

## **Estruturas de Sistemas Operacionais**

Podemos analisar um sistema operacional sob diversos aspectos:

- Os serviços que o sistema operacional oferece.
- A interface que o sistema operacional torna disponível aos usuários e programadores.
- Os componentes do sistema operacional e suas interligações.

## **Componentes de um Sistema**

Um Sistema Operacional só pode ser desenvolvido se for subdividido em partes menores e mais simples. Cada uma dessas partes deve ser bem delineada, com entradas, saídas e comportamento bem definidas.

Obviamente, nem todos os sistemas têm a mesma estrutura.

### ➤ **Gerenciamento de Processos**

A execução de um programa envolve necessariamente a execução de suas instruções por uma CPU. Pode-se pensar em um processo como um programa em execução.

- Um serviço em um sistema batch é um processo.
- Um programa de usuário de um sistema de tempo compartilhado é um processo.
- Um tarefa de um sistema, como impressão pelo mecanismo de spooling (armazenamento temporário dos dados em disco), também é um processo.
- Chamadas ao sistema que permitam que os processos criem subprocessos a serem executados simultaneamente.

Um processo pode usar vários recursos para realizar uma tarefa, como tempo de CPU, memória, arquivos e dispositivos de E/S.

**Esses recursos podem ser alocados ao processo quando ele é criado ou durante sua execução.**

- Um programa por si só não é um processo !!!
  - É uma entidade passiva
  - Por exemplo: O conteúdo de um arquivo armazenado em um disco.
- Um processo é uma entidade ativa, com um contador de instruções que determina a próxima instrução a ser executada.
- Execução de processo deve progredir de maneira seqüencial.
  - A CPU executa uma instrução após a outra, até que o processo termine.

- A qualquer instante, apenas uma instrução de um dado processo é executada.
  - Duas seqüências diferentes de execução de um mesmo programa constituem dois processos distintos.
  - Um programa pode originar diversos processos.
- Um sistema é constituído de um conjunto de processos, sendo alguns desses do Sistema Operacional (rotinas do sistema) e os demais pertencentes aos usuários.
- Esses processos podem ser executados concorrentemente, por meio de um processo de multiplexação da CPU entre eles.

#### Sistema Operacional em relação ao gerenciamento de processos

- Criação e remoção de processos, tanto do sistema quando dos usuários.
- Suspensão e reativação de processos.
- Sincronização de processos.
- Comunicação entre processos.
- Tratamento de impasses entre processos.

#### ➤ Gerenciamento da Memória Principal

A memória principal tem papel central na operação de um sistema computacional moderno.

- É um repositório de dados de acesso rápido, compartilhado pela CPU e pelos dispositivos de E/S.
- As instruções a serem executadas pelo processador central são lidas a partir da memória principal durante o ciclo de busca de instruções do processador, assim como os dados são lidos e armazenados na memória principal durante o ciclo de busca de dados.
- Operações de E/S implementadas via DMA também lêem e escrevem dados na memória principal.
- **Para que a CPU processe dados armazenados em um disco, é preciso que esses dados primeiro sejam transferidos para a memória principal por chamadas a rotinas de E/S geradas pela CPU. Da mesma maneira, para que a CPU possa executar as instruções dos programas, elas devem estar armazenadas na memória.**
- Durante a execução de um programa, são feitos acessos aos endereços absolutos da memória, onde estão armazenadas as instruções e os dados do programa.

- Quando a execução termina, o espaço de memória ocupado pelo programa se torna disponível, podendo outro programa ser então carregado para execução.

#### Sistema Operacional em relação ao gerenciamento de memória principal

- Manter informação sobre quais partes da memória estão sendo usadas no instante atual e por quem.
- Decidir quais processos devem ser carregados na memória quando algum espaço de memória se torna disponível.
- Alocar espaço e remover programas e dados da memória.

#### ➤ Gerenciamento de Arquivos

Um dos componentes mais visíveis de um sistema operacional. Os computadores armazenam as informações em diversos tipos de meios físicos (fita magnética, disco magnético, disco óptico).

Para que um computador possa ser usado de maneira adequada, os sistemas operacionais fornecem uma visão lógica uniforme do armazenamento de informações. Eles abstraem, das propriedades físicas dos dispositivos de armazenamento, uma unidade lógica de armazenamento, **o Arquivo**.

- Associam arquivos a meios físicos e os acessam por intermédio dos dispositivos de armazenamento.
- Um arquivo é uma coleção de informações, relacionadas entre si, definidas na sua criação.
- Ao conceito abstrato de arquivo e às operações sobre arquivos correspondem, respectivamente, dispositivos como fitas, discos e dispositivos que os controlam, além das operações sobre esses dispositivos, gerenciadas pelo sistema operacional.

#### Sistema Operacional em relação ao gerenciamento de arquivos

- Criação e remoção de arquivos.
- Criação e remoção de diretórios.
- Suporte a primitivas para manipulação de arquivos e diretórios.
- Mapeamento de arquivos em memória secundária.
- Cópia de arquivos em meios de armazenamento não-voláteis.

➤ **Gerenciamento de Sistemas de E/S**

Um dos propósitos dos sistemas operacionais é esconder dos usuários peculiaridades de dispositivos de hardware específicos.

O subsistema de E/S é constituído de:

- Componente de gerenciamento de memória, que inclui o controle de usos diversos de áreas de armazenamento temporário.
- Interface geral para os diversos controladores de dispositivos.
- Rotinas de controle (drivers) dos diversos dispositivos.

Somente o controlador de um determinado dispositivo conhece suas peculiaridades específicas.

➤ **Gerenciamento de Memória Secundária**

O propósito principal de um Sistema Computacional é executar programas.

- Esses programas e os dados aos quais eles têm acesso devem estar na memória principal durante a execução.
- A memória principal é muito pequena.
- Deve existir uma memória secundária para auxiliar a memória principal.
- Os programas são armazenados em disco até serem carregados na memória e, depois, durante o processamento, usam o disco como fonte e destino de dados.

**Sistema Operacional em relação ao gerenciamento de memória secundária**

- Gerenciamento de espaço livre.
- Alocação de espaço na memória.
- Ordenamento e seleção das operações para o uso de discos.

➤ **Interpretadores de Comandos**

- Constitui a interface entre o usuário e o Sistema Operacional.
- Alguns Sistemas Operacionais incluem o interpretador de comandos (Shell) no seu núcleo (Kernel). Outros, tais como o MS-DOS e o UNIX, tratam o interpretador de comandos como um programa especial que está em execução quando um serviço é iniciado ou quando um usuário inicia uma sessão pela primeira vez.

- Sua função é bastante simples: obter o próximo comando e providenciar para que ele seja executado.
- A preferência dada por usuários a um determinado sistema é por vezes devida à interface amigável fornecida pelo interpretador de comandos desse sistema. Exemplo: Macintosh e Windows.
- Outros usuários não gostam de facilidades e preferem utilizar um simples terminal como interface. MS-DOS, UNIX.
- As especificações dos comandos podem tratar da criação e gerenciamento de processos, tratamento de E/S, gerenciamento de memória secundária, gerenciamento da memória principal, acesso ao sistema de arquivos, proteção e conexões em rede.

➤ **Serviços Implementados por Sistemas Operacionais**

Os sistemas operacionais criam um ambiente para a execução de programas. Os serviços implementados atendem às requisições de programas e aos seus usuários e podem variar de um sistema operacional para outro.

- Os serviços fornecidos são realizados para maior conveniência do programador, de modo a facilitar sua tarefa de programação:
- Execução de programas.
- Operações de E/S.
  - Por razões de eficiência e proteção, não é normalmente permitido aos usuários controlar diretamente os dispositivos de E/S. Portanto, o sistema operacional deve fornecer aos programas de usuários alguma maneira de realizar operações de E/S.
- Manipulação do sistema de arquivos.
- Comunicação
- Detecção de erros.
- Alocação de recursos.
- Contabilidade.
  - Os registros dos usuários que utilizam, e quanto, cada recurso do computador podem ser empregados como elementos contábeis (de modo que o uso dos recursos possa ser cobrado) ou simplesmente para acumular estatísticas de uso.
- Proteção

➤ **Chamadas ao Sistema**

A interação entre processos dos usuários e o sistema operacional é feita por meio de chamadas ao sistema.

Exemplo de como as chamadas ao sistema são usadas:

**Considere um programa simples que lê dados de um arquivo e os copia para outro arquivo.**

1. A primeira entrada que o programa necessitará são os nomes de dois arquivos: o arquivo de entrada e o de saída.
2. Uma abordagem é solicitar que o usuário forneça os nomes dos dois arquivos.
  - Em um sistema interativo, essa abordagem requer uma seqüência de chamadas ao sistema, primeiro para exibir na tela uma mensagem de solicitação de entrada de dados, depois para interpretar a partir do teclado os caracteres que definem os dois arquivos.
3. Obtidos os nomes dos arquivos, o programa precisa abrir o arquivo de entrada e criar o de saída.
  - Cada uma dessas operações requer uma chamada ao sistema. Existem também situações de erro, que podem ocorrer em cada uma dessas operações. Quando um programa tenta abrir o arquivo de saída, pode descobrir que não existe arquivo algum com aquele nome ou que o arquivo está protegido contra gravação. Nesse caso, ele deve imprimir uma mensagem no console(outra seqüência de chamadas ao sistema) e então provocar seu encerramento(outra chamada ao sistema). Se o arquivo de entrada existir, um novo arquivo de saída deve ser criado. Pode ser que já existe um arquivo com o mesmo nome. Essa situação pode abortar a execução do programa(uma chamada ao sistema) ou causar a remoção do arquivo existente(outra chamada ao sistema) e a criação de um novo arquivo(mais uma chamada ao sistema).
4. Estando preparados ambos os arquivos, é iniciado um ciclo de leitura de um arquivo entrada(uma chamada ao sistema) e de gravação em um arquivo de saída(outra chamada ao sistema). Cada operação de leitura e gravação deve retornar dados relativos a diversas condições possíveis de erro. Durante a leitura, o programa pode chegar ao final do arquivo ou pode ocorrer uma falha de hardware(como um erro de paridade). Durante a gravação podem ocorrer vários erros, dependendo do dispositivo de saída ( não há mais espaço em

disco, alcançou-se o fim da fita, a impressora está sem papel e assim por diante.

5. Finalmente, depois que todo o arquivo for copiado, o programa pode fechar ambos os arquivos(outra chamada ao sistema), escrever uma mensagem no console(mais chamadas ao sistema) e, por fim, terminar normalmente(a última chamada ao sistema).

Como podemos ver, os programas podem fazer uso freqüente do sistema operacional.

### ➤ **Programas de Sistema**

Esses programas estão situados em uma hierarquia dos componentes de um sistema computacional. No nível mais baixo está o hardware. O próximo é o sistema operacional, depois os programas de sistemas e finalmente os programas de aplicativos.

- Podem ser divididos em diversas categorias:
  - Manipulação de arquivos.
  - Informação de status.
  - Modificação de arquivos
  - Suporte às linguagens de programação.
  - Carregamento e execução de programas.
  - Comunicações.
  - Programas aplicativos.

### ➤ **Estrutura de Sistemas**

Um sistema tão grande e complexo quanto um sistema operacional moderno deve ser planejado cuidadosamente, se quisermos que ele funcione de maneira adequada e possa ser facilmente modificado. Uma abordagem comum consiste em decompor a tarefa em componentes menores, em vez de lidar com um sistema monolítico. Cada um desses módulos deve constituir uma parte bem definida do sistema, com entradas, saídas e funções determinadas cuidadosamente.

### ➤ **Estrutura Simples**

Existem vários sistemas disponíveis comercialmente que não têm uma estrutura bem definida. Freqüentemente, esses sistemas operacionais começaram como sistemas pequenos, simples e com recursos bem limitados e então cresceram além do alcance original. O MS-DOS é um exemplo. Ele foi projetado e implementado originalmente por poucas pessoas que não tinham a menor idéia de que ele iria se tornar tão popular. Ele foi escrito para fornecer o

máximo de recursos com uma utilização mínima de espaço, em virtude da limitação do hardware no qual ele era executado e portanto não foi dividido em módulos de maneira cuidadosa.

No MS-DOS, as interfaces e os níveis de funcionalidade não estão bem separados. Por exemplo, os programas aplicativos podem ter acesso às rotinas básicas de E/S para escrever diretamente na tela ou em unidades de disco. Essa liberdade deixa o MS-DOS vulnerável durante a execução de programas incorretos, podendo ocasionar grandes perdas e paradas do sistema. Como o Intel 8088, para o qual ele foi desenvolvido, não fornece um modo dual de operação e nenhum mecanismo de proteção, os projetistas do MS-DOS não tinham outra opção senão deixar livre o acesso ao hardware básico.

Outro exemplo em que não houve um projeto de uma estrutura bem definida ocorreu originalmente no caso do UNIX, outro sistema influenciado inicialmente pelas limitações do hardware. Ele é constituído de duas partes: o núcleo e os programas de sistema. O núcleo é dividido, por sua vez, em uma série de interfaces e rotinas de controle de dispositivos, que foram sendo adicionadas e expandidas ao longo dos anos à medida que o UNIX foi evoluindo. O núcleo é tudo aquilo que está acima do hardware e abaixo do nível de interface das chamadas ao sistema. Nele estão o sistema de arquivos, o sistema para alocação da CPU, o gerenciamento da memória e outras funções do sistema operacional realizadas pelas chamadas ao sistema. Em conjunto, é uma quantidade enorme de funções acumuladas em um único nível.

#### ➤ **Abordagem de divisão de níveis**

As novas versões do UNIX são projetadas para usar um hardware mais avançado. Com um suporte adequado do hardware, os sistemas operacionais podem ser decompostos em partes menores, da maneira mais apropriada. O sistema operacional pode manter um controle muito maior sobre o comportamento de um computador e das aplicações que fazem uso desse computador. Os implementadores têm mais liberdade para realizar mudanças no funcionamento das partes mais internas do sistema.

A modularização de um sistema pode ser feita de diversas maneiras; a mais atraente é a divisão em níveis(ou camadas), que consiste em dividir o sistema operacional em diversos níveis, cada um construído sobre os níveis inferiores. O nível mais baixo é o do hardware; o mais alto é o da interface com o usuário.

Cada nível esconde dos superiores o hardware e certas estruturas de dados e operações usados na sua implementação.

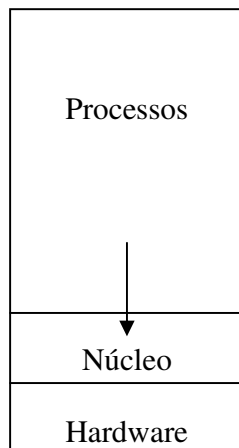


A abordagem de divisão em **níveis(ou camadas)** foi usada pela primeira vez no projeto de sistemas operacionais na Technische Hogeschool Eindhoven. O sistema THE foi definido em seis níveis.

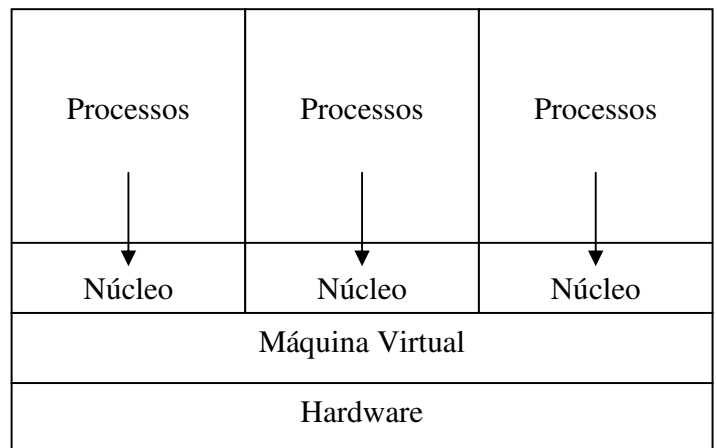
**A maior dificuldade existente com o uso da abordagem de divisão em níveis está relacionada a uma definição apropriada dos diversos níveis. Como um determinado nível só pode usar operações de níveis inferiores, é necessário um planejamento cuidadoso.**

Um outro problema com implementações em camadas é que elas tendem a ser menos eficientes do que as de outros tipos. Por exemplo, para que um programa de algum usuário execute uma operação de E/S, ele precisa fazer uma chamada ao sistema que está no nível de E/S, o qual chama operações do nível de gerenciamento de memória, passando pelo nível de alocação da CPU e finalmente chegando ao hardware.

➤ **Máquinas Virtuais**



Máquina não virtual



Máquina Virtual

Conceitualmente, um sistema computacional pode ser decomposto em níveis. O hardware é o nível mais baixo em todos os sistemas. O núcleo do sistema operacional, que é executado no próximo nível, utiliza as instruções do hardware para criar um conjunto de chamadas ao sistema, que podem então ser usadas nos níveis mais externos. Os programas de sistema no nível acima podem portanto usar tanto as chamadas ao sistema quanto as instruções para operação direta do hardware e, de certo modo, não fazem diferenciações entre elas. Assim, embora o acesso a elas seja diferente, ambas proporcionam uma funcionalidade que o programa pode usar para criar funções ainda mais avançadas. Os programas

de sistema, por sua vez, tratam as chamadas aos sistema e o hardware como se ambos estivessem em um mesmo nível.

**Alguns sistemas estendem esse esquema e permitem que os programas de sistema sejam chamados facilmente por programas aplicativos. Como antes, embora os programas de sistema estejam em um nível mais alto do que o de outras rotinas, os programas aplicativos podem ver tudo o que está embaixo deles na hierarquia, como se fossem partes da própria máquina.**

**A generalização dessa visão dá origem ao conceito de máquina virtual.** O sistema operacional VM da IBM é o melhor exemplo do conceito de máquina virtual, devido ao trabalho pioneiro da IBM nessa área.

Com o uso de técnicas de alocação de CPU e memória virtual um sistema operacional cria a ilusão de múltiplos processos em execução em seu processador com sua memória(virtual).

A abordagem de máquina virtual não oferece qualquer função adicional, mas uma interface idêntica à do hardware sob o qual todos os programas são executados. Assim, uma cópia(virtual) desse hardware fica disponível para cada processo.

Os recursos do computador são compartilhados para a criação das máquinas virtuais. A alocação da CPU pode ser usada para compartilhar a CPU e causar impressão de que os usuários têm seu próprio processador. Com o uso do mecanismo de spooling e de um sistema de arquivos, podem ser criadas leitoras de cartões e impressoras de linha virtuais.

Uma grande dificuldade da abordagem de máquina virtual tem a ver com os sistemas de disco. Suponha que o computador tenha três unidades de disco mas quer oferecer suporte a sete máquinas virtuais. Com certeza, ele não pode alocar uma unidade de disco a cada máquina virtual. O software da própria máquina virtual necessitará de bastante espaço em disco para implementação de spooling e memória virtual. A solução é fornecer discos virtuais, que são idênticos em todos os aspectos com exceção do tamanho. Esses discos são chamados de minidisks no sistema operacional VM, da IBM.

➤ **Implementação**

Embora o conceito de máquina virtual seja útil, ele é difícil de implementar. É necessário um esforço grande para obter uma cópia exata da máquina real. A máquina real pode operar em dois modos: modo usuário e modo monitor. O software da máquina virtual pode ser executado no modo monitor, uma vez que ele é o sistema operacional. As operações disponíveis na máquina virtual só podem ser executadas em modo usuário. Entretanto, a máquina virtual deve poder operar em dois modos, da mesma maneira como a máquina real. Conseqüentemente devemos ter um modo usuário virtual e um modo monitor virtual, ambos executando em modo usuário na máquina real. As ações que causam uma transferência de modo usuário para o modo monitor na máquina real( tais como uma chamada ao sistema ou uma tentativa de executar uma instrução privilegiada) devem também causar uma transferência de modo usuário virtual para o modo monitor virtual.

A principal diferença entre o máquina virtual e a máquina real é, o tempo de processamento. Enquanto a E/S real poderia levar 100 milissegundos, a E/S virtual poderia levar menos tempo(devido ao uso de spooling) ou mais (porque é interpretada). Além disso, a CPU está sendo usada em um sistema de multiprogramação, com várias máquinas virtuais, diminuindo assim o tempo de processamento de uma forma em geral muito difícil de prever. No caso extremo, pode ser necessário simular todas as instruções, para oferecer uma máquina realmente virtual.

➤ **Benefícios**

O conceito de máquina virtual apresenta muitas vantagens. Note que nesse ambiente há uma proteção completa dos vários recursos do sistema. Cada máquina virtual é completamente isolada de todas as outras e, portanto não há problemas de segurança. Por outro lado, não há um compartilhamento direto de recursos.

As máquinas virtuais estão sendo utilizados como meio de resolver problemas de compatibilidade entre sistemas. Por exemplo, existem milhares de programas disponíveis para o MS-DOS em computadores que usam CPUs da Intel. Fabricantes e comerciantes de computadores como a Sun Microsystems e a Digital Equipment Corporation (DEC) usam outros processadores, mais rápidos, mas gostariam que seus clientes pudessem executar esses programas para MS-DOS. A solução é criar uma máquina virtual Intel que execute no processador nativo. Um programa para MS-DOS pode então ser executado nesse ambiente, pois suas instruções Intel são convertidas em instruções do processador nativo. O MS-DOS é

também executado na máquina virtual, de modo que o programa pode fazer chamadas ao sistema de maneira usual. Obtém-se assim um programa que parece estar sendo executado em um sistema que usa um processador Intel, mas que na verdade está sendo executado em um processador bem diferente. Se esse processador for bastante rápido, o programa MS-DOS será executado rapidamente, apesar de cada instrução estar sendo convertida em diversas instruções nativas para execução.

➤ **Java**

Java é uma linguagem projetada pela Sun Microsystems, e sua implementação usa um compilador cuja saída é chamada de bytecode. Esses bytecodes são instruções executadas na máquina virtual Java. Para que os programas Java possam ser executados em uma plataforma, uma máquina virtual Java deve estar sendo executada nessa plataforma. A máquina virtual Java é implementada também em programas de navegação na Internet, como o Microsoft Explores e o Netscape. Esses navegadores, por sua vez, são executados em vários computadores, com vários sistemas operacionais.

➤ **Recapitulando**

Os sistemas operacionais oferecem diversos serviços. No nível mais baixo, as chamadas ao sistema permitem que um programa em execução faça requisições diretamente ao sistema operacional. Em um nível mais alto, o interpretador de comandos fornece um mecanismo para que um usuário faça uma requisição sem escrever um programa. Os comandos podem ser originados de arquivos, quando estão sendo executados em segundo plano, ou diretamente de um terminal, quando usam um modo interativo ou de tempo compartilhado. Os programas de sistema são usados para atender às requisições comuns dos usuários.

Os tipos das requisições variam de acordo com seu nível. O nível das chamadas ao sistema deve fornecer as funções básicas, como controle de processos e manipulação de arquivos e dispositivos. Requisições em níveis superiores, atendidas pelo interpretador de comandos ou programas de sistema, são traduzidas para uma seqüência de chamadas ao sistema. Os serviços fornecidos pelo sistema podem ser classificados em diversas categorias: controle de programas, requisições de informações gerais e requisições implícitas.

Depois que os serviços do sistema estão definidos, a estrutura do sistema operacional pode ser desenvolvida. Várias tabelas são

necessárias para registrar informações que definem o estado do sistema computacional e o estado dos serviços que estão sendo processados.

O projeto de um novo sistema operacional é uma tarefa difícil. É importante que os objetivos do sistema sejam bem definidos antes que o projeto seja iniciado. O tipo de sistema desejado é a base para as escolhas necessárias de algoritmos e estratégias.

Como para todo programa grande, a modularização é importante. O projeto do sistema como uma seqüência de níveis ou camadas é considerado uma técnica importante de projeto. O conceito de máquina virtual leva a abordagem de divisão em níveis ao extremo e trata o núcleo do sistema operacional e o hardware como se ambos fossem hardware. Até mesmo outros sistemas operacionais podem ser carregados na máquina virtual.

Por todo o ciclo de projeto do sistema operacional devemos ser cuidadosos em separar detalhes de implementação de decisões relativas a uma política adotada. Essa separação oferece maior flexibilidade no caso de mudanças posteriores nessas políticas.

Os sistemas operacionais são hoje em dia quase sempre escritos em uma linguagem de implementação de sistemas ou em uma linguagem de nível mais alto. Essa característica proporciona uma melhor implementação, manutenção e portabilidade. Para criar um sistema operacional para uma máquina com determinada configuração, devemos realizar um processo de configuração do sistema operacional.