

# Sistemas Operacionais

## Prof. Robson de Souza

### Aulas 5 e 6

**Conteúdo:** Processos e chamadas de sistema.

#### \*Processos

Um processo é um programa em execução.

Um processo precisa de certos recursos, incluindo tempo de CPU, memória, arquivos e dispositivos de E/S, para realizar sua tarefa.

A cada processo está associado o seu **espaço de endereçamento**, que é uma lista de locais da memória onde o processo pode ler e gravar. O espaço de endereçamento contém o programa executável, os dados do programa e a sua pilha.

Existem uma série de registradores associados a cada processo, incluindo o contador de programa, o ponteiro da pilha, outros registradores e demais informações necessárias à execução do programa.

Em um Sistema Operacional, as informações sobre o processo estão salvas em uma **tabela de processos**, que é uma matriz ou lista encadeada de estruturas, uma para cada processo existente atualmente. Essa tabela contém as informações referentes ao processo sem contar as que já estão salvas no espaço de endereçamento.

Uma das funções do S.O é a gerência de processos, as seguintes atividades são responsabilidade do S.O com relação ao gerenciamento de processos:

- Criar e excluir processos
- Suspender e retomar processos
- Oferecer mecanismos para:
  - sincronismo de processos
  - comunicação entre processos

#### \*Chamadas de Sistema (System Calls)

As **chamadas de sistema (system calls)** fornecem a interface entre um programa em execução e o sistema operacional.

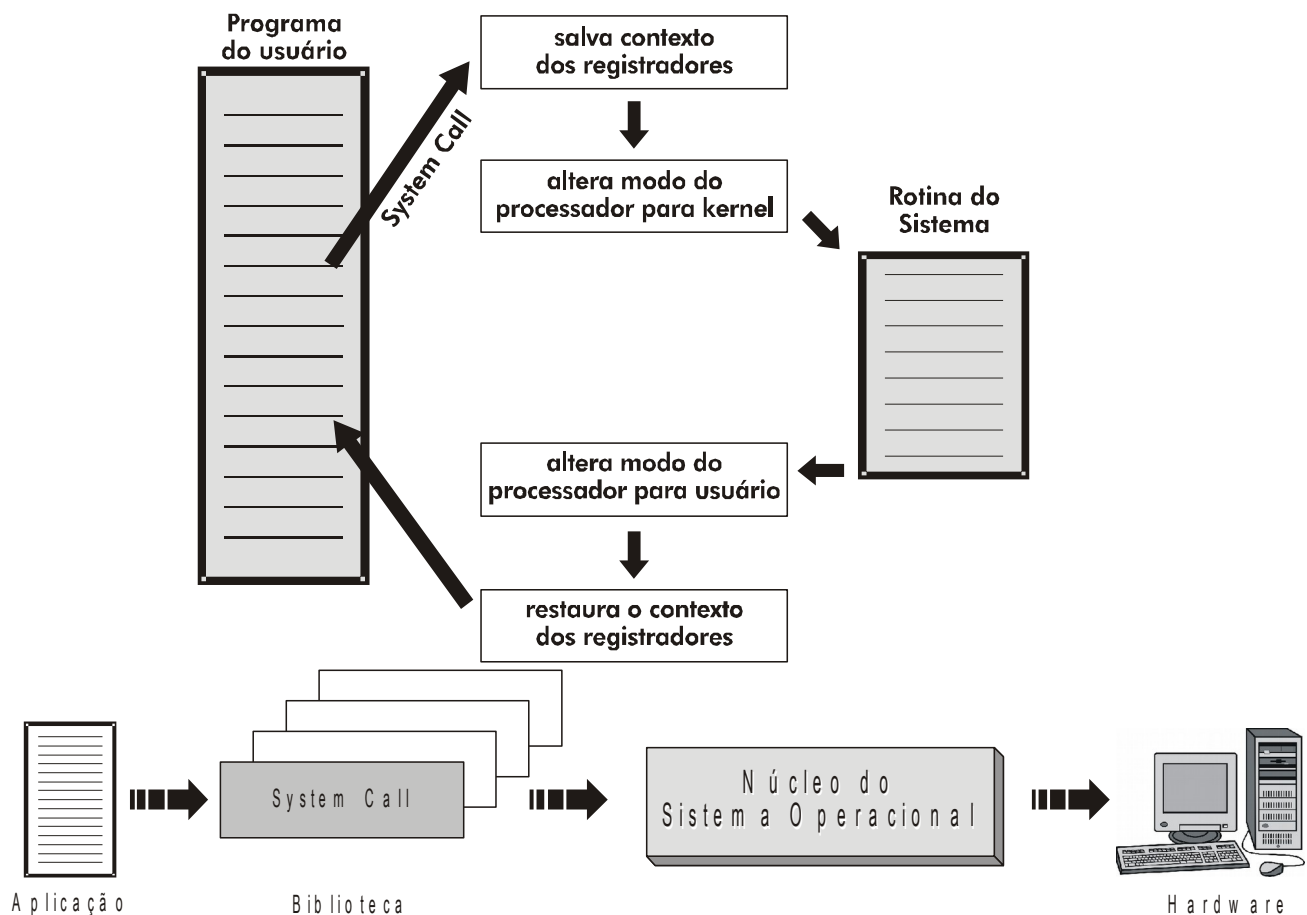
Geralmente disponíveis como instruções em linguagem assembly.

Linguagens definidas para substituir a linguagem assembly para a programação de sistemas permitem que as system calls sejam feitas diretamente (por exemplo, C, C++).

Três métodos são usados para passar parâmetros entre um programa em execução e o sistema operacional

- Passar parâmetros via registradores
- Armazenar os parâmetros em uma tabela na memória, e o endereço na tabela é passado como um parâmetro em um registrador
- O programa coloca (push) os parâmetros na pilha e o Sistema Operacional retira (pop) os parâmetros da pilha

A figura abaixo mostra o fluxo de uma chamada a uma rotina do sistema:



Em um sistema operacional, para manter os arquivos, geralmente se tem o conceito de diretório onde cada arquivo está armazenado em um diretório que representa uma área do dispositivo de armazenamento, o S.O tem a função de trabalhar dessa forma, ocultando os detalhes de hardware com relação ao armazenamento físico. Para criar e remover diretórios são necessárias chamadas de sistema.

Cada arquivo dentro de um diretório pode ser encontrado pelo **nome do caminho** a partir do topo da hierarquia de diretórios, que é chamado de **diretório-raiz**. A cada instante, um processo tem um diretório de trabalho atual.

Os processos podem mudar seu diretório de trabalho emitindo uma chamada de sistema especificando o novo diretório de trabalho.

Por exemplo, Supondo que existe uma hierarquia de pastas até um arquivo chamado “aula.pdf”:

diretório raiz → UNIPAC → Aulas SO → Chamadas de sistema → aula.pdf

Nesse caso, para acessar o arquivo é necessário o caminho:

`/UNIPAC/Aulas SO/Chamadas de sistema/aula.pdf`

Se o diretório de trabalho atual não for a raiz mas o diretório “Aulas SO”, então o caminho abaixo equivaleria ao caminho especificado acima:

`/Chamadas de sistema/aula.pdf`

As chamadas de sistema possuem vários grupos:

### **Chamadas de sistema para gerenciamento de processos**

O primeiro grupo de chamadas de sistema trata do gerenciamento de processos, essas chamadas podem ser utilizadas para criar processos, esperar um processo terminar, terminar a execução de um processo, retornar o status, retornar o id de um processo, entre outros.

Um exemplo de chamada de sistema seria o comando **fork**, que em alguns sistemas operacionais serve para criar um novo processo, nesse caso, é criada uma duplicata exata do processo criador, depois do fork, o processo original e a cópia seguem caminhos diferentes. Como os dados do processo pai e filho são copiados de um para o outro, mudanças futuras nos dados não afetam o outro.

### **Chamadas de sistema para sinalização**

No geral, as comunicações entre os processos do sistema são planejadas, porém podem ocorrer alguns imprevistos e um processo pode precisar enviar algum tipo de sinalização para a execução de algo que não foi planejado. Por exemplo, um usuário digitou um texto em um editor e selecionou a opção de imprimir o texto, assim que o S.O gerou o processo de impressão, o usuário percebeu que existia um erro no texto que precisava ser corrigido antes de imprimir, logo, é necessário algum mecanismo para interromper a impressão atual.

Os sinais também servem para informar sobre alguma exceção detectada pelo hardware, como uma instrução ilegal ou um timeout.

### **Chamadas de sistema para gerenciamento de arquivos**

Essas chamadas de sistema servem para a manipulação dos arquivos no S.O, existem as chamadas que tratam de arquivos individuais e as chamadas utilizadas para manipular diretórios ou o sistema de arquivos como um todo. As chamadas dessa categoria podem ser utilizadas para criar arquivos, ler um arquivo, abrir um arquivo, escrever um arquivo, buscar um texto em um arquivo, entre outros.

Para trabalhar com arquivos, alguns Sistemas Operacionais trabalham com modos distintos:

**Modo processado** → Modo terminal normal, onde os comandos do teclado funcionam normalmente, como por exemplo os comandos iniciados pela tecla CTRL seguidos por algum caractere.

**Modo bruto** → Nesse modo, todas as funções do modo processado são desativadas, os comandos não funcionam e todos os caracteres tem de ser passados individualmente e diretamente para o programa.

**Modo cbreak** → É um meio termo entre os dois modos acima, permite alguns comandos do modo processado, mas não todos eles.

### **Chamadas de sistema para gerenciamento de diretórios**

São chamadas para se trabalhar com diretórios ou o sistema de arquivos como um todo. Nessa categoria existem comandos para criar e excluir diretórios, mover diretórios, permitir que um diretório seja acessado por vários usuários, entre outros.

### **Chamadas de sistema para proteção**

Essas chamadas servem para bloquear algumas coisas que uma aplicação ou um usuário podem fazer no sistema. Essas chamadas estão ligadas aos modos de acesso de um arquivo e permitem alterar essa característica.

É possível por exemplo, fazer com que um arquivo seja somente leitura para os usuários e leitura e escrita apenas para o criador do arquivo. Um exemplo de chamada que faz parte dessa categoria é o comando **chmod** que existe no linux, que serve para alterar as permissões de um determinado arquivo.

O nível de permissão que um usuário tem pode ser checado por meio de um **user id (uid)**, que é um número utilizado para identificação do usuário. Geralmente usuários normais não podem mudar o seu uid.

### **Chamadas de sistema para gerenciamento de tempo**

Nessa categoria estão as chamadas relacionadas a informações como data, hora, entre outras. Existem comandos para retornar a data, configurar o relógio, mudar data e hora de um arquivo, retorno de informações de contagem de um processo, entre outros.

Um exemplo de utilidade importante dessa categoria é o fato de que alguns comandos podem ser utilizados para que um programador seja capaz de medir o tempo de execução de um programa e avaliar seu desempenho.

### **Referências bibliográficas:**

TANENBAUM, Andrew. 2ª ed. **Sistemas Operacionais Modernos**, Editora Pearson, 2003.

SILBERSCHATZ, Abraham. **Sistemas Operacionais com JAVA**, 6ª ed. Editora Campus

MACHADO, Francis B. **Arquitetura de Sistemas Operacionais**, 4ª ed, LTC, 2007.