

## Aula 03: Processos

Introdução, conceitos, estados e implementação

Prof. Rodrigo Campiolo  
Prof. Rogério A. Gonçalves<sup>1</sup>

<sup>1</sup>Universidade Tecnológica Federal do Paraná (UTFPR)  
Departamento de Computação (DACOM)  
Campo Mourão, Paraná, Brasil

**Ciência de Computação**

BCC34G - Sistemas Operacionais

# Sumário

- 1 Introdução
- 2 Ciclos de processos
- 3 Relacionamento entre processos
- 4 Estados de processos
- 5 Implementação de processos
- 6 Referências

- Uma das funcionalidades do SO é o gerenciamento de aplicações.
- As aplicações quando estão em execução são denominadas de **Processos**.

## Processos

- Um processo é um programa em execução.
- Um processo é composto por: código, dados e contexto de execução.
- Um mesmo programa pode possuir várias instâncias em execução, com mesmo código, mas dados e contexto de execução diferentes.
- Processos se referem a:
  - Programas do usuário
  - Programas do sistema (*daemons*)

## Gerenciamento de Processos pelo SO

- criar e finalizar processos.
- suspender e retomar processos.
- bloquear e ativar processos.
- alterar a prioridade de processos.
- selecionar e despachar processos (para CPU).
- viabilizar à comunicação entre processos (IPC).

# Ciclos de um Processo I

## Ciclos

- Criação.
- Execução.
- Término.

## Criação

- Início do sistema.
- Execução de chamada ao sistema de criação de processos.
- Solicitação do usuário para criar um novo processo
- Início de um job em lote

## Execução

- CPU bound (orientados a CPU)
- I/O bound (orientados a E/S)



## Término

- Saída normal (voluntária).
- Saída por erro (voluntária).
- Erro fatal (involuntário).
- Cancelamento por um outro processo.

## Processos independentes

- Não apresentam relacionamento com outros processos.
- Por exemplo, no MS Windows não há relação hierárquica.

## Grupos de processos

- Apresentam uma relação entre os processos.
- Por exemplo, no Linux há uma relação hierárquica de filiação.

## Hierarquia de processos

- Processo criador é denominado pai.
- Processo criado é denominado filho.
- Questões:
  - Quais relações são mantidas entre o pai e filho?
  - O que fazer ao término de um processo?

Nota: mostrar pstree.

# Estados de Processos I

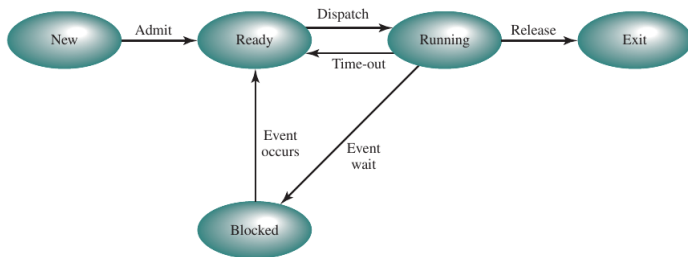


Figura 1: Diagrama de estados de um processo - cinco estados. (Stallings (2012))

## Descrição dos estados de processos

- **New (Novo)**: criado mas não admitido para a execução.
- **Ready (Pronto/Apto)**: disponível para a execução.
- **Blocked (Bloqueado)**: esperando por um evento (por exemplo, E/S).
- **Running (Execução)**: em execução.
- **Exit (Término)**: execução finalizada ou abortada.

# Estados de Processos III

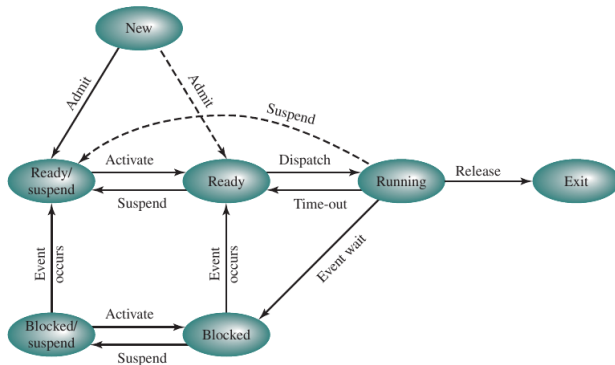


Figura 2: Diagrama de estados de um processo - sete estados. (Stallings (2012))

## Adição de dois novos estados:

- **Blocked/Suspend (Bloqueado/Suspenso):**  
Processo na memória secundária e esperando um evento.
- **Ready/Suspend (Pronto/Suspenso):**  
Processo na memória secundária e disponível para execução.

## Transições de estados

- **Blocked** → **Blocked/Suspend**: liberar espaço na memória principal para outro processo.
- **Blocked/Suspend** → **Ready/Suspend**: evento que estava esperando ocorreu.
- **Ready/Suspend** → **Ready**: não há processos prontos na memória principal.
- **Ready** → **Ready/Suspend**: liberar espaço na memória principal.
- **New** → **Ready/Suspend**: não há espaço na memória principal.
- **Blocked/Suspend** → **Blocked**: há memória disponível, carrega o processo.
- **Running** → **Ready/Suspend**: liberar espaço na memória principal (certas situações).



# Estados de Processos VI

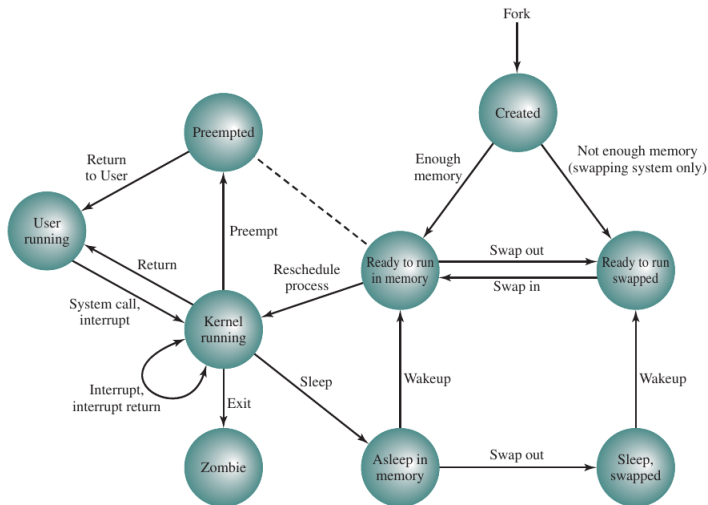


Figura 3: Estados de processos no Unix (1).

## Bloco de Controle de Processos (BCP)

- BCP é usado pelo SO para gerenciar os processos.
- É uma estrutura de dados mantida no núcleo do SO.
- Os BCP são mantidos em uma **Tabela de Controle de Processos**.
- Exemplo de BCP: `task_struct` no Linux<sup>a</sup>.

---

<sup>a</sup><https://github.com/torvalds/linux/blob/master/include/linux/sched.h>

# Implementação de processos II

<b>Process management</b>	<b>Memory management</b>	<b>File management</b>
Registers Program counter Program status word Stack pointer Process state Priority Scheduling parameters Process ID Parent process Process group Signals Time when process started CPU time used Children's CPU time Time of next alarm	Pointer to text segment info Pointer to data segment info Pointer to stack segment info	Root directory Working directory File descriptors User ID Group ID

Figura 4: Campos de um bloco de controle de processos.

- 1 Faça a leitura e anotações dos capítulos 4 e 5 (5.1,5.2 e 5.3) livro do Maziero (2017).
- 2 Faça a leitura e anotações do capítulo 2 (2.1) livro do Tanenbaum and Bos (2016).
- 3 Resolva a lista de exercícios L02 (Moodle).

# Referências I

- Deitel, H. M., Deitel, P. J., and Choffnes, D. R. (2003). *Operating systems*. Prentice-Hall, Inc., 3rd edition.
- Maziero, C. A. (2017). *Sistemas operacionais: conceitos e mecanismos*. online. Disponível em <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf>.
- Silberschatz, A., Galvin, P. B., and Gagne, G. (2015). *Fundamentos de sistemas operacionais*. LTC, 9 edition.
- Stallings, W. (2012). *Operating systems: internals and design principles*. Pearson Education, 7th edition.
- Tanenbaum, A. S. and Bos, H. (2016). *Sistemas operacionais modernos*. Pearson Education do Brasil, 4 edition.