

AULA 01 - Introdução aos Sistemas Operacionais

Definições

1. O sistema operacional é uma camada de software colocada sobre o hardware para gerenciar todos os componentes do sistema (processadores, memória, discos, interface de rede, entre outros) de maneira otimizada. O sistema operacional busca oferecer uma interface mais amigável aos usuários em relação às operações associadas a cada componente da máquina.
2. O sistema operacional é um programa que atua como intermediário entre o usuário e o hardware de um computador. O propósito de um sistema operacional é oferecer um ambiente na qual os usuários possam executar programas. O principal objetivo de um sistema operacional é, portanto, tornar o uso do sistema de computação conveniente. Uma meta secundária é usar o hardware de forma eficiente.

Sempre que o usuário necessita realizar alguma operação sobre qualquer componente do sistema computacional ele recorre ao sistema operacional. Assim, não precisamos conhecer os detalhes associados a cada componente do hardware. A organização de um Sistema computacional pode ser vista na Figura 1.

Serviços oferecidos pelo sistema operacional

Irá depender do sistema operacional considerado. Entretanto, existe um conjunto básico de serviços oferecidos por quase todos os sistemas operacionais. Assim, um sistema operacional deve oferecer:

1. meios para que um programa seja carregado em memória e executado.
2. sistema de arquivos, permitindo criar, ler, escrever e destruir arquivos.
3. interface de acesso aos periféricos (impressoras, fitas, discos, etc).
4. mecanismos de monitoração de recursos, capazes de identificar possíveis gargalos no sistema.
5. meios para armazenar/manter o estado do sistema.
6. mecanismos de compartilhamento de hardware por usuários => necessidade de algum tipo de proteção.

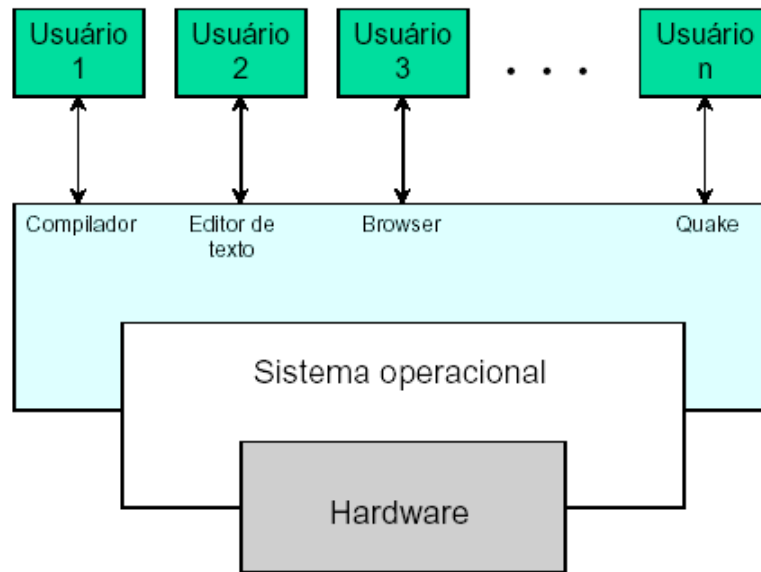


Figura 1. Abstração do Sistema [1].

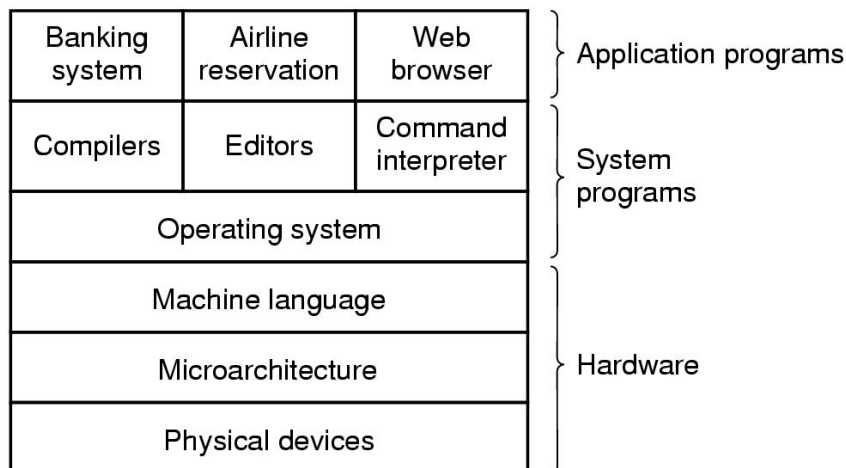


Figura 2. Visão Computacional [2].

Sistema Operacional na Visão do Usuário

Corresponde a imagem que o usuário tem do sistema operacional, definida pela interface através do qual o usuário acessa os serviços do sistema. Todo computador pode ser visto como uma máquina composta por níveis ou camadas. Nesta abordagem podemos visualizar o hardware como sendo a camada de mais baixo nível (nível 0), logo acima o sistema operacional (nível 1) e, posteriormente, a camada de aplicação (nível 2); conforme ilustrado nas Figuras 1 e 2.

Chamadas de Sistema

É a forma como os programas solicitam serviços ao SO e, por isso, é conhecida como a interface entre os programas que estão rodando na camada de aplicação e o sistema operacional. Toda chamada de sistema transfere a execução para o sistema operacional mudando o modo de execução de usuário para modo núcleo, por meio de uma instrução de *trap*. Com isso, o programa, através de um conjunto de parâmetros, informa o que ele realmente quer do SO. Após finalizada a execução da chamada de sistema, o controle é devolvido para o programa do usuário (muda-se do modo núcleo para usuário) na instrução seguinte à da chamada ao sistema. Com isso, o programa continua normalmente a execução de suas instruções. Existem vários tipos de chamadas ao sistema, cada uma com um propósito diferente. Por exemplo, uma divisão por zero no programa pode levar a uma situação de erro e, neste caso, se o problema não for tratado no próprio código da aplicação, o sistema operacional assume o controle e decide o que fazer.

Existem diferentes tipos de chamadas de sistemas e cada sistema operacional implementa um conjunto distinto relacionado as seguintes atividades:

1. chamadas de sistema para gerenciamento de processos: `fork`, `waitpid`, `execve`, `exit`, etc.
2. chamadas de sistema para gerenciamento de arquivos: `open`, `close`, `read`, `write`, `seek`, etc.
3. chamadas de sistema para gerenciamento do sistema de diretórios: `mkdir`, `rmdir`, `link`, `mount`, `umount`, etc.
4. chamadas de sistema diversas: `chmod`, `kill`, `time`, etc.

Kernel

O kernel ou núcleo é a parte do SO que implementa as chamadas ao sistema. As atividades normalmente atribuídas ao *kernel* são: (a) criação, agendamento e finalização de processos; (b) alocação e liberação de memória; (c) controle do sistema de arquivos; (d) operações de entrada e saída com dispositivos periféricos, acesso à memória, entre outros. A implementação do kernel pode ser organizada de diferentes maneiras, tais como:

1. **Monolítica:** BSD, Linux, Solaris, MS-DOS e derivados, incluindo Windows 95, Windows 98 e Windows ME.
2. **Híbrida:** BeOS, Windows NT, XNU.
3. **Microkernel:** Minix;
4. **Exokernel.**
5. **Camadas.**

Programas de Sistema

São programas normais, também chamado de utilitários, executados fora do *kernel* do SO que utilizam as chamadas de sistema para realizarem suas funções. Exemplos: *Windows Explorer*, *Interpretador de Comandos (DOS ou UNIX)* ou *Interface Gráfica de Usuário (Windows)*.

Sistema Operacional na Visão de Projeto

Na visão de projeto a preocupação é com a organização interna do SO, ou seja, como os serviços são implementados. Na solicitação de algum serviço corresponde a uma chamada de sistema que deverá ser tratada pelo SO quais serão as atividades envolvidas no processo e como serão implementadas. Exemplo: na leitura de um registro de arquivo em disco o SO envia comandos à controladora do disco e o programa espera a resposta. Concluída a leitura o SO é informado pela controladora e o programa pode retornar sua execução normal.

Exercícios

1. Como seria utilizar um computador sem um sistema operacional? Quais são suas duas principais funções?
2. Muitos sistemas operacionais atuais são projetados para operar em redes de computadores. Cite algumas vantagens e desvantagens dessa operação em rede, quando comparada com a operação clássica (*stand-alone*)?
3. Por quê a organização do sistema computacional em camadas é fundamental para o desenvolvimento e evolução dos elementos de hardware e também de software?
4. Pesquise e explique as formas como o kernel pode ser organizado.
5. Explique o que é uma chamada de sistema e pesquise exemplos que não foram citados no texto.

Bibliografia

- [1] OLIVEIRA, R. S.; CARISSIMI, A. S. e TOSCANI, S. S. *Sistemas Operacionais*. Sagra Luzzatto, Porto Alegre, 2001. (Cap. 1)
- [2] TANENBAUM, A. S. *Sistemas Operacionais Modernos*. Prentice Hall, 2ª Edição, 2003.