

## Capítulo 2

### Serviços do Sistema Operacional

Uma visão bastante comum do sistema operacional é aquela que encara este *software* como uma extensão da máquina, fornecendo mais serviços para os aplicativos e outros programas básicos. Além disso, o sistema operacional pode ser considerado também um administrador e fornecedor de recursos (incluindo serviços). Ele cuida de todos os recursos que estão disponíveis no computador, permitindo ao usuário utilizar a máquina (*hardware* + SO) de maneira amigável. Isso é fundamental em qualquer máquina, tornando-se mais crítico naquelas que permitem mais de um usuário ao mesmo tempo.

Neste capítulo são apresentados os serviços que são oferecidos pelo sistema operacional e mostradas a visão do usuário e a visão do sistema operacional, em relação à máquina.

## 2.1. Tipos de Serviços

O sistema operacional fornece um ambiente para a execução de programas através de serviços para os programas e para os usuários desses programas. Alguns serviços não têm como preocupação apenas tornar a máquina mais confortável para o usuário, mas também para que o próprio sistema seja mais eficiente e seguro. Esse é o caso dos serviços oferecidos nos sistemas que permitem vários usuários compartilhando todos os recursos da máquina. Apesar da forma como esses serviços são oferecidos variar de sistema para sistema existem algumas classes de serviços que são comuns a todos os sistemas operacionais. Estes compõem a sua própria definição. Como exemplo, temos:

- **Execução de programas:** O sistema operacional é o responsável por carregar um programa na memória principal da máquina e executá-lo. O programa é o responsável pelo término da sua própria execução;
- **Operações de entrada/saída:** Durante a sua execução, um programa pode ter necessidade de se comunicar com o meio externo à máquina. Esta operação recebe o nome de entrada/saída (E/S) e pode envolver qualquer dispositivo de E/S (disco, impressora). Como um programa não pode executar estas operações diretamente, o sistema operacional é o responsável por fornecer meios adequados para isso;
- **Manipulação de sistema de arquivos:** Os usuários de uma máquina têm necessidade de realizar acessos aos arquivos pelo nome para saber se eles existem, para apagá-los ou até para renomeá-los. Um programa em execução pode querer ler ou escrever num arquivo qualquer. O sistema operacional é o responsável por gerenciar o sistema de arquivos da máquina. Este gerenciamento inclui a alocação de espaço no dispositivo de armazenamento secundário, a busca otimizada a um determinado arquivo e o armazenamento de todas as informações necessárias sobre cada arquivo.
- **Deteção de erros:** O sistema operacional é o responsável por detectar erros possíveis que podem comprometer a execução de qualquer programa e a segurança da máquina. Estes erros podem envolver o próprio processador a memória principal (acesso a uma área proibida), os dispositivos de entrada/saída (falta de papel na impressora), ou até mesmo o programa do usuário (uma divisão por zero). Para cada tipo de erro, o sistema operacional tem uma ação apropriada para garantir a correção e a consistência da computação.
- **Alocação de recursos:** O sistema operacional é o responsável pela alocação dos diversos recursos em sistemas com um ou mais usuários. Estes recursos incluem a memória principal, a própria UCP, arquivos e os dispositivos de E/S. A alocação deve ser feita da forma mais eficiente possível para não prejudicar o desempenho do sistema.
- **Proteção:** O sistema operacional é o responsável pela proteção a todo o sistema computacional. Essa proteção se torna necessária tanto em sistemas monousuários quanto em sistemas multiusuários. A única diferença é a sua complexidade. Quando vários usuários estão usando o sistema, a execução de um programa não pode interferir na execução de outro. Além disso, o próprio sistema operacional deve ser protegido de erros cometidos pelos usuários.

## 2.2. Usuários x Sistema Operacional

A seguir serão apresentadas as duas formas de o sistema operacional fornecer serviços aos usuários: chamadas ao sistema e programas utilitários.

### Chamadas ao Sistema

O nível mais fundamental de serviços fornecido pelo sistema operacional é realizado através de **chamadas ao sistema** (*system calls*). As chamadas fornecem uma interface entre um programa em execução e o sistema operacional. Estão, geralmente, disponíveis como instruções nas linguagens de baixo nível ou até mesmo em linguagens de alto nível, como C. As chamadas ao sistema podem ser classificadas, em duas categorias principais:

- **Controle de processos.** Nessa categoria existem chamadas ao sistema para a criação e a finalização de processos, a manipulação de tempo para manter a sincronização entre processos concorrentes, o carregamento e a execução de programas, a obtenção e a ativação de atributos dos processos, entre outras.
- **Gerenciamento de arquivos e de dispositivos de E/S.** Nesta categoria existem chamadas ao sistema para criar, apagar, abrir e fechar um arquivo, ler e escrever em arquivos, e ainda obter e modificar os atributos de um arquivo, entre outras. Os atributos de um arquivo incluem, por exemplo, o seu nome, tipo, códigos de proteção e tamanho. Em geral, a única informação necessária para o sistema realizar um acesso a um arquivo é o seu nome. No caso de dispositivos de E/S, existem ainda as chamadas para requisitar e liberar um dispositivo.

Na figura 2.1 é mostrado um esquema de atendimento do sistema operacional às chamadas ao sistema. Inicialmente (1), o processo escreve nos registradores da máquina o código da chamada e os seus parâmetros. A seguir (2), o processo executa uma instrução **trap** e o sistema operacional interpreta a chamada (3), executa o serviço solicitado (4) e devolve o controle para o processo (5).

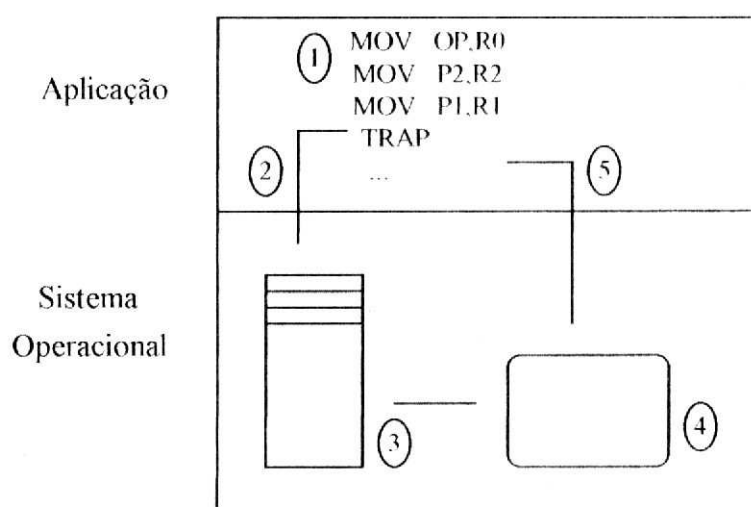


Figura 2.1. Exemplo de chamada ao sistema

---

Uma instrução *trap* muda o estado do processador do modo usuário para o **modo supervisor** (ou **sistema**). No primeiro modo, as instruções que afetam o *hardware* não podem ser executadas e, no segundo, todas as instruções podem ser executadas. A distinção entre o modo usuário e o supervisor ou sistema existe para proteção do sistema operacional e para proteção entre usuários.

Na implementação de uma chamada ao sistema são necessárias mais informações do que simplesmente identificar a chamada desejada. A quantidade e o tipo exato de informações variam de acordo com a chamada e o sistema operacional em particular. Em geral, existem dois métodos usados para passar parâmetros ao sistema operacional. A abordagem mais simples é passá-los em registradores. Entretanto, em alguns casos, pode haver mais parâmetros do que registradores. Nestes casos os parâmetros são geralmente armazenados em blocos ou tabelas na memória principal e o endereço do bloco ou tabela é passado como um parâmetro em um registrador.

Para exemplificar o uso das chamadas ao sistema podemos considerar um programa em execução que leia informações de um arquivo e escreva em outro. As primeiras informações de que o programa necessita são os nomes dos arquivos que serão manipulados. Esses nomes podem ser pré-definidos dentro do próprio programa em execução ou passados pelo usuário durante a execução (isso exige uma outra sequência de chamadas ao sistema). Uma vez que os nomes dos arquivos tenham sido obtidos, o programa deve abrir o arquivo de entrada, que vai ser lido (*open*) e criar o arquivo de saída (*create*). Algum tipo de tratamento de erro pode ser previsto, como por exemplo, se o arquivo de entrada ainda não foi criado, se ele possui algum tipo de proteção ou se já existe algum arquivo com nome igual ao do arquivo de saída. Cada uma dessas operações requer uma chamada ao sistema.

Depois que os dois arquivos já foram abertos, o próximo passo é ler do arquivo de entrada (*read*) e escrever no arquivo de saída. Neste ponto, tanto na leitura como na escrita, outras situações também podem ser previstas, como a chegada ao final do arquivo, erro no dispositivo de armazenamento secundário, falta de espaço no disco para o arquivo de saída.

Finalmente, depois que o arquivo de entrada foi totalmente copiado para o arquivo de saída, o programa em execução deve fechar os dois arquivos (*close*) e terminar normalmente (*end*).

## Programas Utilitários

Outros tipos de serviços oferecidos pelo sistema operacional são os chamados **programas utilitários** (*systems programs*). Esses serviços são de mais alto nível e fornecem uma interface entre o usuário e o sistema operacional. Como foi visto no exemplo anterior, um programador pode escrever um programa que copie o conteúdo de um arquivo de entrada para um arquivo de saída, e utilizá-lo sempre que necessário. Embora possível esta não seja a forma mais confortável, pois obriga o usuário a conhecer detalhes complexos da máquina, e a ter conhecimento de técnicas de programação. Além disso, vários usuários têm necessidade deste tipo de operação e seria bastante interessante que existisse alguma padronização.

Dessa forma então, a maioria dos sistemas operacionais oferece um conjunto de programas utilitários que tornam o ambiente mais conveniente para o desenvolvimento e execução de programas. Podem-se agrupar os programas utilitários da seguinte forma:

- **Manipulação de arquivos:** Inclui apagar, copiar, ver o conteúdo, renomear e imprimir um arquivo e ainda verificar a existência do arquivo;
- **Informações sobre o sistema:** Inclui obter e modificar data e hora da máquina, conhecer a quantidade de memória secundária disponível e outras informações de estado.
- **Suporte para linguagens de programação:** Junto com o sistema operacional podem ser fornecidos, ou vendidos separadamente, programas como compiladores, montadores, interpretadores para as linguagens de programação mais comuns como Pascal, Fortran ou C;
- **Carregamento e execução de programas:** Uma vez compilado, montado e livre de erros, um programa deve ser carregado na memória principal e executado. O sistema pode oferecer instrumentos para que essas duas operações sejam efetuadas mais eficientemente (carregadores absolutos ou realocáveis, *linkage editors* e depuradores).

Existe ainda um programa utilitário muito importante que recebe o nome de **interpretador de comandos**. Esse programa começa a executar quando o sistema é iniciado e fica esperando que um usuário digite um comando para interpretá-lo. Existem duas alternativas possíveis na implementação do interpretador de comandos. O interpretador pode conter o código que executa o comando pedido (ativa os parâmetros e invoca as chamadas ao sistema necessárias) ou então cada comando é implementado por um programa independente. Cada programa possui um nome próprio, e a função do interpretador é carregá-lo na memória principal e executá-lo.

No primeiro caso, o tamanho do interpretador de comandos depende da quantidade de comandos existentes, já que uma parte do seu código contém o código do próprio comando. No segundo caso, o interpretador apenas ativa o programa que contém o comando. Além disso, outros comandos podem ser adicionados ao sistema através da criação de arquivos novos e o tamanho do interpretador pode ser relativamente pequeno.

A visão que os usuários têm do sistema operacional é definida mais pelos programas utilitários que pelas chamadas ao sistema, particularmente pelo interpretador de comandos. A visão que um projetista do sistema operacional tem do sistema como um todo é bastante diferente. Ele vê os recursos físicos (disco, impressora, teclado, vídeo) e deve convertê-los em facilidades lógicas fornecidas aos usuários.

## 2.3. Comportamento do Sistema Operacional

Os sistemas operacionais têm uma característica bastante interessante: se não existir nenhum programa a executar, nenhum dispositivo de entrada/saída a ser atendido e nenhum usuário aguardando uma resposta, o sistema operacional fica esperando a ocorrência de algum evento. Eventos são, em geral, representados por **interrupções**.

Quando uma interrupção ocorre, o *hardware* transfere o controle para o sistema operacional. Nesse momento, o sistema operacional salva o estado da máquina (registradores e contador de programa) e determina qual foi o tipo de interrupção que ocorreu, já que existem diferentes tipos de interrupção. Por exemplo, uma chamada ao sistema (*create*), uma interrupção de um dispositivo de E/S (impressora), uma interrupção de erro (tentativa de executar uma instrução privilegiada). Para cada tipo de interrupção um tratamento diferente tem de ser dado.

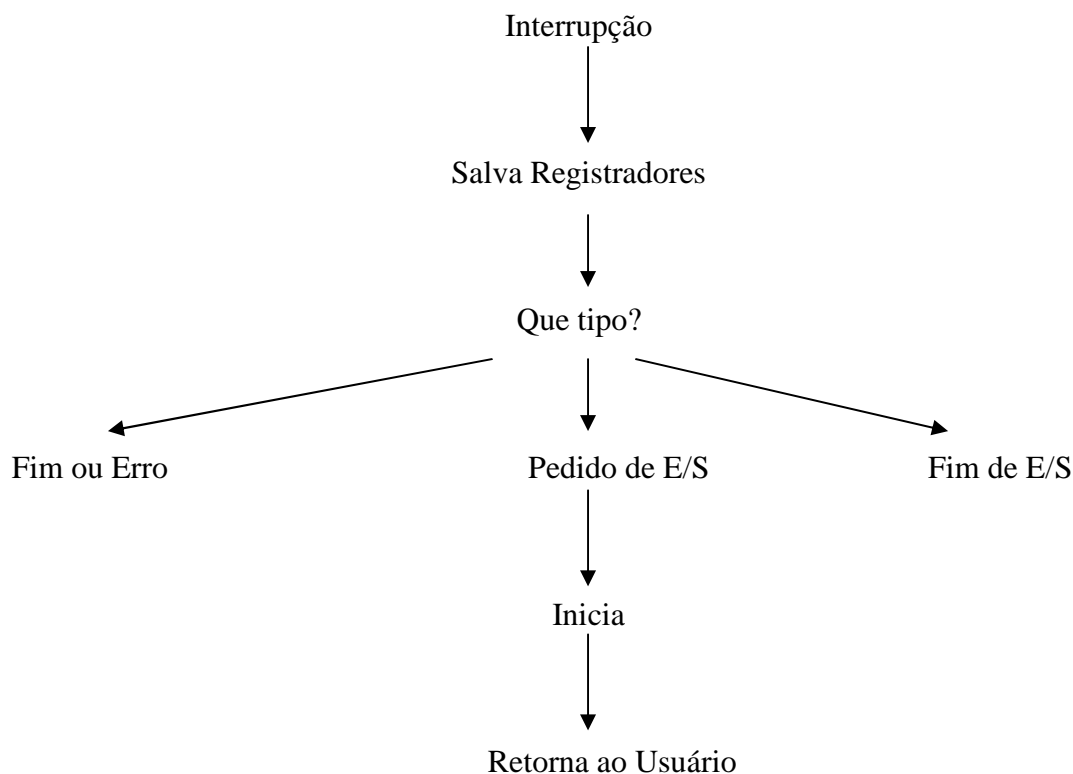
Na visão do sistema operacional, as chamadas ao sistema são agrupadas de acordo com o seu tipo. Para cada chamada um segmento de código é executado. Apesar de o sistema tratar muitas chamadas, a maioria dos eventos que ocorrem pertencem à classe de interrupções dos dispositivos de E/S. Uma operação de E/S é resultante de uma chamada ao sistema requisitando tal serviço. Uma situação prática seria um programa abrir um arquivo e escrever alguma informação nele. Neste caso, o dispositivo de E/S usado poderia ser um disco e várias operações de E/S e uma sequência de chamadas ao sistema seriam realizadas.

Uma vez que uma operação de entrada/saída tem início, dois cenários podem ocorrer. O primeiro é quando o controle só retorna para o programa do usuário, que gerou a chamada ao sistema, quando a operação tiver sido terminada. No segundo, o controle retorna ao programa do usuário sem esperar que a operação de entrada/saída tenha sido terminada. Ambas as situações têm vantagens e desvantagens.

O primeiro caso é mais simples, pois apenas uma operação de entrada/saída fica pendente a cada momento. Em compensação, limita a quantidade de operações simultâneas que podem ser feitas. O segundo caso é mais, complexo, pois, várias operações de entrada/saída podem ficar pendentes ao mesmo tempo e o sistema operacional precisa identificá-las para poder tratá-las.

Outro tipo de interrupções que merecem uma atenção especial são as interrupções de erro. Elas são geradas quando o programa que está sendo executado tenta realizar algo não permitido pelo sistema operacional. Por exemplo, realizar o acesso uma posição de memória protegida. Sempre que uma interrupção de erro ocorre e o sistema operacional deve terminar o programa em execução de forma anormal. Uma mensagem de erro é enviada e a área de memória utilizada pode ser copiada num arquivo (*dump*) para que o usuário possa tentar descobrir a causa do erro.

A figura 2.2 a seguir ilustra o fluxo geral do comportamento de um sistema operacional.



**Figura 2.2. Fluxo geral do sistema operacional**

## 2.4. Resumo

O sistema operacional oferece diversos serviços aos usuários. Num nível mais baixo, existem as chamadas ao sistema, que fornecem as funções básicas e permitem a um programa em execução fazer pedidos diretamente ao sistema operacional. Num nível mais alto, existem os programas utilitários. Entre eles se destaca o **interpretador de comandos**. O interpretador fornece ao usuário um meio mais amigável de requisitar serviços ao sistema. Neste nível um pedido satisfeito pelo interpretador de comandos ou por algum utilitário é traduzido numa sequência de chamadas ao sistema.

**Leitura Obrigatória:** Capítulo 1 do livro “Sistemas Operacionais Modernos”, 2ª edição, TANEMBAUM, A.