



### 3 CONCEITOS DE SISTEMAS OPERACIONAIS

#### 3.1 COMPONENTES DO SISTEMA

Os Sistemas Operacionais são divididos em algumas partes básicas, conforme ilustrado na Figura 3.1.

1. Gerente de Processos;
2. Gerente de Memória;
3. Gerente de Arquivos;
4. Gerente de I/O;
5. Sistema de Proteção;
6. Interpretador de Comandos.

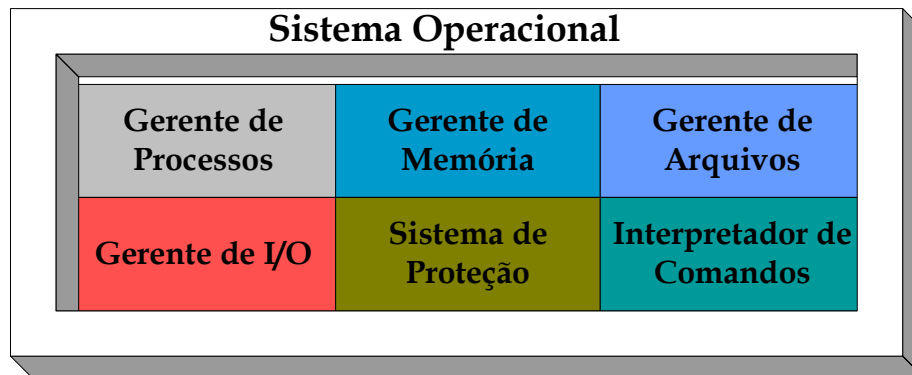


Figura 3-1 - Sistema Operacional

##### 3.1.1 GERÊNCIA DE PROCESSOS

Um processo é um programa em execução. Sua constituição é:

1. Código executável;
2. Pilha de execução;
3. Contador de programa;
4. Ponteiro de pilha;
5. Valores dos registradores;

O Sistema Operacional tem como principais funções:

- ☐ Criar e excluir os processos do usuário e do sistema;
- ☐ Suspender e retornar processos;



- ☐ Fornecer mecanismos para sincronizar os processos;
- ☐ Fornecer mecanismos para comunicação de processos;

As informações de cada processo são armazenadas na Tabela de Processos. Um processo pode criar outros processos. Dessa maneira, um processo que gera outro processo é chamado processo-pai. Já o processo que foi gerado a partir de outro processo é chamado processo-filho.

### 3.1.2 MEMÓRIA PRINCIPAL

A memória é o local onde as informações que são trabalhadas pelo processador ficam armazenadas. Ela pode ser considerada um grande vetor que armazena alguns milhões de caracteres. Cada palavra ou expressão possui seu próprio endereço.

O programa que é executado deve ter os endereços de memória das informações que irá utilizar compilados quando ele é carregado. É importante ressaltar que atualmente vários programas ficam residentes na memória, os quais possuem área própria e nem sempre estão com todo o seu código. Referente à gerência de memória, o sistema operacional tem as seguintes responsabilidades:

- ☐ Manter registro das partes da memória que estão sendo usadas no momento e por quem;
- ☐ Decidir que processos serão carregados na memória quando houver disponibilidade na mesma;
- ☐ Alocar e desalocar espaço na memória, conforme necessário;

### 3.1.3 GERÊNCIA DE ARQUIVOS

O gerenciador de Arquivos é uma outra parte importante de um Sistema Operacional. Ele é responsável pela criação, leitura, escrita e exclusão de um arquivo. Um arquivo pode ser definido como uma coleção de informações relacionadas definidas por seu criador. Geralmente, os arquivos representam programas e dados. Os arquivos de dados podem ser numéricos, alfanuméricos ou alfabéticos. O sistema operacional mapeia os arquivos nos dispositivos físicos de armazenamento permanente, abstraindo as suas características de forma a obter uma unidade lógica de armazenamento.

Atualmente os sistemas operacionais implementam o conceito de diretório para agrupar um conjunto de arquivos. Um usuário pode criar um diretório proprietário para armazenar seus arquivos. Também podem ser criados diretórios dentro de diretórios, cujo acesso é hierárquico.



Um usuário pode definir que tipo de acesso os demais usuários podem ter sobre os seus arquivos: Leitura, Escrita, Execução. Para tornar mais fácil o manuseio de dispositivos de I/O, alguns sistemas operacionais oferecem estruturas de arquivos especiais. Existem dois tipos de arquivos especiais:

- ☐ **Arquivos em Blocos:** o endereçamento é feito através de blocos de armazenamento. Exemplo: memória RAM;
- ☐ **Arquivos de Caractere:** o endereçamento é feito em uma cadeia de caracteres. Exemplo: fila de impressão.

### 3.1.4 GERÊNCIA DE I/O

Uma das funções do Sistema Operacional mais importantes para o usuário é ocultar as peculiaridades do manuseio dos dispositivos, o qual é desempenhado por um subsistema que possuem as seguintes características:

- ☐ Um componente de gerência de memória;
- ☐ Uma interface geral de *driver* do dispositivo;
- ☐ *Drivers* para dispositivos específicos.

### 3.1.5 SISTEMAS DE PROTEÇÃO

A multiprogramação foi um dos grandes avanços na busca por melhorar o desempenho dos sistemas computacionais. Entretanto, o fato de permitir o compartilhamento de recursos introduziu também uma série de pontos de conflitos. Foram necessários mecanismos para gerenciar o uso dos mesmos, assim como proteger o uso dos dispositivos que inclui espaço em memória, discos e CPU. Um programa só pode usar o seu espaço de endereçamento.

O esquema de proteção também inclui a autenticação de usuário e os recursos que tem direito de usar. Esta proteção também é importante para detectar erros evitando que eles se propaguem.

### 3.1.6 INTERPRETADOR DE COMANDOS

O interpretador de comandos é a interface entre o usuário e o sistema operacional. Através dele que o usuário realiza as chamadas de sistema.

## 3.2 SERVIÇOS DO SISTEMA OPERACIONAL

Um sistema operacional fornece serviços a programas e usuários. Estes serviços variam de um sistema operacional para outro, sendo oferecidos para maior comodidade dos usuários. Mas, existem os serviços padrões que diferem somente na forma que são ofertadas:



- ❑ *Execução do programa*: O Sistema Operacional deve ser capaz de carregar um programa na memória e executá-lo. Da mesma maneira, ele deve ser capaz de encerrá-lo, seja diretamente ou indiretamente;
- ❑ *Operações de I/O*: É muito provável que um programa necessite de usar algum dispositivo de I/O. Para dispositivos específicos pode haver funções específicas, os quais não são acessíveis diretamente pelo usuário como forma de proteção;
- ❑ *Manipulação do Sistema de Arquivos*: É responsável pela manipulação direta dos arquivos. As operações de leitura, gravação, criação e exclusão dos arquivos são realizadas pelo Sistema Operacional;
- ❑ *Comunicação*: Não é raro um processo trocar informações com outro processo. Uma das maneiras de fazer isso é usar uma memória compartilhada pelos processos;
- ❑ *Deteção de Erros*: O Sistema Operacional deve estar em constante vigilância quanto aos erros. Erros ocorrem tanto na CPU quanto na memória e, principalmente, no programa do usuário. Para cada tipo de erro, o Sistema Operacional deve ter um procedimento adequado;
- ❑ *Alocação de Recursos*: Na execução de vários processos, recursos devem ser alocados para cada um deles. Vários recursos são administrados pelo Sistema Operacional e, alguns deles, possuem um código de alocação especial;
- ❑ *Contabilização*: Ele mantém um registro do uso dos recursos e de quem são os usuários. Essas informações nem sempre são usadas pelo sistema;
- ❑ *Proteção*: O Sistema Operacional deve garantir a integridade dos dados armazenados por cada usuário. Entende-se por usuário um programa ou pessoa. Nem um processo deve interferir indevidamente com outro;

### 3.3 CHAMADAS DE SISTEMA

A execução de programas é feita através de chamadas de sistema, que são a interface entre o processo e o Sistema Operacional. Elas são construídas a partir de instruções em linguagem de montagem (*assembly*). Elas foram idealizadas com objetivo de não permitir o acesso direto do usuário no sistema, ou seja, faz parte do esquema de proteção do Sistema Operacional.

Alguns Sistemas Operacionais permitem que as chamadas sejam feitas diretamente das linguagens de alto nível. Linguagens como o C++, C e Perl substituem o *assembly* na programação de sistemas, e fazem chamadas de forma



direta. O Java não permite, por ela não estar diretamente ligada ao Sistema Operacional.

As operações mais simples executadas pelo usuário são formadas por uma série de chamadas de sistemas. Vamos tomar como exemplo a cópia de um arquivo para outro.

### **EXEMPLO:**

- ✓ Uma chamada para definir os nomes que definem os dois arquivos;
- ✓ É aberto o arquivo de entrada e criado o de saída;
- ✓ Para abrir o arquivo deve ser verificado se o usuário tem permissão para isso ou mesmo se o programa já está em uso por outro processo. Se estiver deve ser enviada uma mensagem informando a impossibilidade momentânea da operação;
- ✓ Se o nome dado ao arquivo de saída já pertencem a um outro programa, o Sistema questiona se o usuário deseja substituir;
- ✓ É iniciado um *loop* para leitura dos dados e sua posterior gravação;
- ✓ Cada leitura/escrita deve retornar informações de *status* de erros possíveis, devido às características dos dispositivos;
- ✓ Após o último caractere ser copiado, os dois arquivos são fechados.

Figura 3-2 - Exemplo

Os usuários, geralmente, desconhecem este processo e elas podem ocorrer de várias formas, pois a interface com as linguagens de programação é mais simplificada. As chamadas de sistema podem ocorrer de várias formas. Não basta simplesmente identificar a chamada. Elas são agrupadas em cinco categorias:

1. Controle de processo;
2. Manipulação de arquivos;
3. Manipulação de dispositivos;
4. Manutenção de arquivos;
5. Comunicações;

### **3.3.1 CONTROLE DE PROCESSO**

Um programa precisa ser capaz de suspender sua execução de forma normal ou anormal. As chamadas de sistemas de controle permitem armazenar uma mensagem indicando o erro ocorrido para posterior análise.



Programas podem carregar outros programas. Existem chamadas que possibilitam a execução de um programa por intermédio de outro. Isto permite que um usuário digite o nome de um programa a ser executado.

Se for criado um novo processo, o sistema operacional deve fornecer mecanismos para controlar o mesmo. Isto inclui a capacidade de determinar e redefinir os atributos de um processo, incluindo prioridade e tempo de execução.

Existem primitivas que bloqueiam a execução de processos por um determinado período de tempo ou que permitem fazer o histórico da execução do programa, registrando quais partes são executadas e quais interrupções são geradas.

- ☐ *End, abort;*
- ☐ *Load, execute;*
- ☐ *Create process, terminate process;*
- ☐ *Get process attributes, set process attributes;*
- ☐ *Wait for time;*
- ☐ *Wait event*
- ☐ *Allocate and free memory.*

### 3.3.2 MANIPULAÇÃO DE ARQUIVOS

As chamadas usadas pelo gerenciador de arquivos serão estudadas no capítulo destinado à gerência de arquivos. Mas para ilustrar, as chamadas de sistemas são:

- ☐ *Create file, delete file;*
- ☐ *Open, close;*
- ☐ *Read, write, reposition;*
- ☐ *Get file attributes, set file attributes.*

### 3.3.3 GERÊNCIA DE DISPOSITIVOS

Assim como na manipulação de arquivos, quando um processo precisa de um recurso adicional, ele recorre às chamadas de sistema que são descritas a seguir:

- ☐ *Request device, release device;*
- ☐ *Read, write, reposition;*
- ☐ *Get device attributes, set device attributes;*



- ☐ *Logically attach, detach devices.*

### 3.3.4 MANUTENÇÃO DE INFORMAÇÕES

É comum que os programas utilizarem algumas informações residentes na CPU e dispositivos como, por exemplo, data e hora do sistema.

- ☐ *Get time, get date, set time, set date;*
- ☐ *Get system data, set system data;*
- ☐ *Get process, file, device attributes;*
- ☐ *Set process, file, attributes.*

### 3.3.5 COMUNICAÇÃO

Este tópico será abordado na disciplina de Sistemas Distribuídos, mas as primitivas usadas são:

- ☐ *Create, delete communication connection;*
- ☐ *Send, receive messages;*
- ☐ *Transfer status information;*
- ☐ *Attach, detach remote devices.*

## 3.4 PROGRAMAS DE SISTEMA

O sistema operacional não trabalha sozinho. Existem outros programas de sistemas, cuja função é tornar o ambiente de computação mais amigável. Eles são divididos nas seguintes categorias:

- ☐ *Gerência de arquivos:* esses programas são específicos para manipulação de arquivos;
- ☐ *Informações de status:* faz o gerenciamento do ambiente de trabalho dos usuários;
- ☐ *Modificação de arquivo:* compreende os editores de texto de arquivos;
- ☐ *Suporte às linguagens de programação:* compiladores, montadores e interpretadores;
- ☐ *Carregamento e execução de programas:* carrega e executa os programas;
- ☐ *Comunicações:* permitem a comunicação entre usuários e processos.

Existem também os programas aplicativos que são usados para fins do usuário como, por exemplo, editores de texto, planilhas e sistemas de banco de dados.



### 3.5 ESTRUTURA DO SISTEMA

O Sistema operacional é bastante complexo, como pode ser concluído a partir da breve análise dos tópicos anteriores. Tanto que é dividido em partes, conforme a Figura 3.1. No intuito de melhor organizar o sistema operacional, existe uma série de formas de construção do Sistema Operacional.

#### 3.5.1 ESTRUTURA SIMPLES

Vários sistemas operacionais não possuem divisão, principalmente os primeiros. Eles foram construídos como blocos únicos, cujas melhorias foram inseridas ao longo de suas versões como, por exemplo, o MS-DOS. Ele foi construído para funcionar no menor espaço possível do disco. A estrutura do MS-DOS é ilustrada na Figura 3.3.

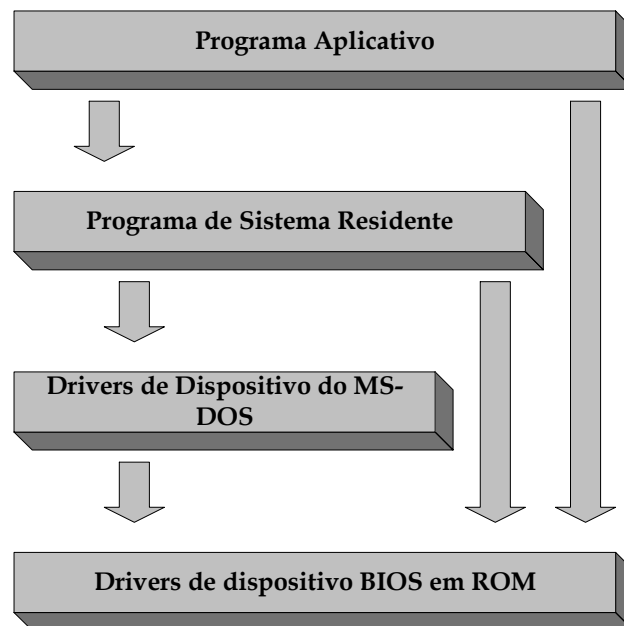


Figura 3-3 - Estrutura do MS-DOS

Outro sistema que possui estrutura simplificada é o UNIX. Este sistema merece uma atenção especial. Ele é dividido em duas partes: *kernel* e programas de sistema. O *kernel* ainda possui uma série de divisões internas. Todos os recursos do sistema são oferecidos através de chamadas de sistema. A sua estrutura é ilustrada na Figura 3.4.



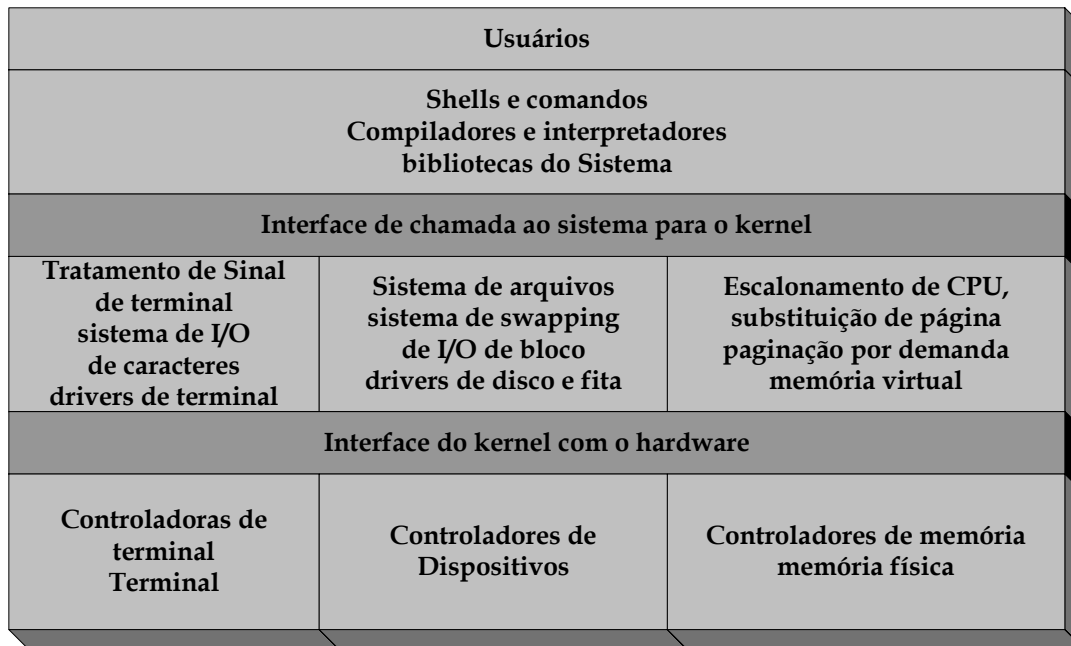


Figura 3-4 - Estrutura do UNIX

## 3.5.2 ABORDAGEM EM CAMADAS

A modularização de um sistema pode ser feita de diversas formas. Uma das mais utilizadas é a divisão em camadas. A modularidade dessa metodologia permite que uma camada use serviços e recursos apenas das camadas inferiores, o que simplifica a depuração e verificação do sistema. Cada camada é implementada apenas usando as operações das camadas inferiores, não sendo necessário o conhecimento do processo da operação usada.

A abordagem em camadas é bastante eficiente, mas requer muito cuidado com o planejamento, exatamente pelo fato da comunicação ser apenas com a camada inferior.

Outro ponto deficiente é que o fato de passar pelas camadas torna os sistemas construídos nessa abordagem menos eficiente que os demais. Um sistema que foi implementado com esta filosofia é a primeira versão do Windows NT e do OS/2, cuja estrutura está ilustrada na Figura 3.5.

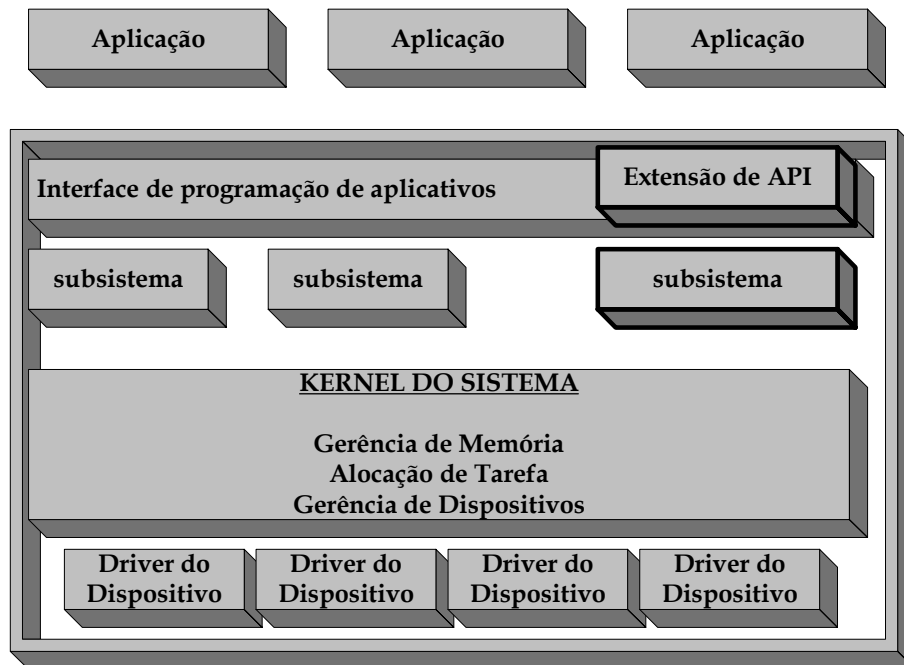


Figura 3-5 - Estrutura do OS/2

### 3.5.3 MICROKERNELS

O desenvolvimento do UNIX deixou seu *kernel* bastante grande. Com o objetivo de aumentar a eficiência do sistema, foi criada a metodologia de *microkernel*. O ponto chave dessa metodologia é que todas as partes não essenciais foram transformadas em programas de sistema, deixando o *kernel* enxuto. Não existe um padrão de partes essenciais ou não, o que torna a estrutura dos sistemas, que adotam essa metodologia, diferentes entre si.

A principal função do *microkernel* é permitir a comunicação dos usuários com os programas de sistema. Esta abordagem permite a expansão do Sistema Operacional sem degradar significativamente o seu desempenho. Um exemplo bastante claro é o ambiente cliente-servidor do Windows NT, conforme é ilustrado na Figura 3.6.

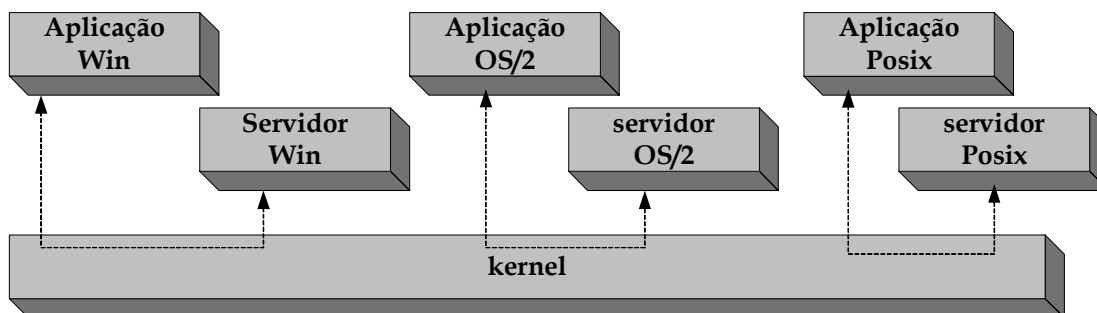


Figura 3-6 - Estrutura Cliente Servidor do Windows NT



### 3.6 MÁQUINA VIRTUAL

Um sistema computacional é formado por camadas, onde o mais baixo é representado pelo hardware. Alguns sistemas permitem que os programas de sistema sejam chamados pelos aplicativos, podendo fazer chamadas de sistema e usar instruções do hardware. Nesta abordagem, os programas aplicativos têm visão das camadas inferiores como se fossem parte da máquina. Dessa maneira, os processos podem assumir que cada um possui uma máquina dedicada, embora estejam compartilhando a mesma máquina física. Esta abordagem é chamada de **máquina virtual**.

Usando vários recursos, o escalonador de processo simula que cada processo possui seu próprio processador, memória e dispositivos. É importante deixar claro que as máquinas virtuais não aumentam o desempenho do sistema, ele apenas divide os recursos e cria a ilusão de máquinas dedicadas. Cada usuário possui sua própria máquina virtual, como é ilustrado na Figura 3.7.

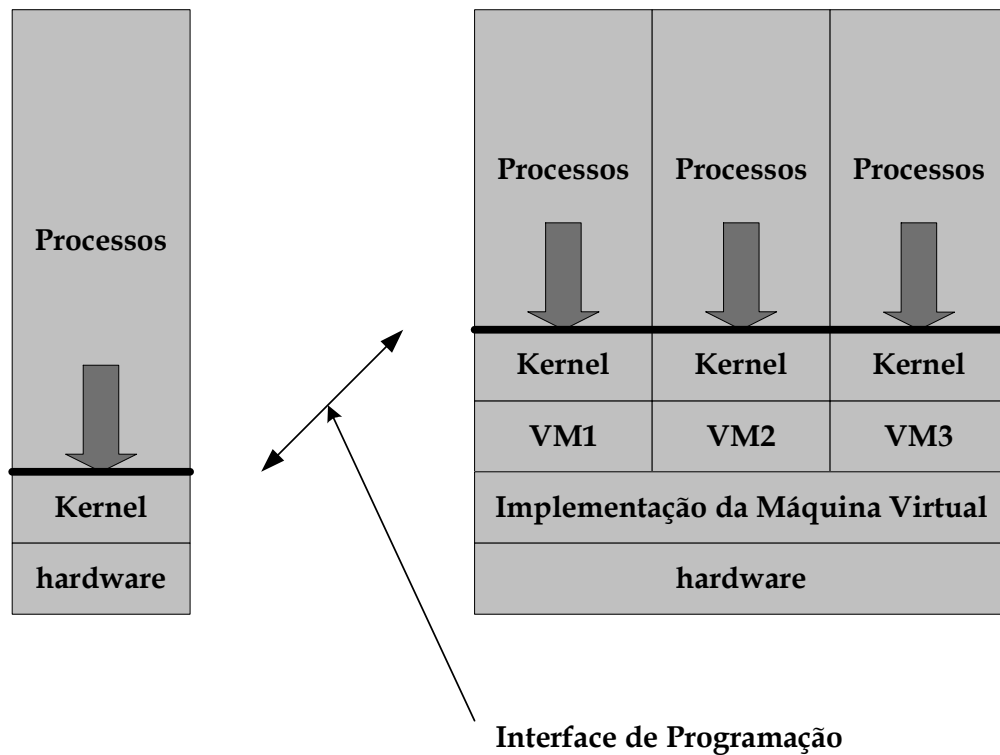


Figura 3-7 - Máquina Virtual

#### 3.6.1 IMPLEMENTAÇÃO

O conceito de máquina virtual é bastante útil, mas extremamente difícil de implementar. Existe muito trabalho para duplicar uma máquina de forma lógica. Um ponto importante é que a máquina virtual é que ela executa no modo usuário,



mas internamente ela possui o modo usuário e monitor, que devem funcionar como em uma máquina real.

No modo monitor, é possível alterar os valores contidos nos registradores virtuais, os contadores de programa. O desempenho da máquina como um todo, quanto a velocidade diminui, pois foi adicionado uma camada. Esta afirmação é facilmente verificada como verdadeira, basta analisar o desempenho de um programa construído em Java.

### 3.6.2 BENEFÍCIOS

O conceito de máquinas traz uma série de vantagens que serão descritas a seguir:

1. Proteção total aos vários recursos do Sistema, pois cada máquina virtual é totalmente isolada das demais;
2. O compartilhamento dos recursos é possível compartilha-se o disco através de software ou implantando uma rede de máquinas virtuais;
3. É uma opção para pesquisa em sistemas operacionais, pois ele simula uma máquina em software. Se houver erros apenas aquela máquina virtual será afetada;

### 3.7 EXERCÍCIOS

1. Quais as principais atividades de um sistema operacional, como gerente de processos? Qual a sua importância?
2. Quanto à memória? Qual a sua importância?
3. Qual a razão para a divisão do sistema operacional em várias partes?
4. Qual a importância das chamadas de sistema?
5. Quais as razões para o surgimento da abordagem de microkernel?
6. Faça uma analogia entre a abordagem microkernel e o conceito de máquinas virtuais.