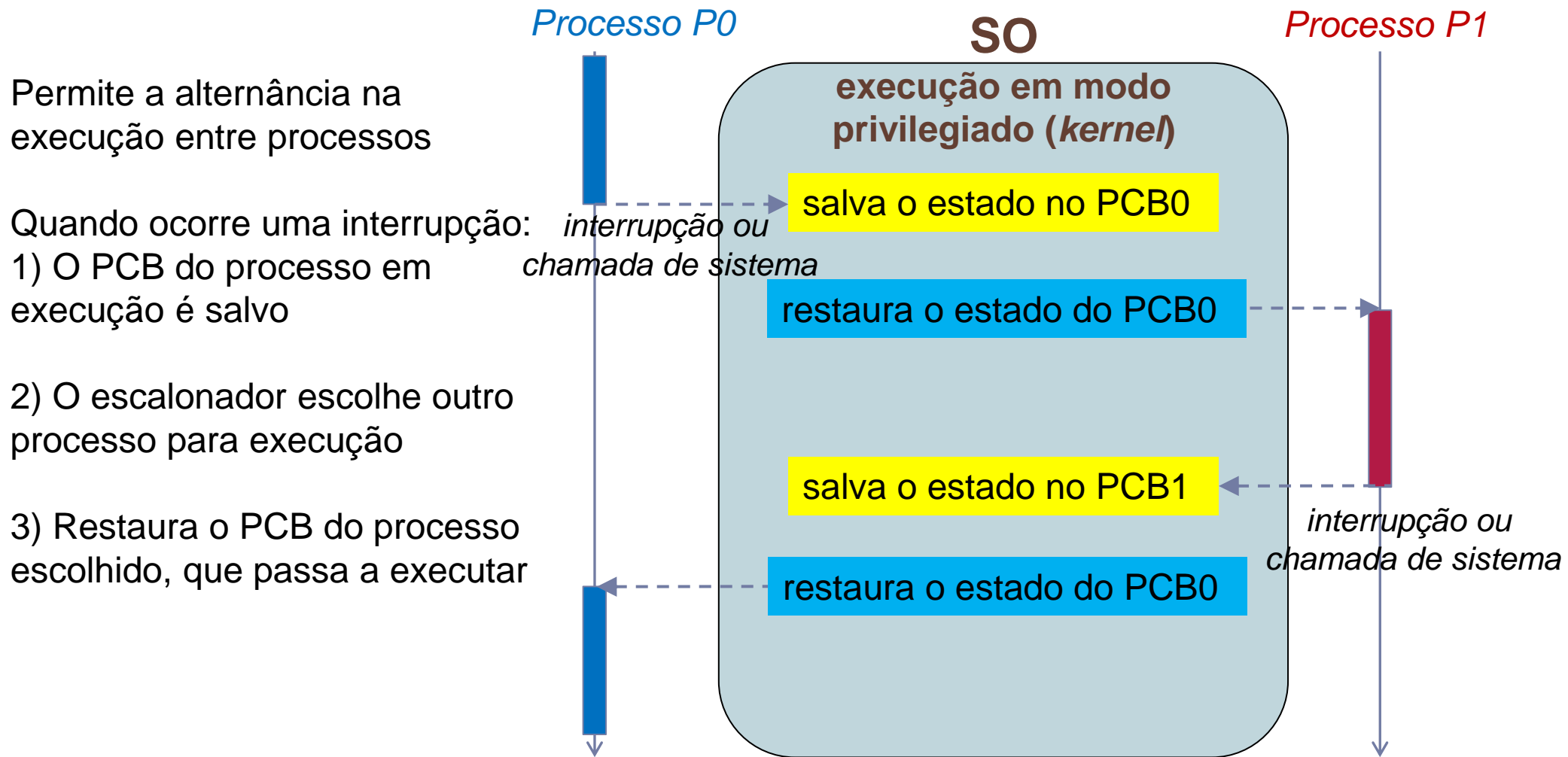
Filas de escalonamento



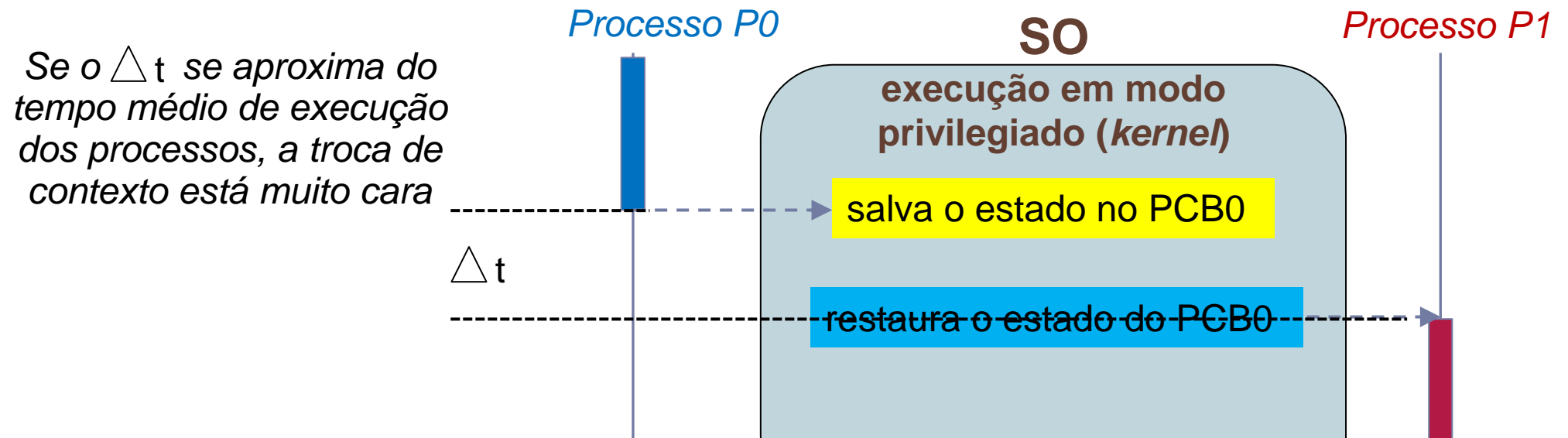
Representação do processo de escalonamento



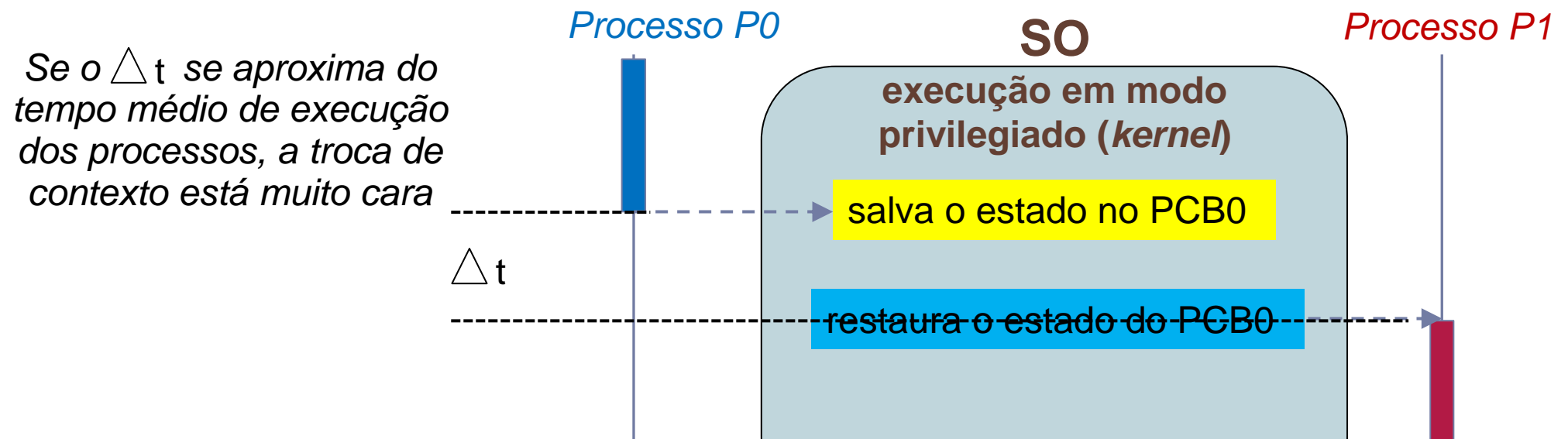
Troca de contexto



Troca de contexto



Troca de contexto



Custo da troca de contexto

- Interrupção leva à mudança do modo usuário para o modo privilegiado (*kernel*)
- Salvamento dos registradores da CPU e outras informações do processo para o PCB
- Escolha de um processo na fila de prontos (custo do escalonamento)
- Carregamento dos registradores da PCB escolhida para o CPU

Impacto na escolha do escalonador

- A escolha de um processo aguardando por execução precisa ser rápida
 - Isto reduz o tempo levado pela troca de contexto
- Escolhas rápidas exigem políticas de escalonamento eficientes
 - Algoritmos de escalonamento serão discutidos nas próximas aulas
- Por exemplo:
 - Escalonador leva **10ms** para decidir executar um processo por **100ms**
 - Então:
 $10/(100+10) = 9\%$ da CPU executando o escalonador
Esta taxa de utilização é com gerenciamento apenas

Próxima aula

- Criação e finalização de processos
- Chamadas ao sistema relacionadas à processos
- Gerenciamento de processos

Referências

Parte destes slides são baseadas em material de aula dos livros:

- OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva.; TOSCANI, Simão Sirineo. *Sistemas operacionais*. 4. ed. Porto Alegre: Bookman, 2010. xii, 374p. (Livros didáticos, n.11) ISBN 9788577805211
- SILBERSCHATZ, Abraham.; GAGME, Greg; GALVIN, Peter B. *Sistemas operacionais com Java*. Rio de Janeiro: Elsevier, 2008. 673 p. ISBN 9788535224061
- TANENBAUM, Andrew S. *Sistemas operacionais modernos*. 3. ed. Rio de Janeiro (RJ): Prentice-Hall do Brasil, 2010. xiii, 653p. ISBN 9788576052371

