

Estruturas de Sistemas Operacionais

- Componentes de Sistemas Operacionais
- Serviços de Sistemas Operacionais
- Chamadas ao Sistema (System Calls)
- Programas do Sistema
- Estrutura do Sistema
- Máquinas Virtuais
- Projeto e Implementação de Sistemas
- Geração de Sistemas

Referência: Silberschatz & Galvin, Sistemas Operacionais: Conceitos, Capítulo 3.

Componentes Comuns de Sistemas Operacionais

- Gerenciamento de Processos
- Gerenciamento de Memória Principal
- Gerenciamento de Arquivos
- Gerenciamento de E/S
- Gerenciamento de Memória Secundária
- Redes
- Sistema de Proteção
- Sistema de Interpretação de Comandos

Gerenciamento de Processos

- Um processo é um programa em execução. Um processo precisa de certos recursos, tais como tempo de CPU, memória, arquivos e dispositivos de E/S, para efetuar sua tarefa.
- O sistema operacional é responsável pelas seguintes atividades de gerenciamento de processos:
 - ◆ Criação e destruição de processos.
 - ◆ Suspensão e readmissão de processos
 - ◆ Disponibilidade de mecanismos para:
 - ✓ Sincronização de processos
 - ✓ Comunicação entre processos

Gerenciamento de Memória Principal

- Memória é um grande vetor de palavras(ou bytes), cada qual com seu próprio endereço. Ela é um repositório de dados compartilhados facilmente acessáveis pela memória e dispositivos de E/S.
- Memória principal é um dispositivo volátil. Ela perde seu conteúdo em caso de desligamento ou falha de sistema.
- São tarefas do sistema operacional quanto ao gerenciamento de memória principal:
 - ◆ Controlar que partes da memória estão sendo utilizadas e por quem.
 - ◆ Decidir que processos carregar quando a memória se torna disponível.
 - ◆ Alocar e liberar memória quando necessário.

Gerenciamento de Arquivos

- Um arquivo é uma coleção de informações relacionadas, definidas pelo seu criador. Comunmente, arquivos representam programas (fonte e objetos) e dados.
- O gerenciamento de arquivos de um sistema operacional é responsável por:
 - ◆ Criação e remoção de arquivos.
 - ◆ Criação e remoção de diretórios.
 - ◆ Suporte de primitivas para manipulação de arquivos e diretórios.
 - ◆ Mapear arquivos no sistema de armazenamento secundário.
 - ◆ Backup em meio de armazenamento estável e não volátil.

Gerenciamento de E/S

- O sistema de E/S consiste de:
 - ◆ Um sistema de cache em buffer
 - ◆ Uma interface com device drivers
 - ◆ Drivers para dispositivos de hardware específicos

Gerenciamento de Armazenamento Secundário

- Como a memória principal (armazenamento primário) é volátil e pequena demais para acomodar todos os dados e programas permanentemente, o sistema precisa disponibilizar armazenamento secundário para backup da memória principal.
- A maioria dos sistemas modernos usa discos como meio de armazenamento padrão, tanto para programas quanto para dados.
- Tarefas do gerenciamento de armazenamento secundário:
 - ◆ Gerenciamento de espaço livre
 - ◆ Alocação
 - ◆ Escalonamento dos discos

Redes (Sistemas Distribuídos)

- Um sistema distribuído é uma coleção de processadores que não compartilham nem memória nem relógio(clock). Cada processador tem sua própria memória local.
- Os processadores são conectados através de uma rede de comunicação.
- Regras de comunicação são definidas por um protocolo.
- Um sistema distribuído permite um usuário acessar vários recursos de sistemas.
- O acesso a recursos compartilhados permite:
 - ◆ Aceleração de computação
 - ◆ Aumento da disponibilidade de dados
 - ◆ Aumento de confiabilidade

Sistema de Proteção

- *Proteção* refere-se ao mecanismo de controle de acesso por programas, processos ou usuários, tanto pelo lado do sistema quanto dos recursos do usuário.
- O mecanismo de proteção precisa:
 - ◆ Distinguir entre usos autorizados e não autorizados.
 - ◆ Especificar que controles serão impostos.

Sistema de Interpretação de Comandos

- Muitos comandos são dados para o sistema operacional por blocos de controle relacionados com:
 - ◆ Criação e gerenciamento de processos
 - ◆ Manipulação de E/S
 - ◆ Gerenciamento de armazenamento secundário
 - ◆ Gerenciamento de memória principal
 - ◆ Acesso ao sistema de arquivos
 - ◆ Proteção
 - ◆ Redes

Sistema de Interpretação de Comandos(Cont.)

■ O programa que lê e interpreta blocos de controle recebe alguns nomes:

- ◆ Interpretador em linha de comando
- ◆ shell (no UNIX e LINUX)

Sua função é localizar e executar o próximo bloco de comando.

Serviços de um Sistema Operacional

- Execução de programas
- Operações de E/S
- Manipulação do sistema de arquivos
- Comunicações
- Detecção de erros

Funções adicionais

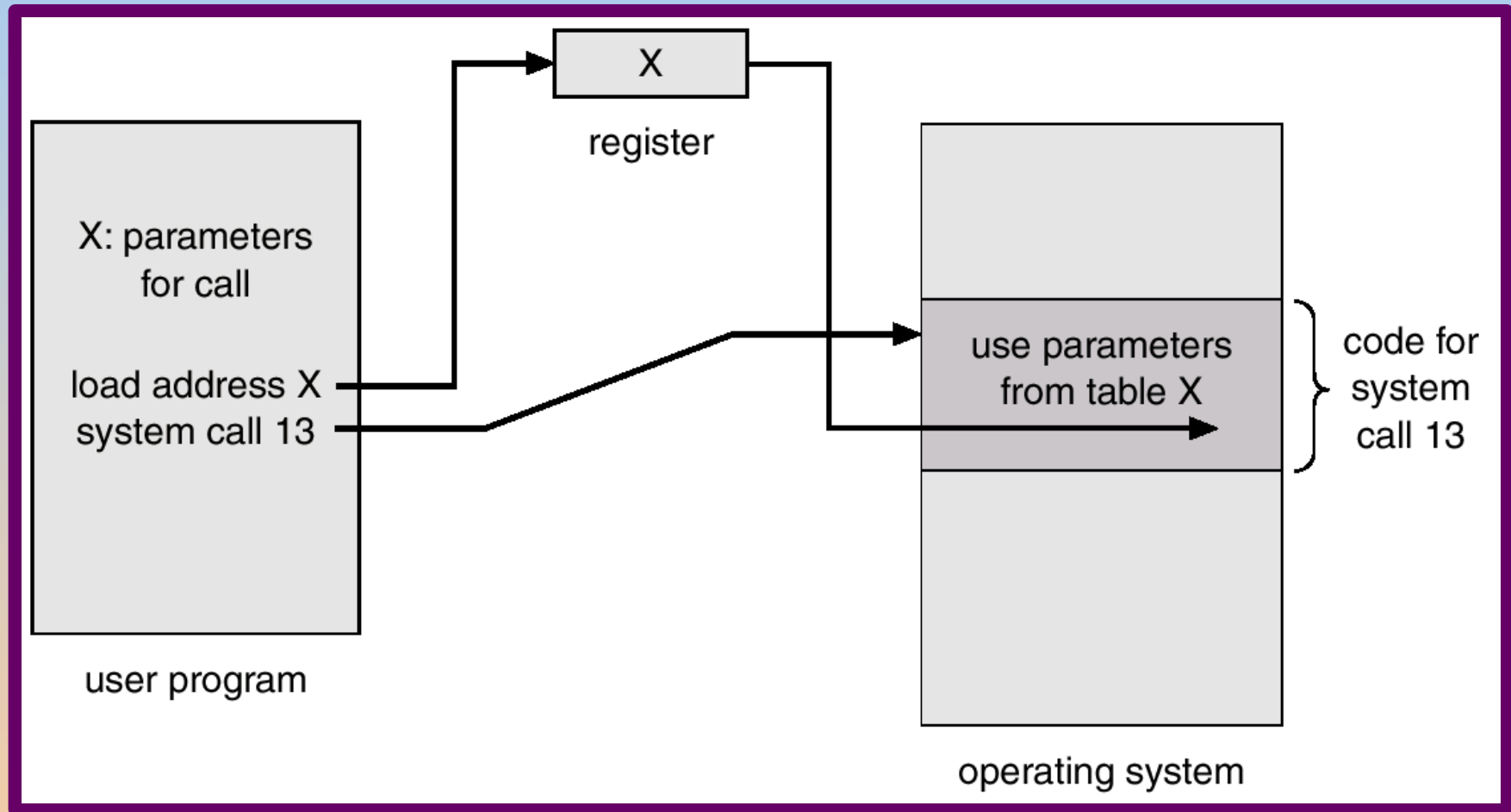
Funções adicionais existem não para ajudar o usuário, mas para garantir operações eficientes do sistema:

- Alocação de recursos – alocação de recursos para múltiplos usuários ou tarefas rodando simultaneamente.
- Gerenciamento de contas – manter registros de utilização dos vários recursos de sistemas pelos usuários.
- Proteção – garantir que todos os acessos ao recursos do sistema são controlados.

Chamadas ao Sistema(System Calls)

- System calls disponibilizam a interface entre um programa em execução e o sistema operacional:
 - ◆ São disponibilizadas, normalmente, como instruções em linguagem Assembly.
 - ◆ Linguagens definidas para substituir a linguagem Assembly para programação de sistemas permitem que system calls possam ser feitas diretamente. (por exemplo, C, C++)
- Três métodos gerais são usados para passar parâmetros entre um programa em execução e o sistema operacional:
 - ◆ Passar parâmetros em registradores
 - ◆ Armazenar os parâmetros em uma tabela na memória. O endereço da tabela é passado como um parâmetro em um registrador.
 - ◆ *Empilhar(push)* os parâmetros numa pilha(stack) pelo programa. O sistema operacional desempilha(pop) os parâmetros.

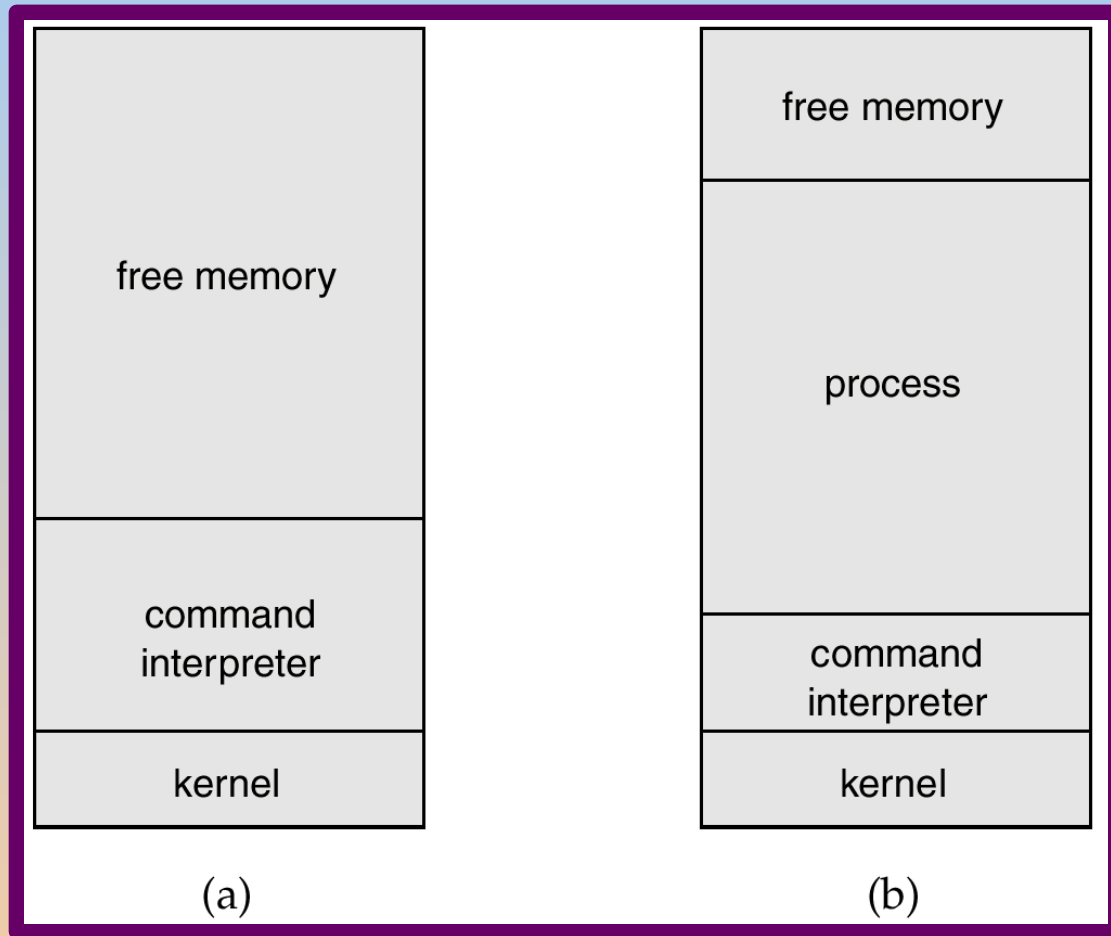
Passagem de parâmetros como uma tabela



Tipos de System Calls

- Controle de processos
- Gerenciamento de arquivos
- Gerenciamento de dispositivos
- Comunicações

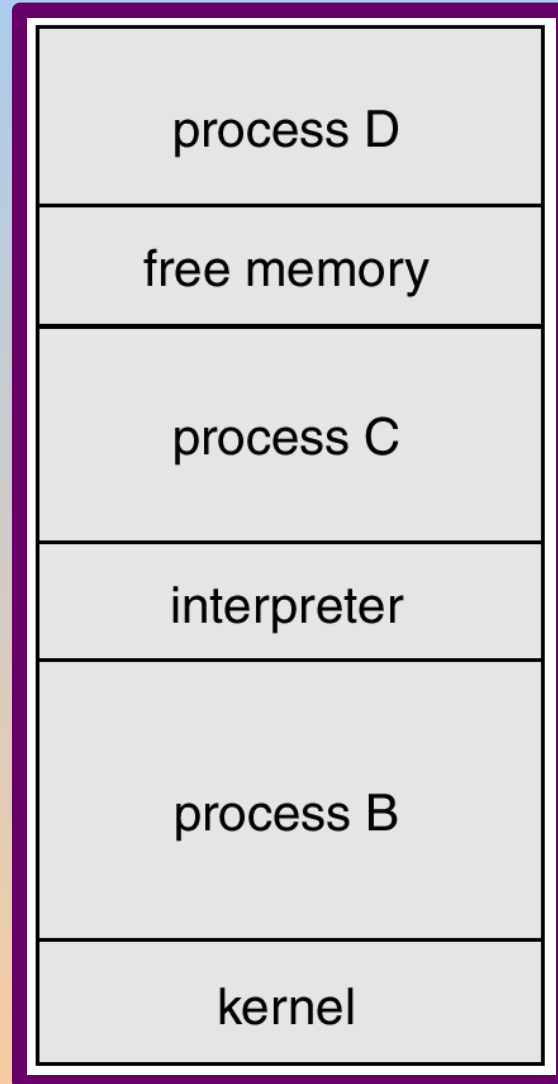
Execução no MS-DOS



Após carga do sistema

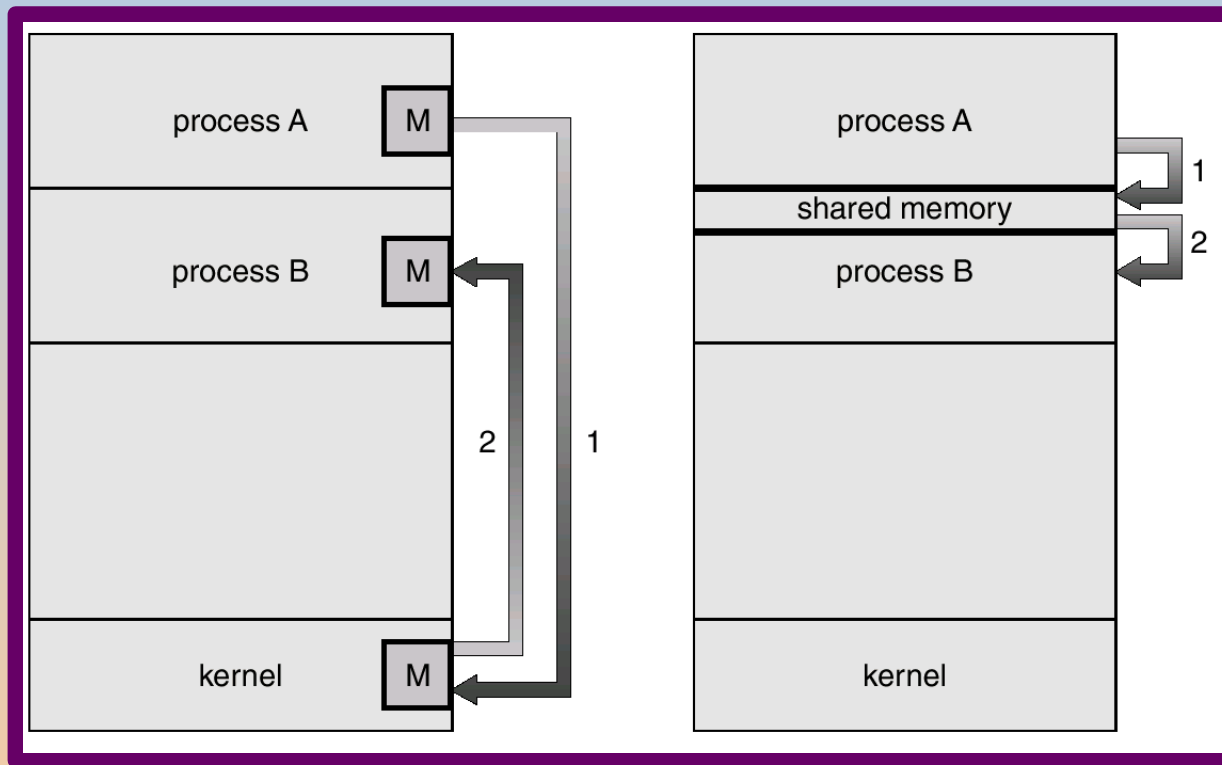
Rodando um programa

UNIX rodando múltiplos programas



Modelos de Comunicação entre processos

- Comunicação entre processos pode ser implementada por passagem de parâmetros ou por memória compartilhada.



Passagem de Parâmetros

Memória Compartilhada

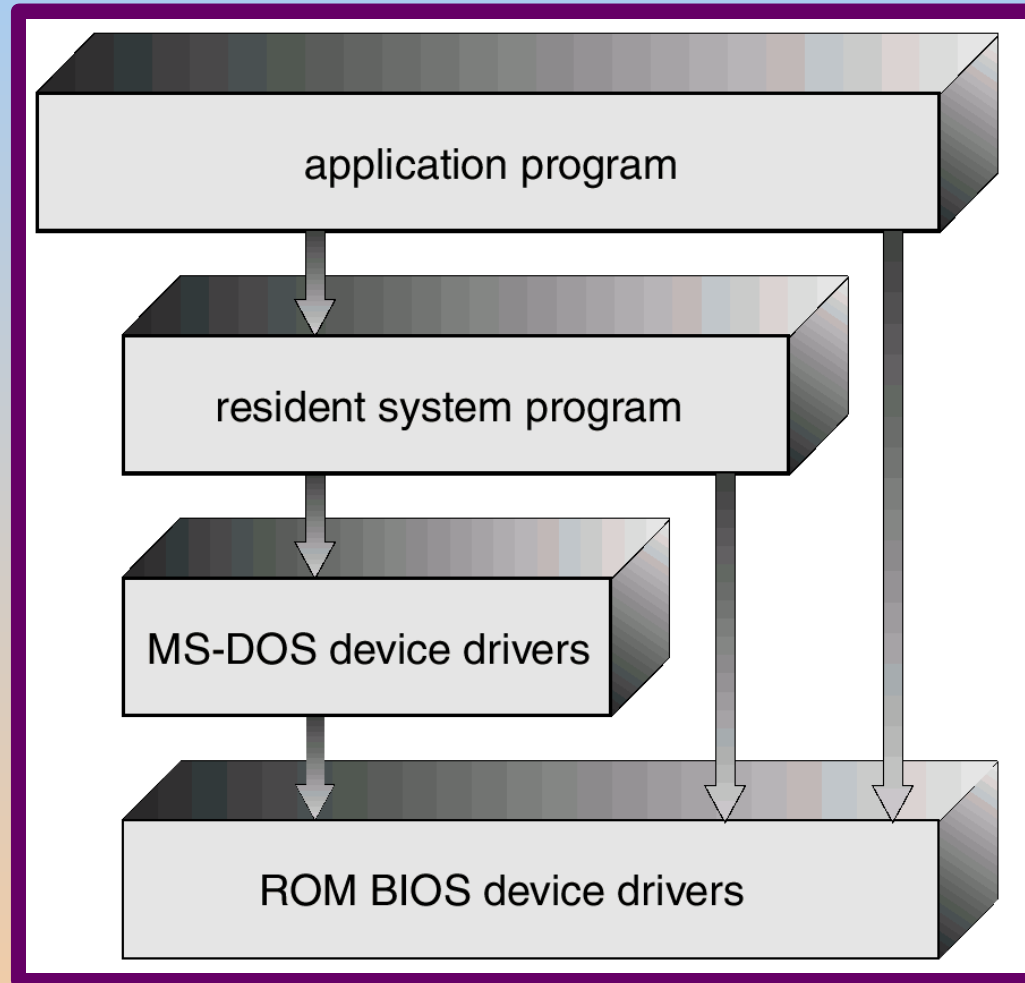
Programas do Sistema

- Programas do sistema disponibilizam um ambiente conveniente para desenvolvimento e execução de programas. Eles podem ser divididos em:
 - ◆ Manipulação de arquivos
 - ◆ Informação de status
 - ◆ Modificação de arquivos
 - ◆ Suporte para linguagens de programação
 - ◆ Carregamento e execução de programas
 - ◆ Comunicações
 - ◆ Programas de aplicação
- A visão de sistema operacional da maioria dos usuários é formada pelos programas de sistema, não das system calls.

Estrutura do MS-DOS

- MS-DOS – escrito para a funcionalidade máxima no menor espaço de memória:
 - ◆ Não era dividido em memória
 - ◆ Apesar do MS-DOS ter alguma estrutura, suas interfaces e níveis de funcionalidade não eram bem separados.

Estrutura de camadas do MS-DOS



Sistema UNIX

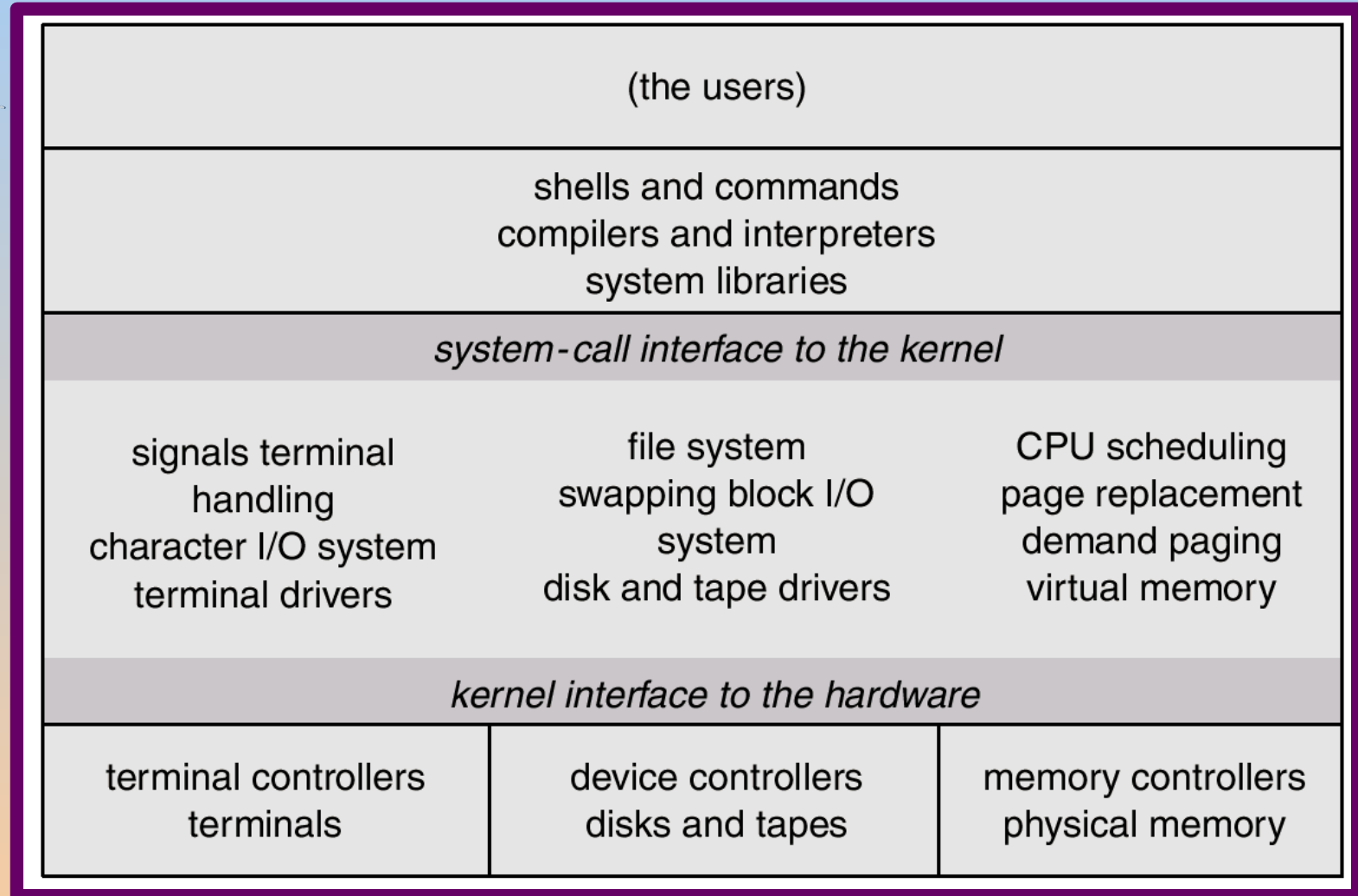
- UNIX – limitado pela funcionalidade do hardware, o UNIX original tem estruturação limitada. O sistema UNIX consiste de duas partes separadas:

- ◆ Programas de sistema

- ◆ O kernel (núcleo)

- ✓ Consiste de quase tudo abaixo da interface de system call e acima da camada de hardware.
- ✓ Disponibiliza o sistema de arquivos, escalonamento de CPU, gerenciamento de memória e outras funcionalidades; grande parte das função é de um nível.

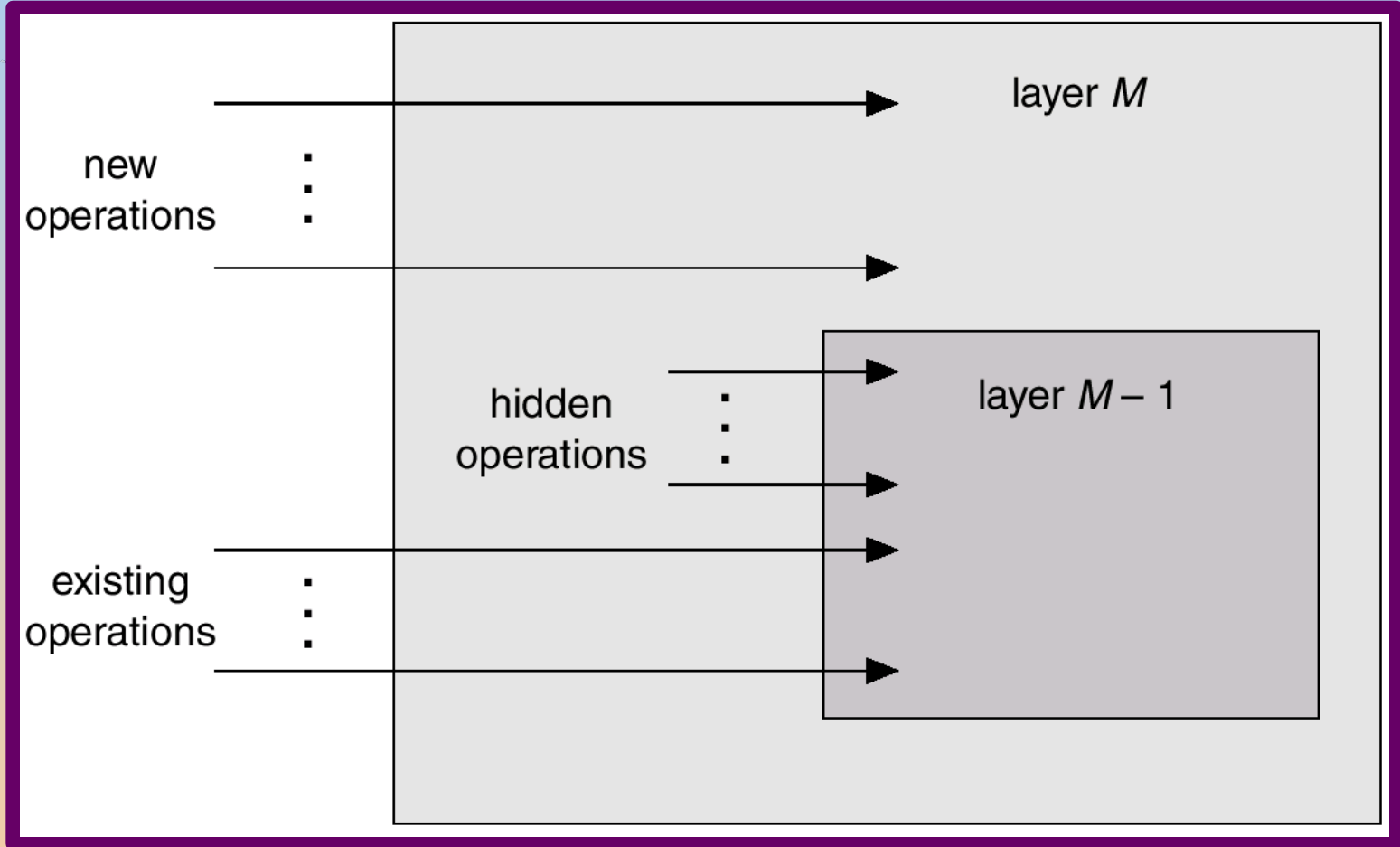
Estrutura do Sistema UNIX



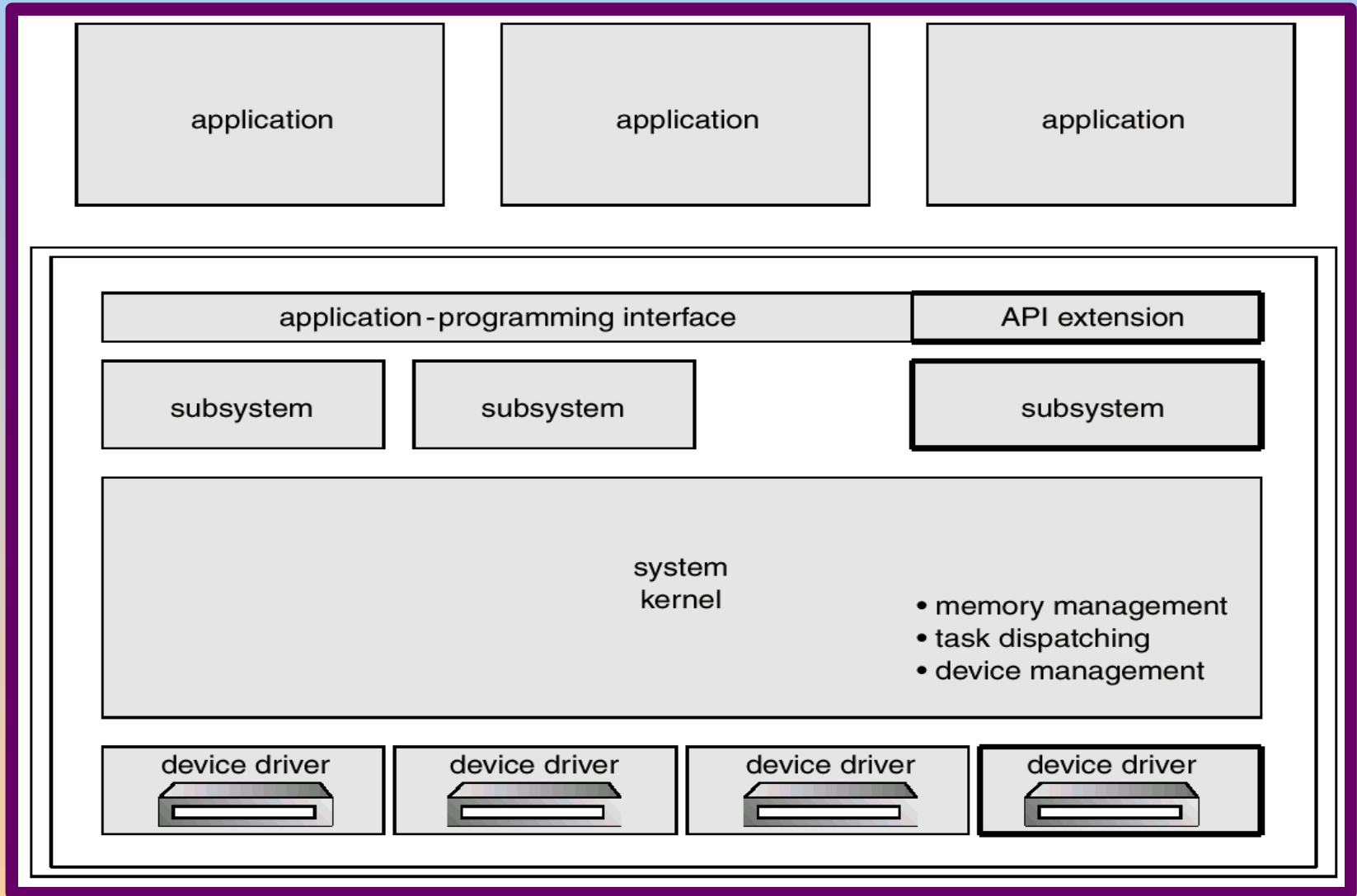
Arquitetura em Camadas

- O sistema operacional é dividido em camadas(níveis), cada uma construída sobre outras camadas. A camada mais inferior (nível 0) é o hardware; a camada superior (nível N) é a interface com o usuário.
- Com modularização, camadas são organizadas de tal modo que usem funções(operações) e serviços de somente camadas de níveis mais baixos.

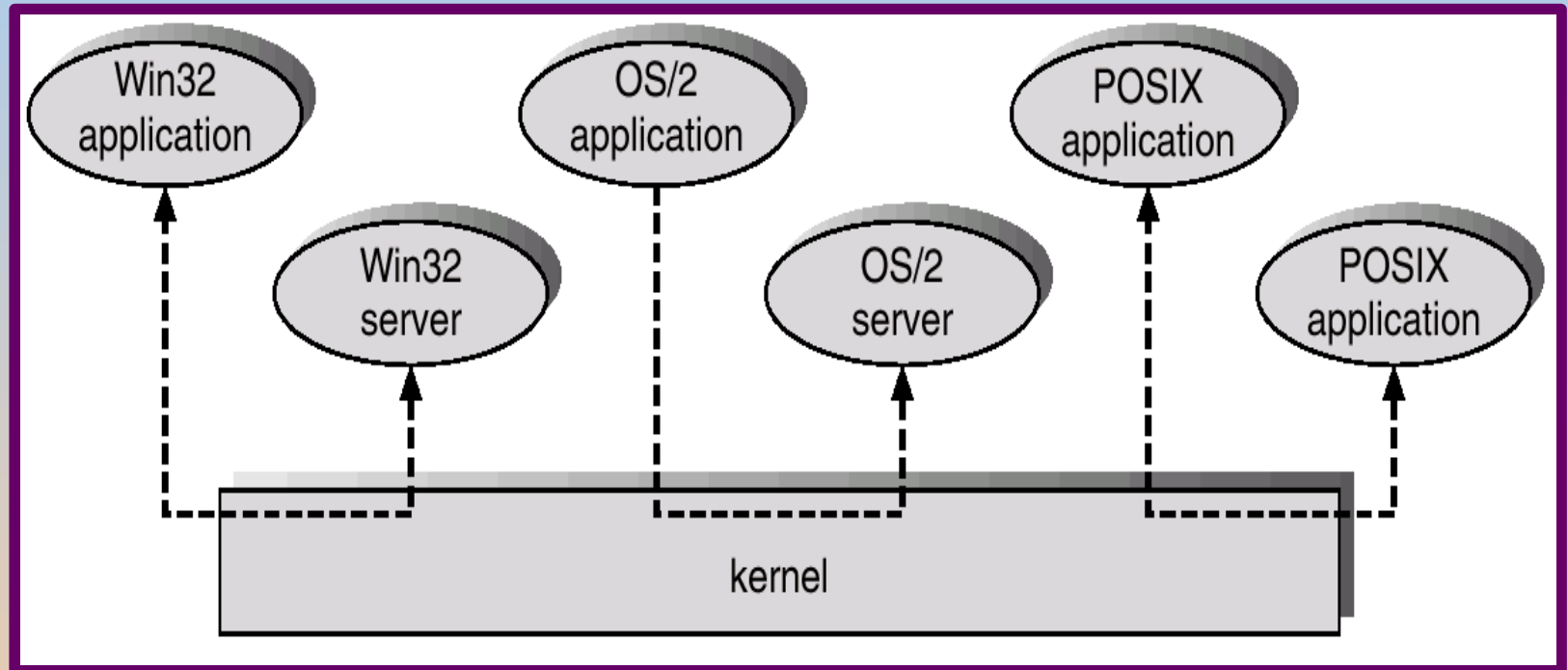
Uma camada do SO



Estrutura de Camadas do OS/2



Estrutura cliente/servidor do Windows NT



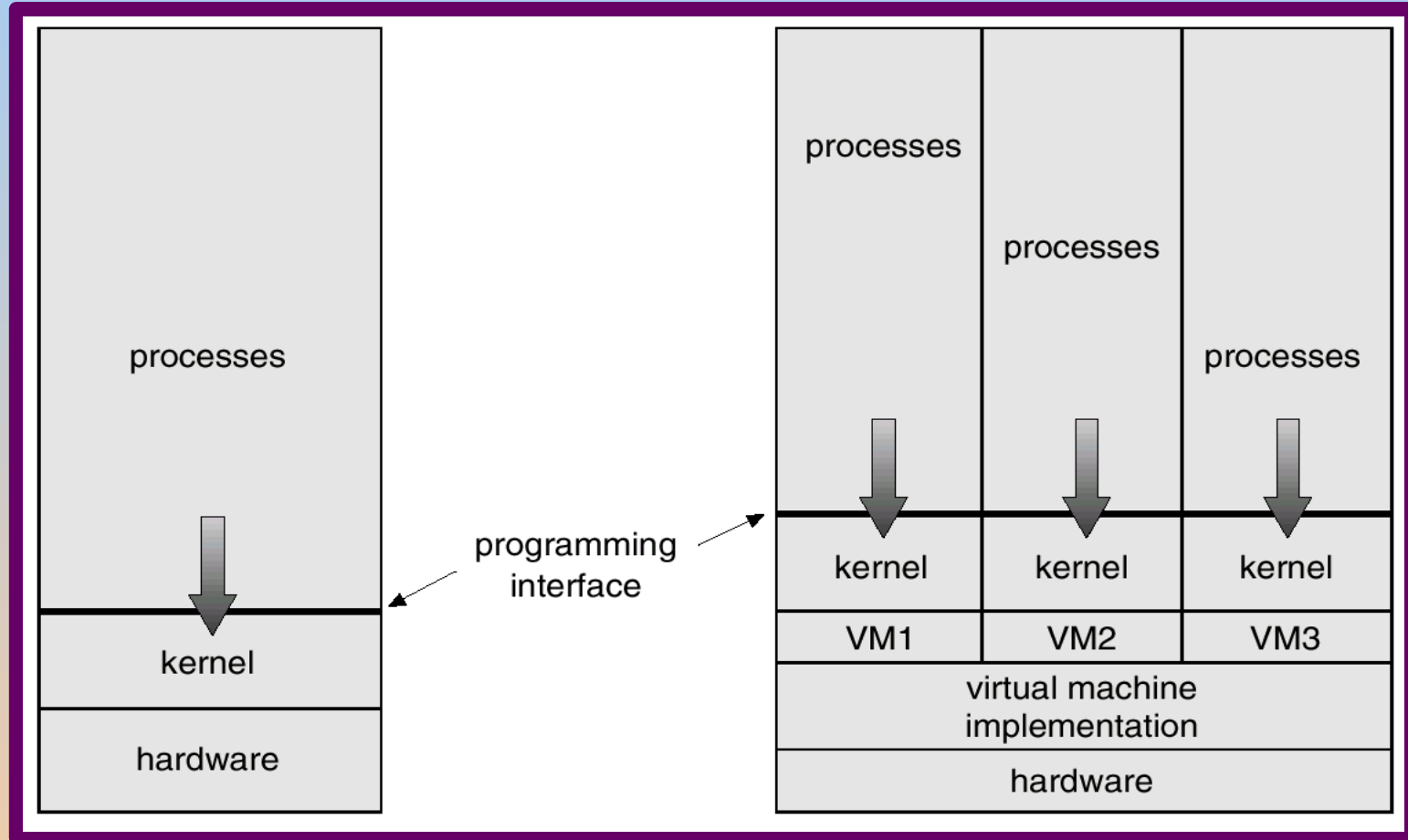
Máquinas virtuais

- Uma máquina virtual trata o hardware e o sistema operacional como se fosse tudo hardware.
- Uma máquina virtual disponibiliza uma interface idêntica ao sistema operacional que lhe serve de base.
- O sistema operacional, via máquinas virtuais, cria a ilusão de múltiplos processos, cada qual executando em seu próprio processador com sua própria memória virtual.

Máquinas Virtuais (Cont.)

- Os recursos físicos de um computador são compartilhados para se criar máquinas virtuais :
 - ◆ Escalonamento da CPU pode criar a aparência que usuários têm o seu próprio processador.
 - ◆ Spooling and um sistema de arquivos podem disponibilizar um leitor de cartões virtual e impressoras de linha virtuais.
 - ◆ Um terminal de usuário de tempo compartilhado normal como um console para o operador da máquina virtual.

Modelo de Sistema



Sem máquina virtual

Com máquina virtual

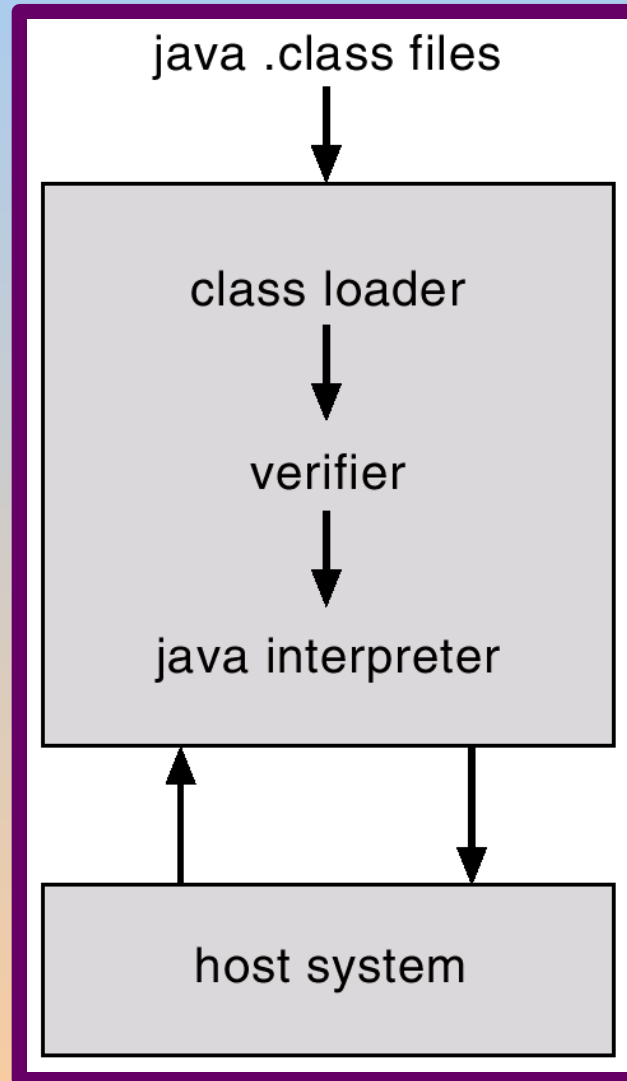
Vantagens/desvantagens de máquinas virtuais

- O conceito de máquina virtual disponibiliza proteção completa dos recursos do sistema, uma vez que cada máquina virtual é isolada de todas as outras máquinas virtuais. Este isolamento, todavia, não permite compartilhamento direto de recursos.
- Máquinas virtuais representam um ambiente perfeito para pesquisa e desenvolvimento de sistemas operacionais. O desenvolvimento de sistemas é feito na máquina virtual, ao invés da máquina física.
- O conceito de máquina virtual é difícil de se implementar, devido ao esforço requerido para disponibilizar uma duplicata exata da máquina que serve como base.

Java Virtual Machine(JVM)

- Programas Java compilados são conjuntos de bytecodes independentes da plataforma, executados pela máquina virtual Java (JVM).
- JVM consiste de:
 - carregador de classes (class loader)
 - verificador de classes (class verifier)
 - interpretador em tempo de execução(runtime interpreter)
- Compiladores Just-In-Time (JIT) compilers possibilitam aumento de performance.

Java Virtual Machine



Objetivos num projeto de SO

- **Objetivos do lado do usuários** – SO deveria ser conveniente de usar, fácil de aprender, confiável, seguro e rápido.
- **Objetivos do lado do sistema** – SO deveria ser fácil de projetar, implementar e manter, bem como ser flexível, confiável, livre de bugs e eficiente.

Mecanismos e políticas

- Mecanismo determinam como fazer alguma coisa; políticas determinam o que será feito.
- A separação entre política e mecanismo é muito importante, pois permite flexibilidade máxima, uma vez que decisões de política possam ser feitas mais tarde.

Implementação de SO

- No início, SOs eram escritos em Assembly; Atualmente, são escritos em linguagens de alto nível (C,C++,Java) .
- Código escrito em uma linguagem de alto nível:
 - ◆ Pode ser escrito mais rápido.
 - ◆ É mais compacto.
 - ◆ É mais fácil de se entender e depurar.
- Um sistema operacional é mais fácil de ser portar ser for escrito em uma linguagem de alto nível.