



## Processos

**Prof. Edwar Saliba Júnior**  
**Março de 2007**



## Processos

- **Programa em execução:**
  - processos do próprio sistema (*SYSTEM no gerenciador de tarefas*);
  - processos do usuário;
- **Sistemas multiprogramáveis: muitos processos em execução ao mesmo tempo;**
- **Curiosidade: *Job* = termo utilizado para designar processos em sistemas em *batch*.**



## Processos

- **Quem gerencia a execução de processos?**
  - **O sistema operacional:**
    - Alocação de recursos;
    - Compartilhamento de dados;
    - Troca de informações e
    - Sincronização da execução;
- **Alerta: programa != processo**
- **Processo = instruções do programa + informações necessárias para a sua execução (lembrando que o processo pode ser executado em várias “prestações”);**
- **Exemplo: Programa “Olá Mundo”.**

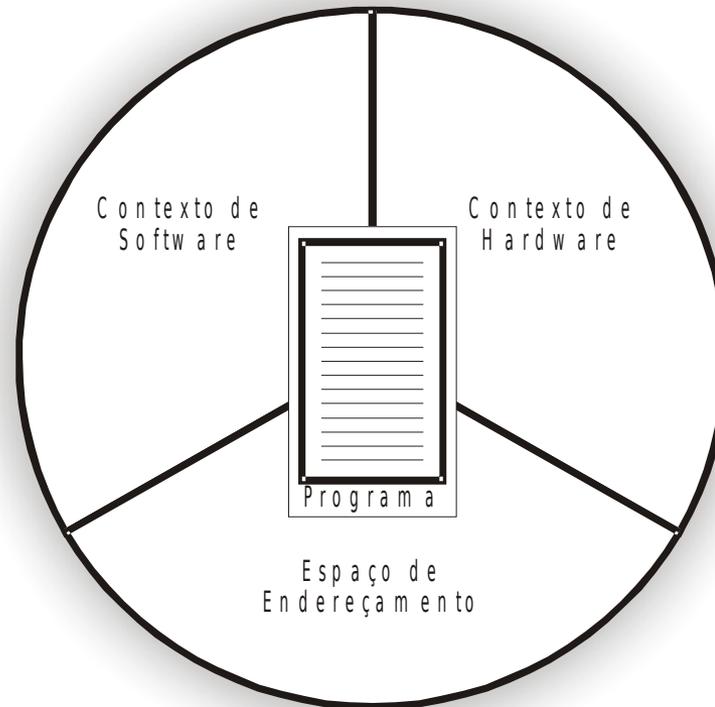


## Estrutura do Processo

- **Informações necessárias para execução do processo:**
  - Contexto de *hardware*;
  - Contexto de *software*;
  - Espaço de endereçamento.



## Estrutura do Processo





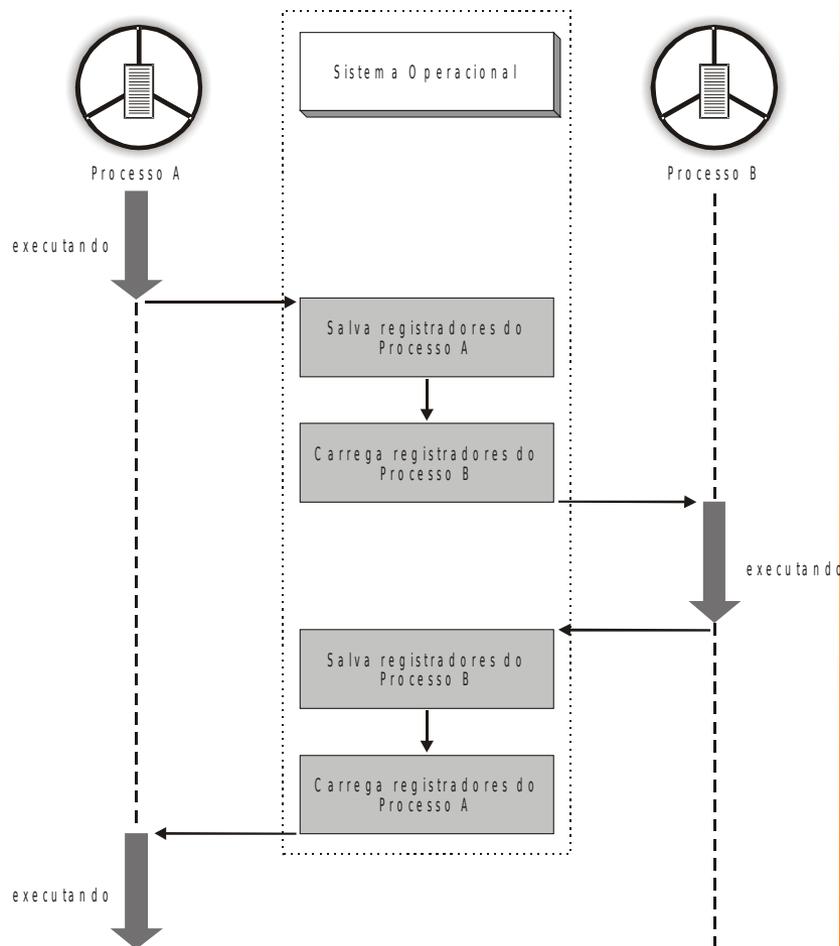
## Estrutura do Processo

- **Contexto de *hardware*: estado do *hardware* no momento em que o processo é interrompido para ceder lugar a um outro processo em execução (mudança de contexto):**
  - Ex: registradores.



## Contexto de Hardware

- Mudança de Contexto



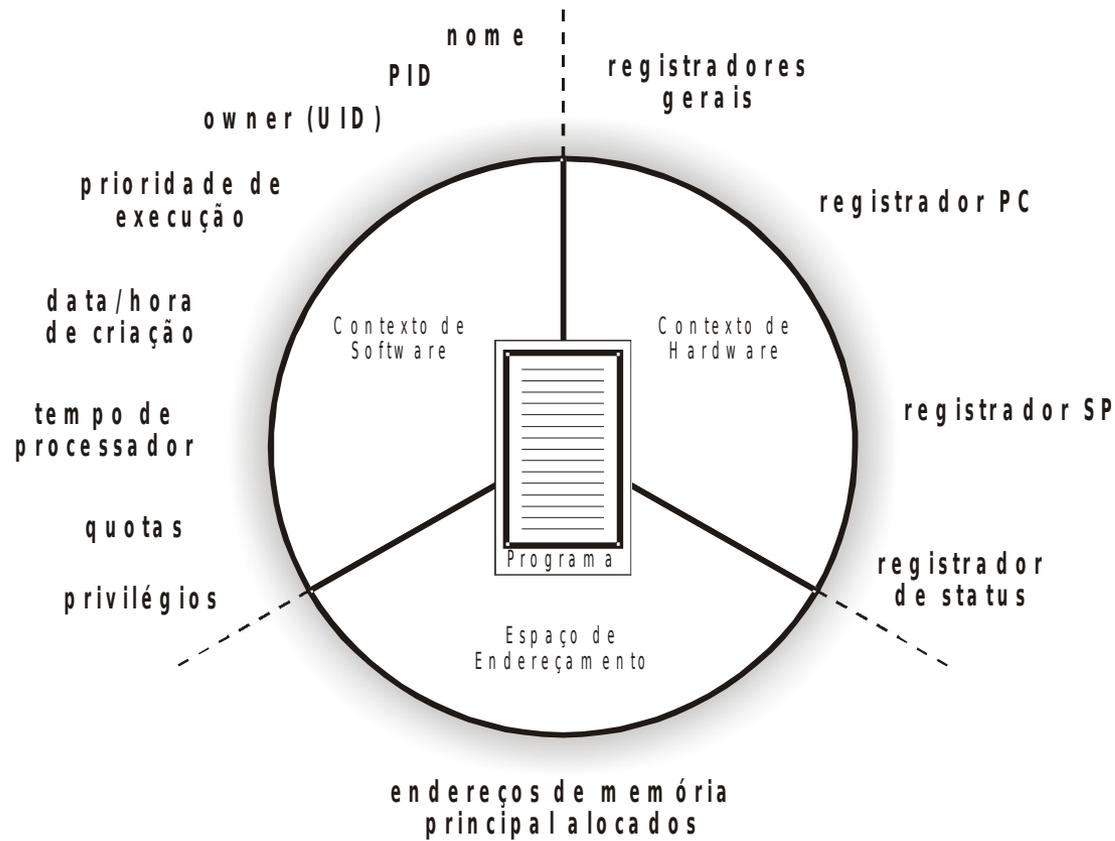


## Estrutura do Processo

- **Contexto de *software***: Características e limites dos recursos que podem ser alocados pelo processo:
  - Ex: identificação do processo (PID), usuário “dono” do processo, tamanho máximo em RAM que o processo pode alocar, tamanho máximo do *buffer* para operações de I/O;
- **Espaço de endereçamento**: Endereço da memória principal alocado para armazenamento de instruções e os dados utilizados pelo processo.



## Estrutura do Processo





## Processos

- **No Windows podemos ver algumas propriedades dos processos em execução através do gerenciador de tarefas.**



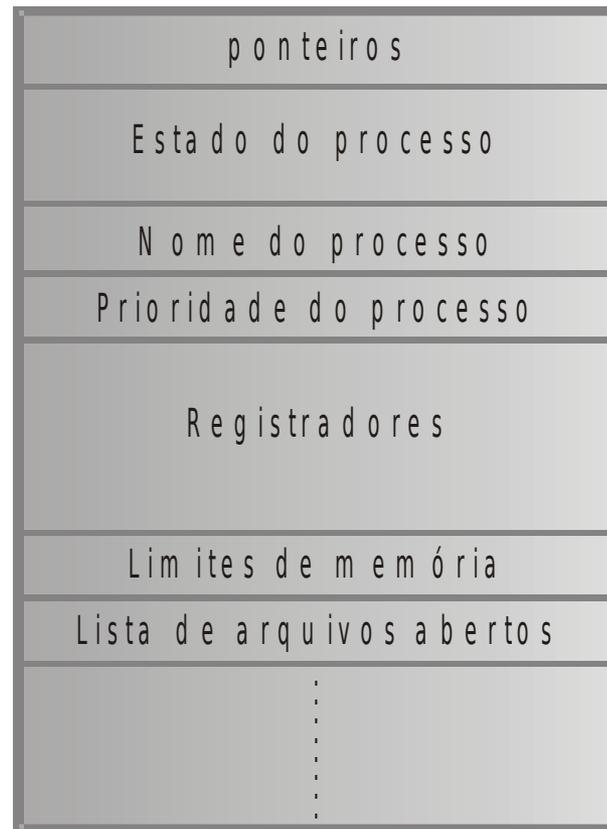
## PCB

- **Bloco de Controle do Processo (PCB – *Process Control Block*):** Estrutura utilizada para armazenar na memória principal as informações sobre os processos em execução (contexto de *hardware*, contexto de *software* e espaço de endereçamento).



# Fundamentos de Sistemas Operacionais

## PCB



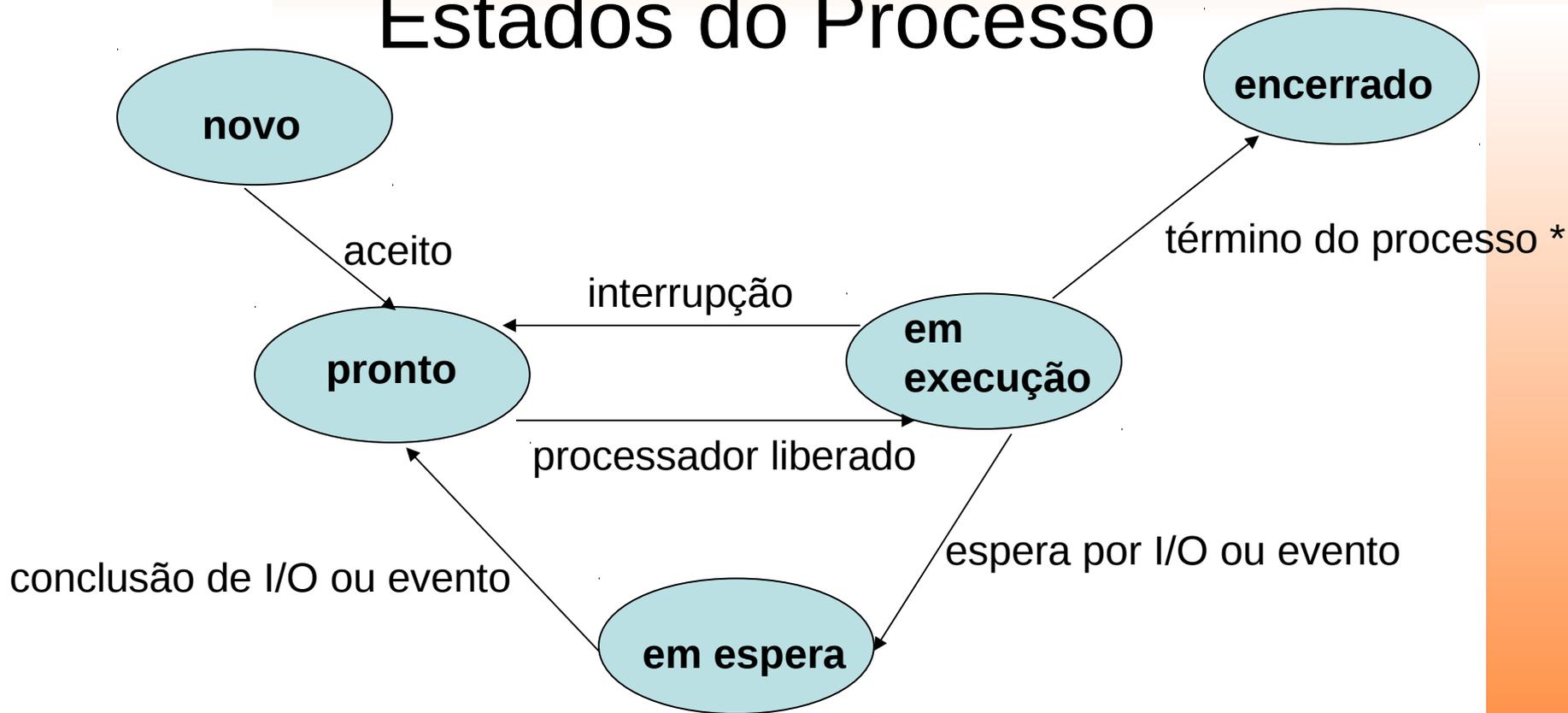


## Estados do Processo

- **Novo (*new*)**: O processo está sendo criado (BCP criado, porém ainda não pode ser executado);
- **Execução (*running*)**: As instruções estão sendo executadas:
  - Máximo de um processo em execução por processador em um dado instante;
- **Pronto (*ready*)**: Esperando para ser atribuído a um processador;
- **Espera (*wait*)**: Esperando a ocorrência de algum evento (conclusão de operação de I/O, por exemplo);
- **Encerrado (*exit*)**: Terminou a execução (recursos ainda não foram liberados e BCP ainda existe).



## Estados do Processo



\* O término do processo, na maioria das vezes, ocorre devido a: término das instruções a serem executadas, eliminação por ausência de recursos.



## Operações com Processos

- Um processo (dito Processo Pai) pode criar novos processos (ditos Processos Filhos) via chamada de sistema:
  - \_ Um processo filho, por sua vez, pode criar outros processos, formando uma árvore de processos;
  - \_ Um usuário pode solicitar a criação a um *shell*, que entende o pedido do usuário através de uma linguagem dita de controle (textual, gráfica, por eventos);
- Compartilhamento de Recursos (memória, arquivos, dispositivos de E/S), três formas distintas:
  - \_ Pai e filho compartilham todos os recursos ;
  - \_ Filho compartilha um subconjunto dos recursos do pai;
  - \_ Pai e filho não compartilham recursos;
- Execução, duas formas distintas:
  - \_ Pai e filho executam concorrentemente;
  - \_ Pai espera até o término do filho.



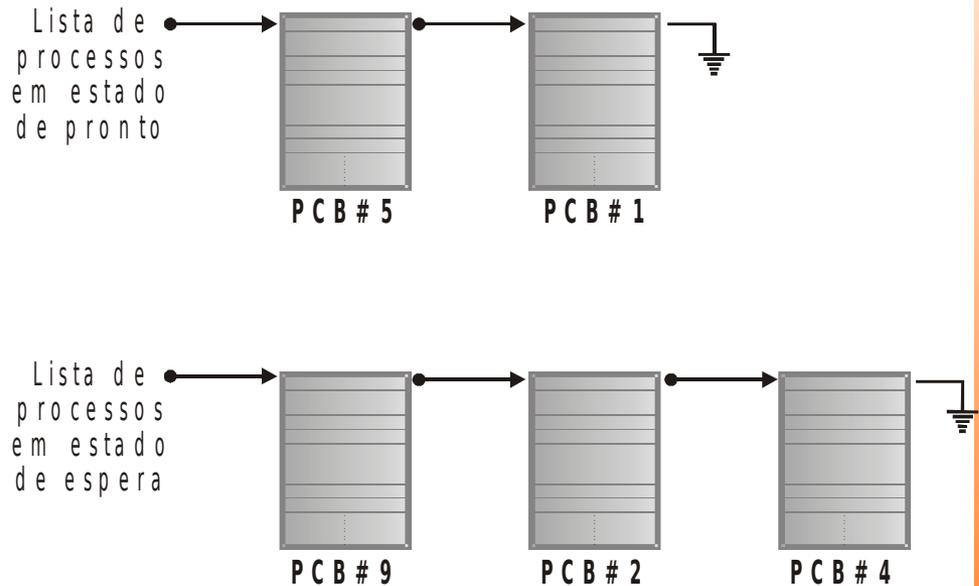
## Operação com Processos

- Processo executa sua última instrução e pede ao SO para terminá-lo (*exit*):
  - Dados de saída do filho para o pai (via *wait*);
  - *exit()*: todos os recursos do processo são liberados pelo SO;
- Um processo pai pode terminar a execução de um processo filho via chamada de sistema. Quando:
  - O filho excedeu os recursos alocados;
  - A tarefa atribuída ao filho não é mais necessária;
  - Se o pai está terminando;
    - Obs.: Alguns sistemas operacionais não permitem que o filho continue. Neste caso, se o pai terminou, então todos os filhos também terminam – terminação em cascata.



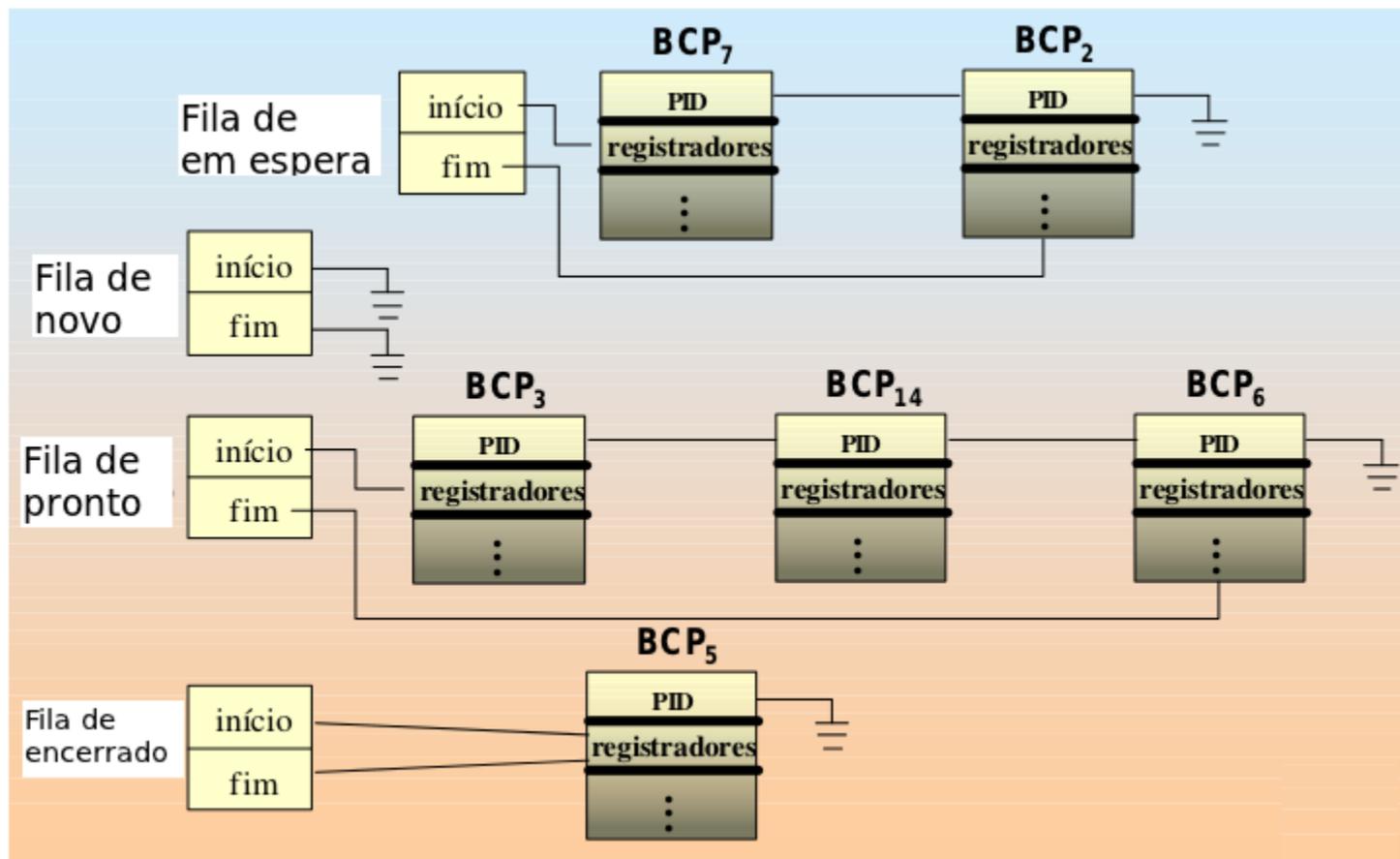
## Estados do processo

- **Para cada estado do processo existe uma fila de espera. Cada uma dessas filas é gerenciada pelo escalonador.**



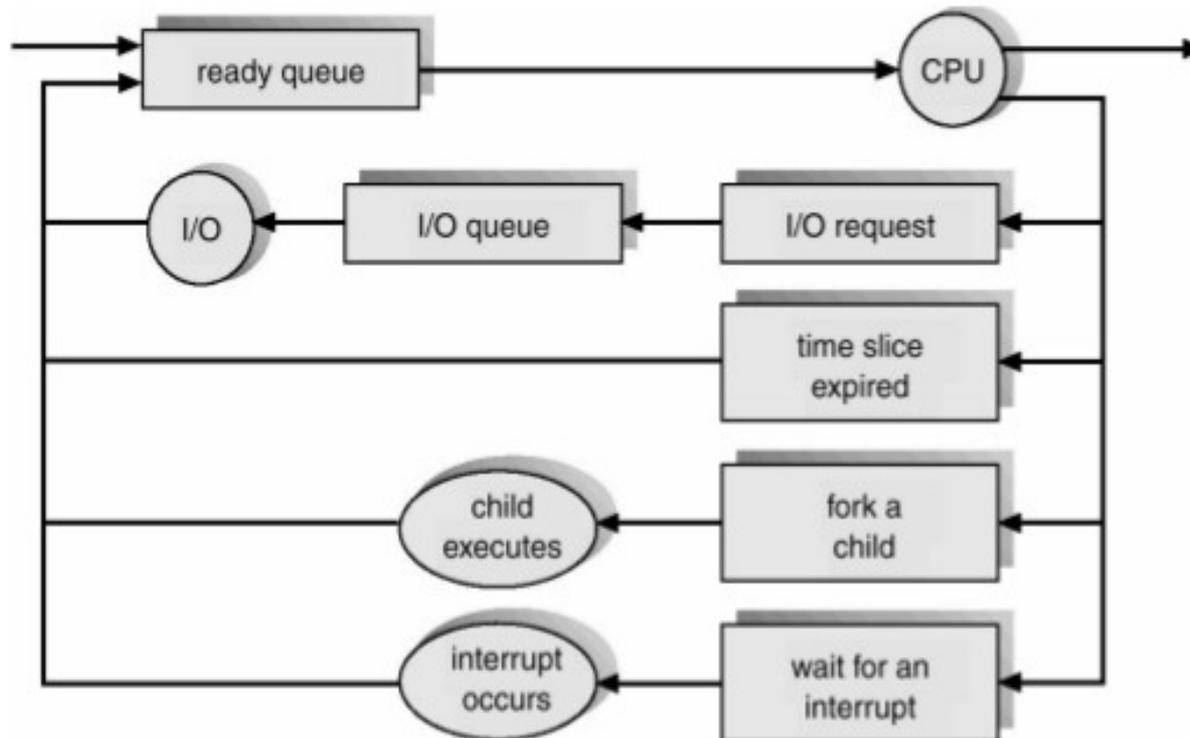


## Filas de Escalonamento





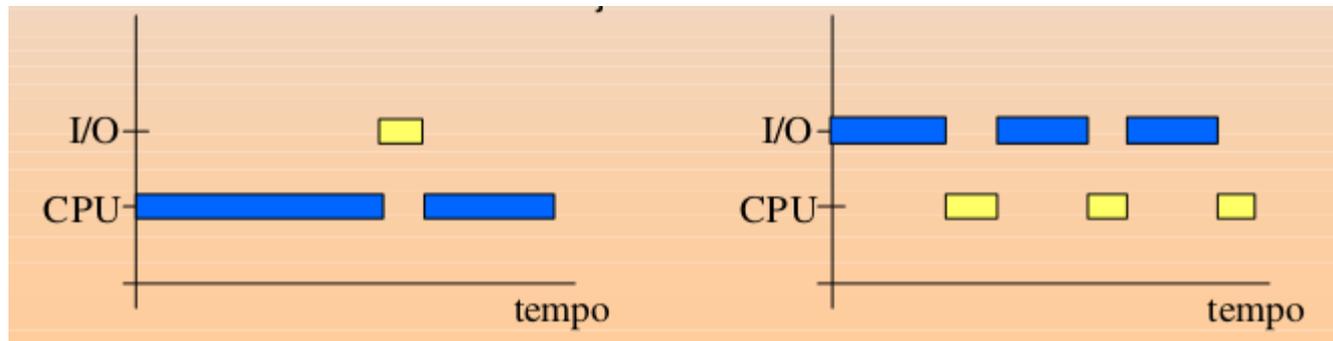
## Representação do Escalonamento de um Processo





## Tipos de Processos

- Processos podem ser descritos de acordo com o tipo de processamento que executam:
  - \_ CPU-bound (ligado à CPU): passa a maior parte do tempo fazendo cálculos (em execução) e realiza poucas operações de E/S (aplicações matemáticas e científicas);
- I/O-bound (ligado à E/S): passa a maior parte do tempo no estado bloqueado, pois realiza muitas operações de E/S (aplicações comerciais e interativas).



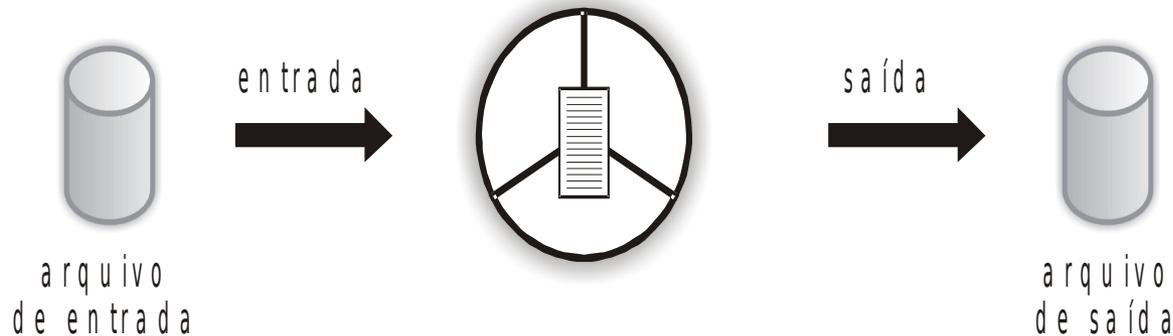


## *Background x Foreground*

(a) Processo Foreground



(b) Processo Background





## Relembrando...

- **Diferenciemos programa de processo;**
- **Quais as partes que compõem um processo?**
- **Resumidamente, quais os possíveis estados de um processo?**



## Bibliografia

- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**, 3<sup>a</sup> Ed., Rio de Janeiro: LTC Editora, 2002.
- SILVA, Guilherme Baião S. *Slides da disciplina de Sistemas Operacionais de Arquitetura Fechada*. Faculdade INED, 2005.