



## PROCESSOS EM SISTEMAS OPERACIONAIS

**Bruna Lopes Marques**<sup>1</sup>

**Guilherme Silveira da Rosa**<sup>2</sup>

**João Padilha Moreira**<sup>3</sup>

### RESUMO

Diferente das versões iniciais de sistemas operacionais, os computadores são capazes de realizar diferentes tarefas ao mesmo tempo. No conceito fundamental de qualquer sistema operacional, um processo nada mais é que a abstração de um programa em execução, incluindo o sistema operacional. Embora dois processos possam estar associados a um mesmo programa, eles são considerados duas execuções totalmente distintas. Conceitualmente, cada processo possui sua CPU virtual. Na verdade, a CPU troca a todo momento de um processo para outro, mas é muito mais fácil pensar em um pseudo-paralelismo (ou seja, todos os processos são executados simultaneamente) para compreender como funciona este gerenciamento administrado pelo CPU. Este mecanismo de troca é chamado de Multiprogramação.

**Palavras-chave:** processos, sistemas operacionais, computadores, abstração, multiprogramação.

---

<sup>1</sup>Acadêmica do Curso Superior em Tecnologia em Programação para Internet – Faculdade Alcides Maya. brunam0191@gmail.com

<sup>2</sup> Acadêmico do Curso Superior em Tecnologia em Programação para Internet – Faculdade Alcides Maya. gui.srosa@gmail.com

<sup>3</sup> Professor do Curso S **Palavras-chave:** Algoritmos e Programação, raciocínio lógico-matemático, TDICs, Scratch.uperior em Tecnologia em Programação para Internet – Faculdade Alcides Maya. Joao\_moreira@alcidesmaya.edu.br



## 1. MULTIPROGRAMAÇÃO

Cada processo possui:

- Conjunto de instruções
- Espaço de endereçamento - espaço reservado na memória para que o processo possa ler e escrever, podendo sua posição variar de 0 até o máximo
- Contexto de hardware - é o armazenamento do estado o qual se encontra o hardware quando há troca de contexto para restaurá-lo posteriormente
- Contexto de Software - são especificados limites de recursos para cada processo. As características do processo são divididas em: identificação, quotas e privilégios.

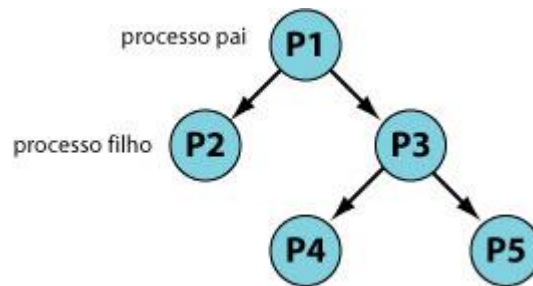
Sobre programas Nobrega nos diz :

Um programa é uma sequência de instruções ao computador. Quando o programador de software (uma pessoa que escreve programas para serem executados em um computador) desenvolve um programa, este é convertido em uma longa lista de instruções que são executadas pelo sistema de computador. Em sistemas operacionais nós falamos mais de um *processo* do que de um *programa*. Nos sistemas operacionais modernos, só uma porção de um programa é carregada em cada instante. O resto do programa espera numa unidade de disco até que se precise do mesmo. Isso economiza espaço de memória. Os programas no computador são executados por processadores. Um *processador* é um chip no computador que executa instruções de programa. Processadores executam milhões de instruções por segundo (NOBREGA, 1998, p.2).

## 1. HIERARQUIA DOS PROCESSOS

Em alguns sistemas, quando um processo (processo pai) cria outro (processo filho), eles continuam, de certa maneira, associados. O processo filho pode gerar outros processos, dando origem à chamada hierarquia de processos, conforme a figura 1.

Figura 1 – Hierarquia de Processos



Fonte: Sanches (2018).

Sobre hierarquia de processos, Tanenbaum diz que:

Outro exemplo de atuação dessa hierarquia pode ser observado no início do UNIX, quando um computador é ligado. Um processo especial, chamado *init*, está presente na imagem de carga do sistema. Quando começa a executar, ele lê um arquivo dizendo quantos terminais existem. Então ele se bifurca várias vezes para ter um novo processo em cada terminal. Esses processos esperam por alguma conexão do usuário. Se algum usuário se conectar, o processo de conexão executará um interpretador de comandos para aceitar comandos dos usuários. Esses comandos podem iniciar mais processos, e assim por diante (TANENBAUM, 2010, p.653).

## 2. CRIAÇÃO DE PROCESSOS

Em alguns sistemas, todos os processos necessários para realização da função são criados quando da inicialização do sistema. Indiferente do sistema, é necessário um mecanismo para início e término de processos durante a operação.

Existem 4 eventos principais que fazem com que os processos sejam criados.

### 1. Início do sistema

Processo *init*, é o primeiro processo a ser executado quando iniciado o sistema operacional

### 2. Execução da chamada de criação de um processo

Processo derivado de outros, sem a necessidade da intervenção do usuário.

### 3. Uma requisição do usuário para criar um processo

Quando o usuário inicia um programa.

### 4. O início de uma tarefa em lote



Este tipo de processo é disparado somente quando um arquivo de processamento em lote é executado.

### 3. TIPOS DE PROCESSO

Quando o sistema operacional é iniciado, são gerados vários processos. Alguns são em foreground (primeiro plano), ou seja, existe interação direta com o usuário e realizam tarefas por eles. Outros são processos em background (segundo plano), que são executadas sem interação com o usuário.

### 4. ESTADOS DE UM PROCESSO

Processos podem ser classificados em estados específicos. Estes estados especificam a forma em que o processo se encontra.

São eles:

- **Novo** - Um programa é lançado para execução, ou seja, é aberto pelo usuário. Este cria um ou mais processos novos que passam a ser reconhecidos pelo processador.

- **Suspenso** - Um processo está agendado para execução, ou está parado aguardando algum evento acontecer.

- **Pronto** - Um processo fica neste estado quando ele está realmente pronto para ser executado, e está apenas esperando que o Escalonador de Processos o dê permissão para

- **Executando** - O processo está sendo executado, ou seja, está utilizando completamente a CPU por um período determinado pelo Escalonador de Processos, podendo concluir sua execução antes deste período.

- **Espera** - Um processo está em espera quando ele é colocado na file de E/S e está aguardando até que obtenha o dado que precisa para passar para um novo estado. Geralmente processos ficam neste estado esperando que o usuário digite algo, ou esperando retorno de outro processo.

- **Completo** - Neste estado o processo será finalizado. passar para o próximo estado.



## 5. ESCALONADOR DE PROCESSOS

O Escalonador de Processos é um subsistema do Sistema Operacional que utiliza algoritmos de escalonamento para fazer a escolha lógica de qual processo executar por vez. Este algoritmo pode ser preemptivo, ou não-preemptivo.

Chama-se de preemptivo um algoritmo que utiliza tempo (chamado de Quantum) como parâmetro delimitador para execução de Processos. Um processo é interrompido quando ele não acaba antes do Quantum, então o Escalonador escolhe outro Processo para ser executado colocando este último no final da fila de execução.

Outro tipo de algoritmo escalonador é o não-preemptivo. Este trata a execução de um processo do início ao fim, sendo interrompido apenas no fim do processo, ou por algum comando dentro do próprio programa que está executando o processo.

## 6. TROCA DE CONTEXTO

Para que nós usuários tenhamos a ideia de paralelismo na utilização de programas, o CPU utiliza um conceito chamado de Troca de Contexto. Esta ação consiste em salvar o estado de um processo, armazenar este estado em memória e registradores e em seguida interromper o mesmo. Logo, a CPU está apta a executar outro processo. Caso este não seja um novo processo, ela utiliza os dados do estado deste processo que foram previamente salvos na última execução do mesmo para seguir da onde ela parou.

## 7. TÉRMINO DE PROCESSOS

Após concluir o seu trabalho, o processo é finalizado conforme uma das seguintes condições abaixo:

1. Saída normal neste caso, o processo concluiu seu trabalho e encerra normalmente
2. Saída por erro quando um parâmetro inexistente é fornecido.



3. Erro fatal, quando o término de um processo é decorrente de um erro não tratado pelo programa
4. Cancelamento por um outro processo (involuntário), é quando o sistema executa uma chamada para finalizar outro processo. Quando um processo termina, todos os processos filho derivados deste são imediatamente cancelados.

## 8. REFERÊNCIAS

NOBREGA, Raimundo (Ed.). **Introdução aos Sistemas Operacionais**. 1998. Disponível em: <<http://www.di.ufpb.br/raimundo/SistOper/os01port.htm>>. Acesso em: 21 maio 2019.

SANCHES, Renan de Oliveira. **Hierarquia de Processos no Unix e Windows**. 2018. Disponível em: <<https://www.devmedia.com.br/hierarquia-de-processos-no-unix-e-windows/24739>>. Acesso em: 10 maio 2018.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Companion Website, 2010.