Sistemas Operacionais

(Estruturas de Sistemas Operacionais))

Prof. Sérgio Murilo Maciel Fernandes

Objetivos

- Eficácia para o usuário: criar um sistema de computação utilizável (exemplo: interfaces gráficas – GUIs);
- Operação eficiente do sistema de computação: compartilhamento de recursos;

Tópicos

- Parte 1
  - Componentes do Sistema
  - Serviços de Sistemas Operacionais
  - Chamadas ao Sistema
- Parte 2
  - Programas do Sistema
  - Estrutura do Sistema
  - Máquinas Virtuais

Componentes do S.O.

- Gerência de Processos
- Gerência de Memória Principal
- Gerência de Arquivos
- Gerência do Sistema de E/S
- Gerência de Armazenamento Secundário
- Redes
- Sistema de Proteção
- Sistema Interpretador de Comandos

Gerência de Processos

- Um processo é um programa em execução.
- Um processo possui necessidade de recursos:
  - tempo de CPU;
  - memória;
  - arquivos;
  - dispositivos de E/S para finalizar sua tarefa.

Gerência de Processos

- O sistema operacional é responsável pelas seguintes tarefas relativas ao gerenciamento de processos:
  - criação e eliminação de processos
  - suspensão e reativação de processos
  - fornecer mecanismos para:
    - sincronização de processos
    - comunicação entre processos

Memória Principal

- A memória é um grande conjunto de palavras e bytes, cada um com endereço próprio
- A memória é um repositório de rápido acesso compartilhada pela CPU e dispositivos de E/S
- A memória principal é um dispositivo de armazenamento volátil - perde o conteúdo em caso de falhas do sistema

Gerência de Memória Principal

- O SO é responsável pelas seguintes atividades relacionadas ao gerenciamento de memória:
  - manter registro das partes da memória que estão sendo usadas no momento e por quem.
  - decidir qual processo deve ser carregado na memória quando houver espaço disponível.
  - alocar e desalocar espaço de memória, conforme necessário.

Arquivos

- Um arquivo é uma coleção de informações relacionadas definidas por seu criador
- Normalmente, arquivos representam programas (fontes ou objetos) e dados.

Gerenciamento de Arquivos

- O SO é responsável pelas seguintes atividades relacionadas à gerência de arquivos:
  - criação e remoção de arquivos
  - criação e remoção de diretórios
  - suporte a primitivas para manipular arquivos e diretórios
  - mapeamento de arquivos em memória secundária
  - backup de arquivos em meios de armazenamento estáveis (não-voláteis)

Gerência do sistema de I/O

- Um dos objetivos de um SO é ocultar as peculiaridades de dispositivos de hardware específicos do usuário

- O sistema de E/S consiste de:
  - *buffering* e *spooling*
  - interface de *device-driver*
  - drivers para dispositivos de hardware específicos

Gerência de memória secundária

- Como a memória principal é volátil e muito pequena para acomodar todos os dados e programas, o computador deve fornecer armazenamento secundário para dar suporte à memória principal.

- O SO é responsável pelas seguintes atividades relacionadas à gerência de memória secundária:
  - gerenciamento do espaço livre
  - alocação de espaço (armazenamento)

Redes (Sistemas Distribuídos)

- Um sistema distribuído é uma coleção de processadores que não compartilham memória ou dispositivos periféricos ou clock e são vistos como compondo um único sistema.
- Um SD reúne sistemas fisicamente separados e possivelmente heterogêneos em um único sistema

Redes (Sistemas Distribuídos)

- Processadores são conectados através de uma rede de comunicação
  - pode ser configurada de formas variadas
  - projeto deve considerar:
    - as estratégias de conexão e roteamento de mensagens
    - os problemas de disputa e segurança

Redes (Sistemas Distribuídos)

- A comunicação se dá através do uso de *protocolos*
- Os protocolos podem ter grande efeito na utilidade e popularidade do sistema

Sistemas Distribuídos – World Wide Web

- criou um novo método de acesso e compartilhamento de informações
- melhorou o protocolo de transferência de arquivos (FTP – File Transfer Protocol) e o protocolo de sistemas de arquivos de redes (NFS - Network File System), removendo a necessidade do logon antes do acesso remoto aos arquivos
- definiu um novo protocolo (http) para uso na comunicação entre um servidor e um navegador Web
- um navegador Web só precisa enviar um pedido de informações ao servidor Web de uma máquina remota e as informações (texto, gráficos, links, etc.) são devolvidas

Sistemas de Proteção

- *Proteção* refere-se ao mecanismo para controlar o acesso de programas, processos ou usuários a recursos do sistema e dos usuários.
- O mecanismo de proteção deve:
  - distinguir entre uso autorizado e não autorizado
  - especificar os controles a serem impostos
  - fornecer meios para reforçar a segurança sobre os recursos

Sistema Interpretador de Comandos

- Muitos comandos são passados para o sistema operacional através de instruções de controle:
  - criação e gerenciamento de processos
  - gerenciamento de E/S
  - gerenciamento de memória secundária
  - gerenciamento de memória principal
  - acesso ao sistema de arquivos
  - proteção
  - gerenciamento de redes

Sistema Interpretador de Comandos

- O programa que lê e interpreta instruções de controle são chamados:
  - interpretador de linha de comando
  - shell (no UNIX)
- O interpretador de comando ou shell pode ter grande influência na facilidade de uso e popularidade do sistema
  - MS-DOS usa uma interface através de comandos seguidos de <enter>
  - Windows utiliza sistema de janelas

Serviços dos S.Os.

- Execução de programa: *o sistema deve ser capaz de carregar um programa na memória e executá-lo.*
- Operações de E/S: *como o usuário não pode controlar os dispositivos de E/S diretamente, o SO deve fornecer meios para operar tais dispositivos*
- Manipulação do sistema de arquivos: *torna possível ler, escrever, criar e remover arquivos.*

Serviços dos S.Os.

- Comunicações: *troca de informações entre processos executando em um mesmo processador ou em processadores distintos, podendo ser implementada via memória compartilhada ou passagem de mensagens*

- Detecção de erros: *maior segurança pela detecção de erros no hardware da CPU ou memória, dispositivos de E/S, ou nos programas do usuário*

Funções adicionais dos S.Os.

Elas existem não para auxiliar o usuário, mas para garantir uma operação eficiente do sistema

- Alocação de recursos: *alocação de recursos (CPU, memória principal, dispositivo de E/S) entre múltiplos processos executando simultaneamente.*
- Contabilidade: *manter um registro sobre quanto e quais tipos de recursos estão sendo usados por cada usuário, com a finalidade de ajustar o sistema para oferecer um melhor serviço*
- Proteção: *garantir que todos os acessos aos recursos do sistema sejam controlados*

Chamadas do Sistema

- Fornecem a interface entre um processo e o SO
  - Chamadas geralmente estão disponíveis como instruções assembly, listadas em manuais usados por programadores assembly
  - certos sistemas permitem que linguagens de alto nível façam chamadas ao sistema diretamente (C, C++ e Perl)
  - As chamadas ao sistema para as plataformas Microsoft Windows fazem parte da API Win32, disponível a todos os compiladores escritos para o Microsoft Windows
  - Java não permite que as chamadas ao sistema sejam feitas diretamente, pois ela é específica a um SO e resulta em código específico daquela plataforma
  - Java pode chamar um método escrito em outra linguagem capaz de fazer a chamada ao sistema

Chamadas ao Sistema: métodos

- Passar parâmetros pelos registradores
- se houver mais parâmetros que registradores, armazena-se os parâmetros em uma tabela na memória e seu endereço é passado como parâmetro em um registrador
- outra possibilidade é ter os parâmetros inseridos em uma pilha pelo programa e lidos pelo sistema operacional

Tipos de Chamadas do Sistema

- Controle de Processo
  - end (processo pára sua execução normalmente)
  - abort (processo termina de forma anormal)
  - load, execute
  - create process, terminate process
  - get process attributes, set process attributes
  - wait for time (aguardar que processo conclua execução)
  - wait event, signal event
  - allocate and free memory

MS-DOS

(Sistema monotarefa)

- possui um interpretador de comandos carregado no início
- carrega o programa na memória por cima de si mesmo para fornecer mais memória
- define o ponteiro de instruções para a primeira instrução do programa
- programa executa até que ocorra um erro ou o programa efetuar a chamada de sistema para terminar
- erro é armazenado
- parte do interpretador não sobreposta reassume
- recarrega o resto do interpretador e disponibiliza o código de erro ao usuário

UNIX

(Sistema multitarefa)

- o interpretador de comandos é carregado e continua sua execução enquanto outros programas são executados
- inicia um novo processo através do comando *fork*
- o programa é carregado na memória via a chamada de sistema *exec*
- o programa é executado
- shell aguarda que o processo seja finalizado ou executa-o em *background*

UNIX

- no último caso, o shell solicita, imediatamente outro comando e o processo não pode receber entradas do teclado, pois o shell está usando este recurso
- operações de E/S feitas através de arquivos ou usando o mouse em uma interface de janelas
- usuário fica livre para solicitar que o shell execute outros programas, monitore o andamento dos processos, etc.
- Quando o processo terminar, ele executa a chamada *exit*

Tipos de Chamadas de Sistema

- Gerenciamento de arquivos
  - create file, delete file
  - open, close
  - read, write, reposition
  - get file attributes, set file attributes

Tipos de Chamadas de Sistema

- Gerenciamento de dispositivos
  - request device, release device
  - read, write, reposition
  - get device attributes, set device attributes

Tipos de Chamadas de Sistema

- Manutenção de informações
  - get time or date, set time or date;
  - get system data, set system data;
  - get process, file or device attributes;
  - set process, file or device attributes.

Tipos de Chamadas de Sistema

- Comunicação
  - create, delete communication connection
  - send, receive messages
  - transfer status information
  - attach or detach remote device

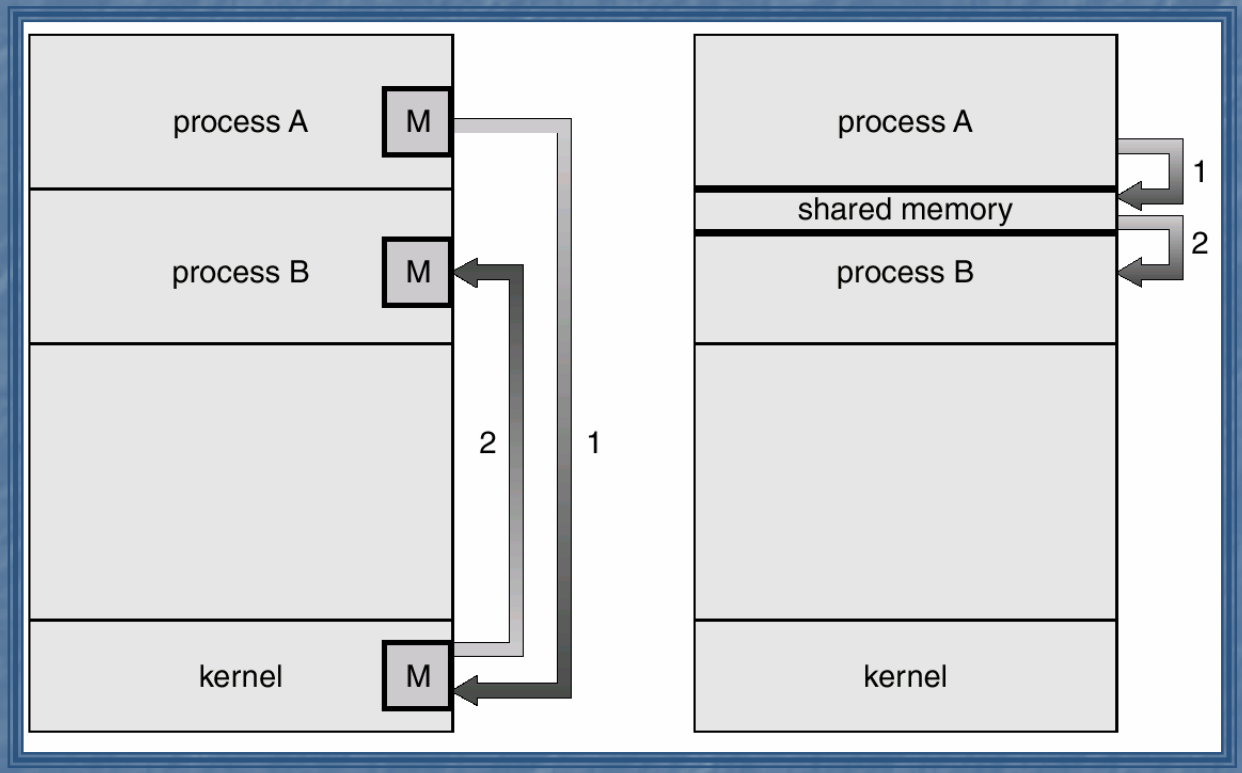
Modelo de Comunicação

- troca de mensagens
  - identificação do outro comunicador deve ser conhecida
  - útil quando uma pequena quantidade de dados é trocada
  - mais fácil de implementar

Modelo de Comunicação

- memória compartilhada
  - requer que processos concordem em remover a restrição imposta pelo SO que impede um processo de acessar a área de memória do outro
  - alto desempenho
  - problemas relacionados à proteção e sincronização

Modelo de Comunicação



Passagem de mensagens

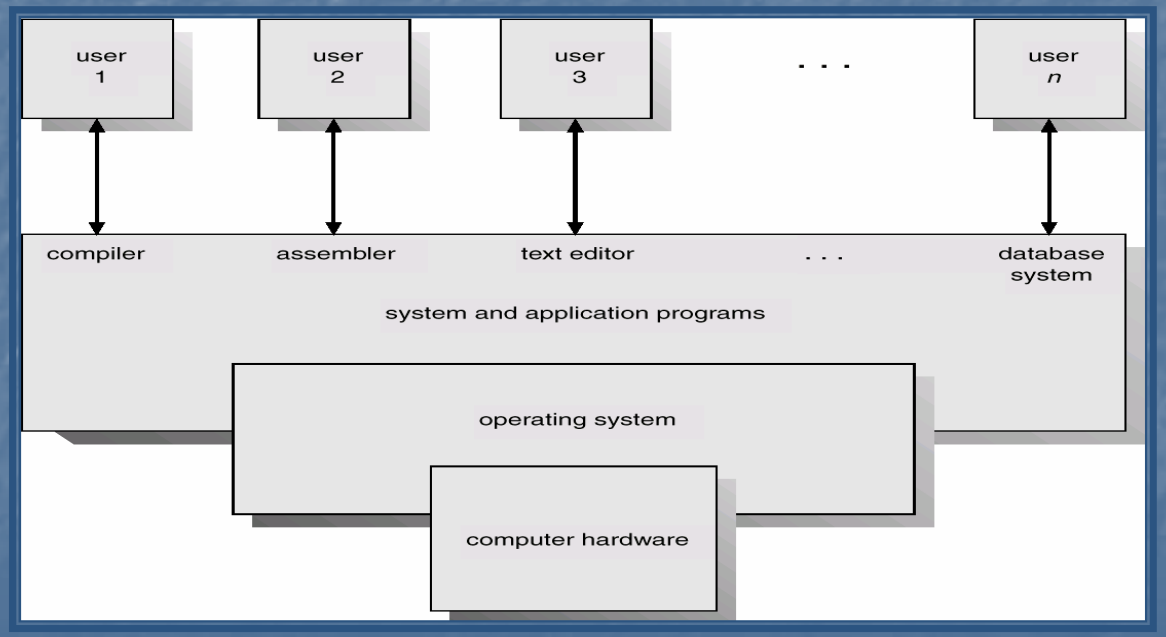
Memória compartilhada

Exercícios

- Cite as principais atividades de um SO com relação à:gerência de processos (5 atividades)
  - gerência de memória (3 atividades)
  - gerência de armazenamento secundário (2 atividades)
  - gerência de arquivos (5 atividades)
  - gerência de E/S (3 atividades)
- Qual o objetivo de um interpretador de comandos?
- Liste cinco serviços fornecidos por um SO. Explique como cada um fornece conveniência aos usuários. Explique em que casos seria impossível para os programas de nível de usuário fornecerem esses serviços.
- Qual o propósito das chamadas ao sistema?
- Cite três tipos de chamadas de sistema para cada tipo de atividade exercida pelo SO listada a seguir
  - Controle de processo
  - Gerência de arquivos
  - Gerência de dispositivos
  - Manutenção de informações
  - Comunicação

Programas de Sistema

- Programas de sistema fornecem um ambiente adequado para o desenvolvimento de programas e sua execução
- Alguns programas de sistema são simples interfaces para chamadas ao sistema



Programas de Sistema: Tipos

- Gerência de arquivos: *criam, excluem, copiam, renomeiam, imprimem, listam e manipulam arquivos e diretórios*
- Modificação de arquivos: *editores de texto*
- Informação de status: *pedem ao sistema informações relativas à data, hora, quantidade de memória ou espaço em disco disponível, número de usuários, etc. Estas informações são então formatadas e impressas no terminal ou dispositivo de E/S*

Programas de Sistema: Tipos

- Suporte à linguagem de programação: *compiladores, montadores e interpretadores para linguagens de programação são geralmente fornecidos ao usuário com o SO.*
- Carregamento e execução de programas: *depois que um programa é montado e compilado, deve ser carregado na memória para ser executado. O sistema pode fornecer utilitários de carga.*

Programas de Sistema: Tipos

- Comunicações: *fornecem mecanismos para criar conexões virtuais entre processos, usuários e outros sistemas de computação; permitem usuários enviar mensagens às telas uns dos outros, navegar pelas páginas da Web, enviar mensagens de correio eletrônico, efetuar logon remotamente ou transferir arquivos de uma máquina para outra*
- Programas aplicativos: *a maioria dos SO possuem programas úteis para a resolução de problemas comuns (navegadores Web, editores de texto, planilhas, banco de dados, geradores de compiladores, etc.)*

Sistema interpretador de Comandos

Formas de implementação

- interpretador contém código para executar o comando
  - ex: comando para excluir um arquivo faz com que interpretador salte para uma seção de código que configura os parâmetros e faz a chamada ao sistema adequada
  - tamanho do interpretador depende do número de comandos que ele aceita

Sistema interpretador de Comandos

Formas de implementação

- interpretador não entende o comando, ele implementa os comandos através de programas de sistema (UNIX)
  - nome do comando simplesmente identifica o arquivo a ser carregado na memória e executado
  - ex: `rm Arq.txt` - procura o arquivo rm, carrega o arquivo na memória, e o executa
  - programadores podem adicionar novos comandos ao sistema criando novos arquivos com nomes adequados
  - o interpretador não precisa ser alterado para que os novos comandos sejam acrescentados

Sistema interpretador de Comandos: Problemas

- o código para executar o comando é um programa separado do SO
- o SO deverá fornecer mecanismos para passar parâmetros do interpretador para o programa
- o programa pode ainda não estar na memória
- lista de parâmetros pode ser grande
- é mais lento carregar um programa e executá-lo do que simplesmente pular para uma seção do código dentro do programa

Programas de Sistema

... muitas vezes, a visão que o usuário tem sobre o sistema é definida pelos programas de sistema e não pelas chamadas ao sistema propriamente ditas ...

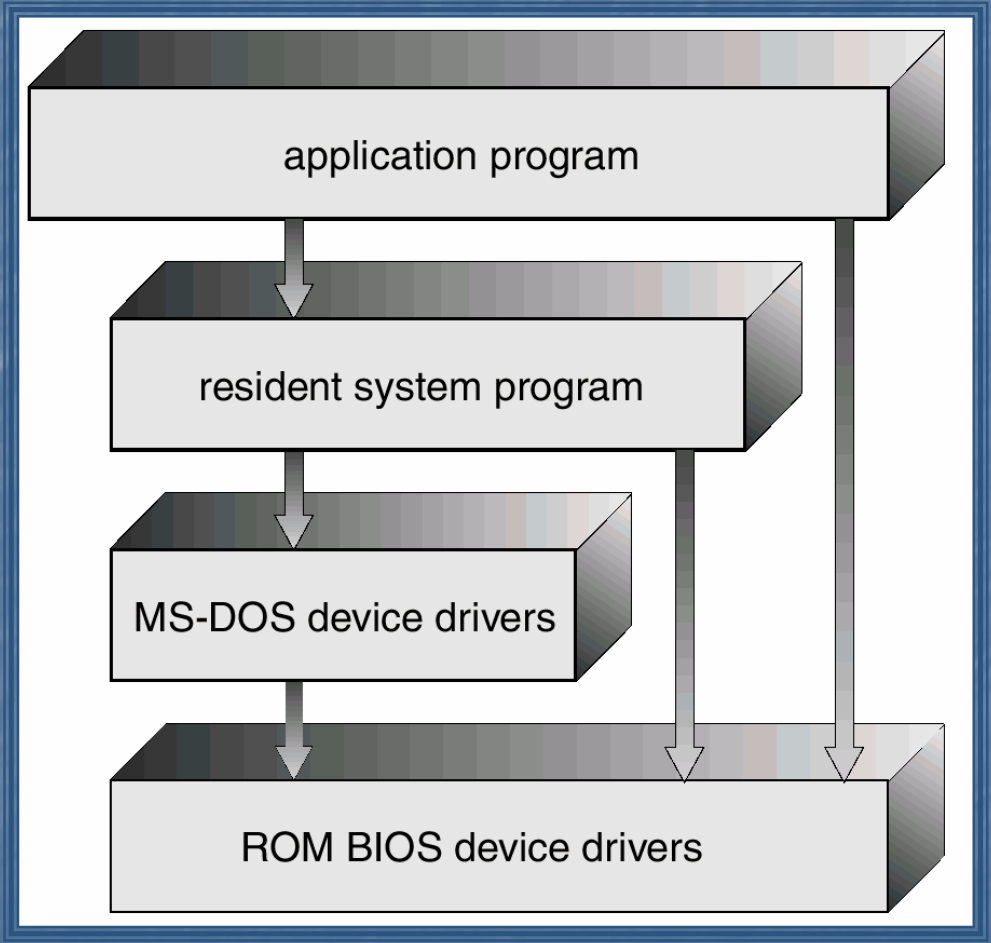
Estrutura do Sistema MS-DOS

MS-DOS – escrito para oferecer o máximo de funcionalidade utilizando pouco recurso

- Não é dividido em módulos
- Embora possua uma estrutura, a interface e o nível de funcionalidades não são bem separados no MS-DOS
- programas aplicativos podem acessar rotinas básicas de I/O para escrever diretamente na tela
- tal liberdade deixa o MS-DOS vulnerável a programas com erros ou maliciosos

O MS-DOS foi escrito para o Intel 8088, que não possui o modo dual, nem proteção de hardware

Estrutura de Níveis do MS-DOS

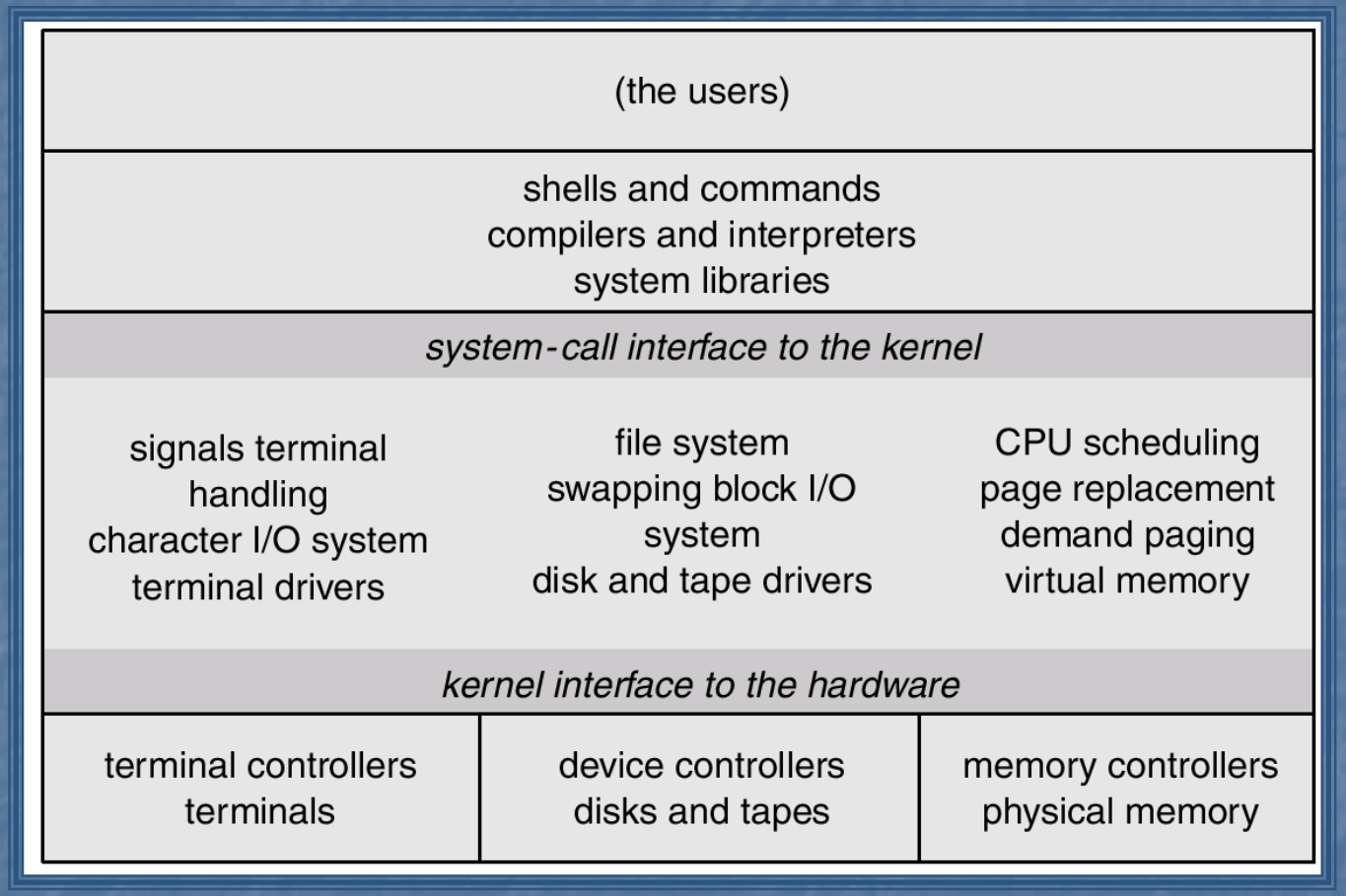


Estrutura do Sistema UNIX

UNIX consiste de duas partes separadas

- Programas de Sistema
- O kernel
  - consiste de tudo abaixo da interface de chamadas ao sistema e acima do hardware
  - possui o sistema de arquivos, escalonador da CPU, gerenciamento de memória, e outras funções do SO

Estrutura do Sistema UNIX

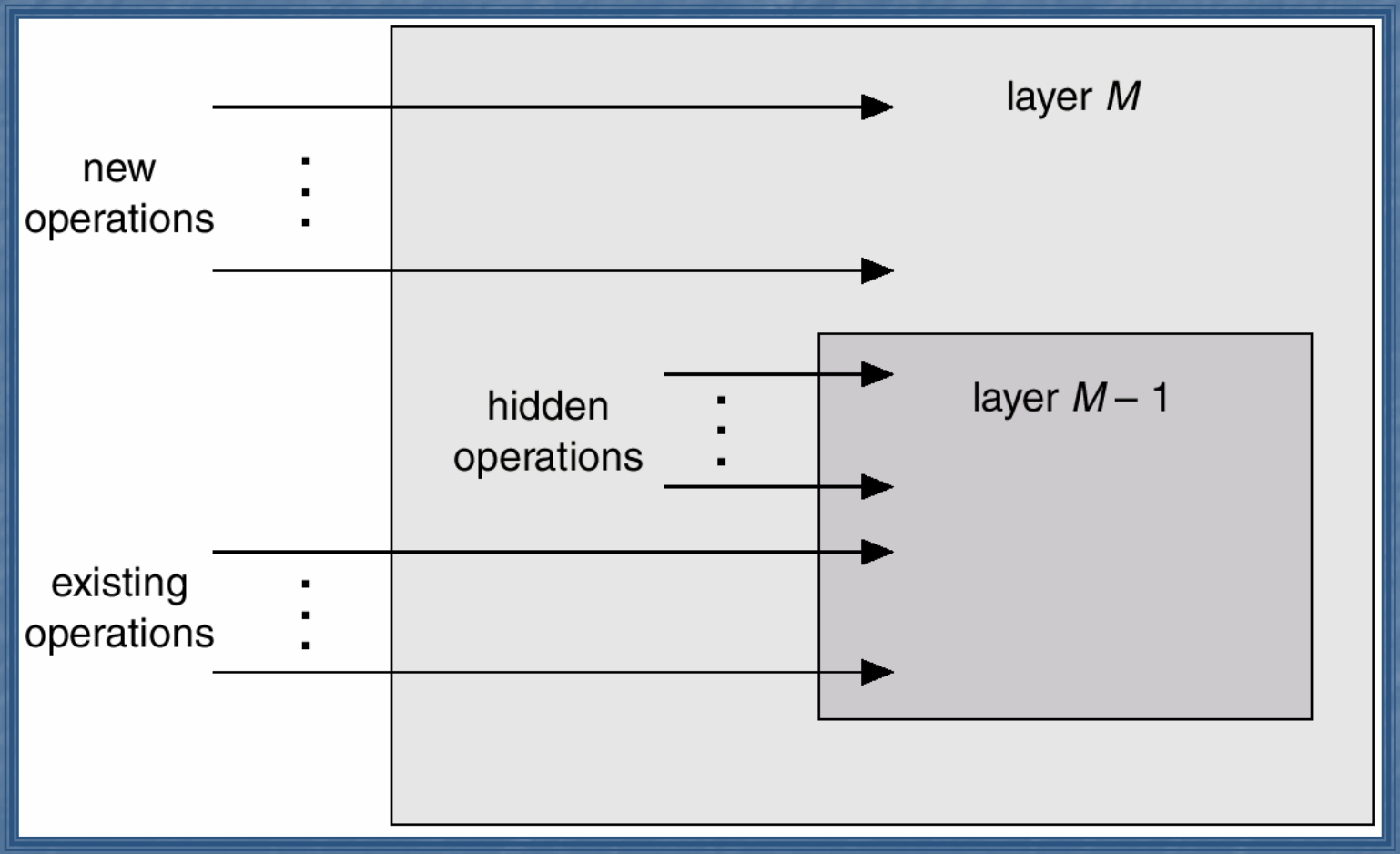


Abordagem em Camadas

- O SO é dividido em camadas (níveis), construídos sobre níveis mais baixos
  - nível 0 - hardware
  - nível mais alto - interface do usuário

- camadas são organizadas para que cada uma utilize as operações e serviços das camadas de nível mais baixo

Camadas de um Sistema Operacional



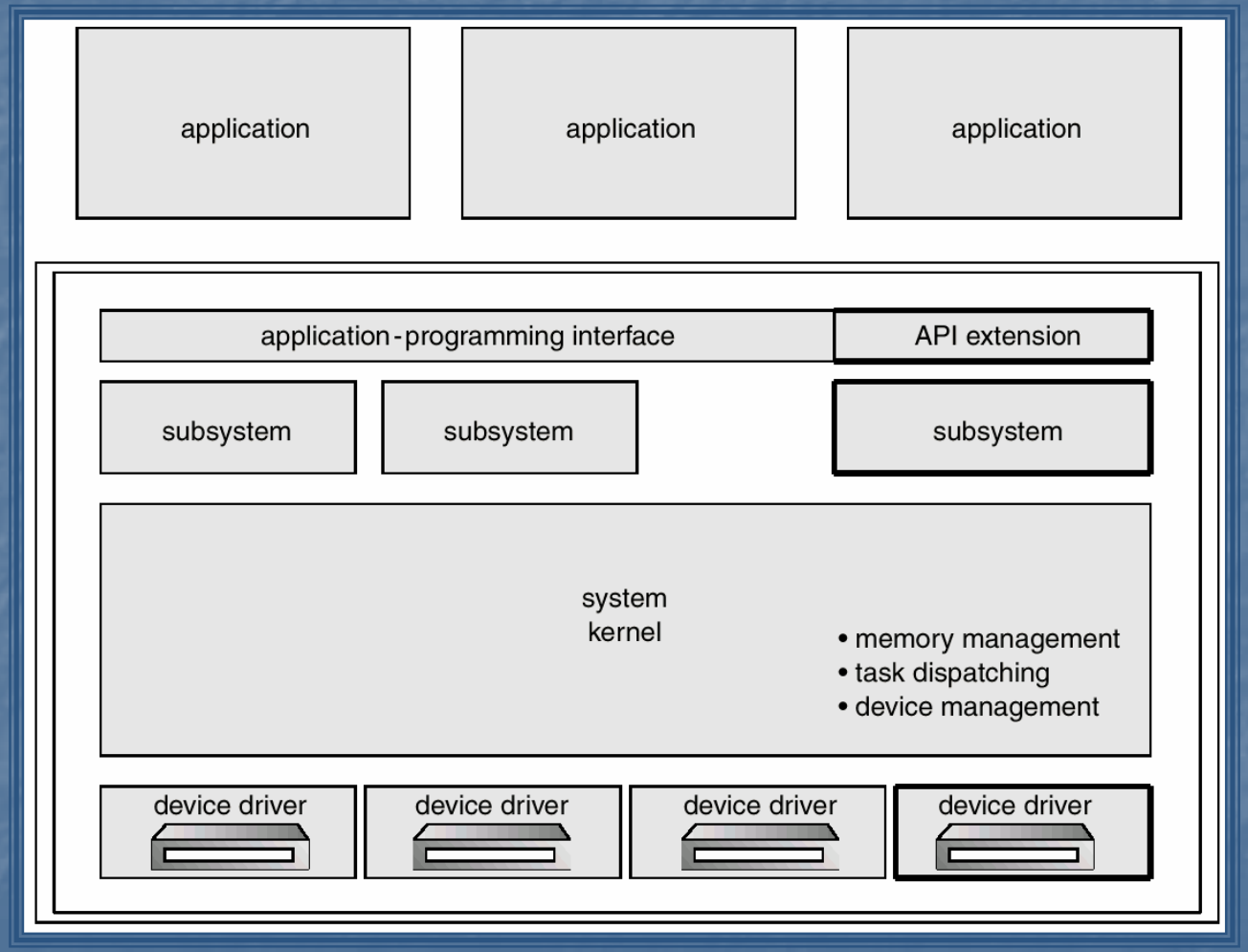
Abordagem em Camadas

- Cada camada é implementada apenas com as operações fornecidas pela camada inferior
- Não é necessário saber como essas operações são implementadas
- se for encontrado um erro durante a depuração de uma camada, o erro deve estar nesta camada, pois as camadas inferiores já foram depuradas

Abordagem em Camadas: Problemas

- Encontrar a definição adequada das várias camadas
  - cada camada só pode usar as camadas em um nível inferior => deve haver um planejamento cuidadoso
  - Ex: Driver do HD deve estar em um nível abaixo das rotinas de gerência de memória, porque a gerência de memória utiliza a memória auxiliar
- Tendem a ser menos eficientes: muitos níveis intermediários !

Estrutura de Níveis do OS/2



Estrutura de Níveis do Windows NT

- Primeira versão
  - fortemente orientada a camadas
  - desempenho inferior ao Windows 95
- Windows NT 4.0
  - moveu camadas do espaço do usuário para o espaço do Kernel, promovendo uma maior integração entre elas
  - melhor desempenho !

Microkernels

- Move o máximo possível do Kernel para o espaço do usuário
- Resultado: Kernel menor
- Pouco consenso sobre o que manter no Kernel
  - Em geral, fornecem gerência de memória e processos, além de recursos de comunicação
- Benefícios
  - fácil de estender
  - fácil mover o SO para novas arquiteturas
  - mais confiável (menos código no modo Kernel)
  - mais segurança

Microkernels

- Uma tendência dos sistemas operacionais é tornar o núcleo menor e mais simples possível e para implementar esta idéia o sistema é dividido em processos
- Desta forma, sempre que uma aplicação deseja algum serviço ela solicita ao processo responsável, assim, a aplicação que solicita um serviço é chamada de cliente e o processo que responde a solicitação é chamado de servidor

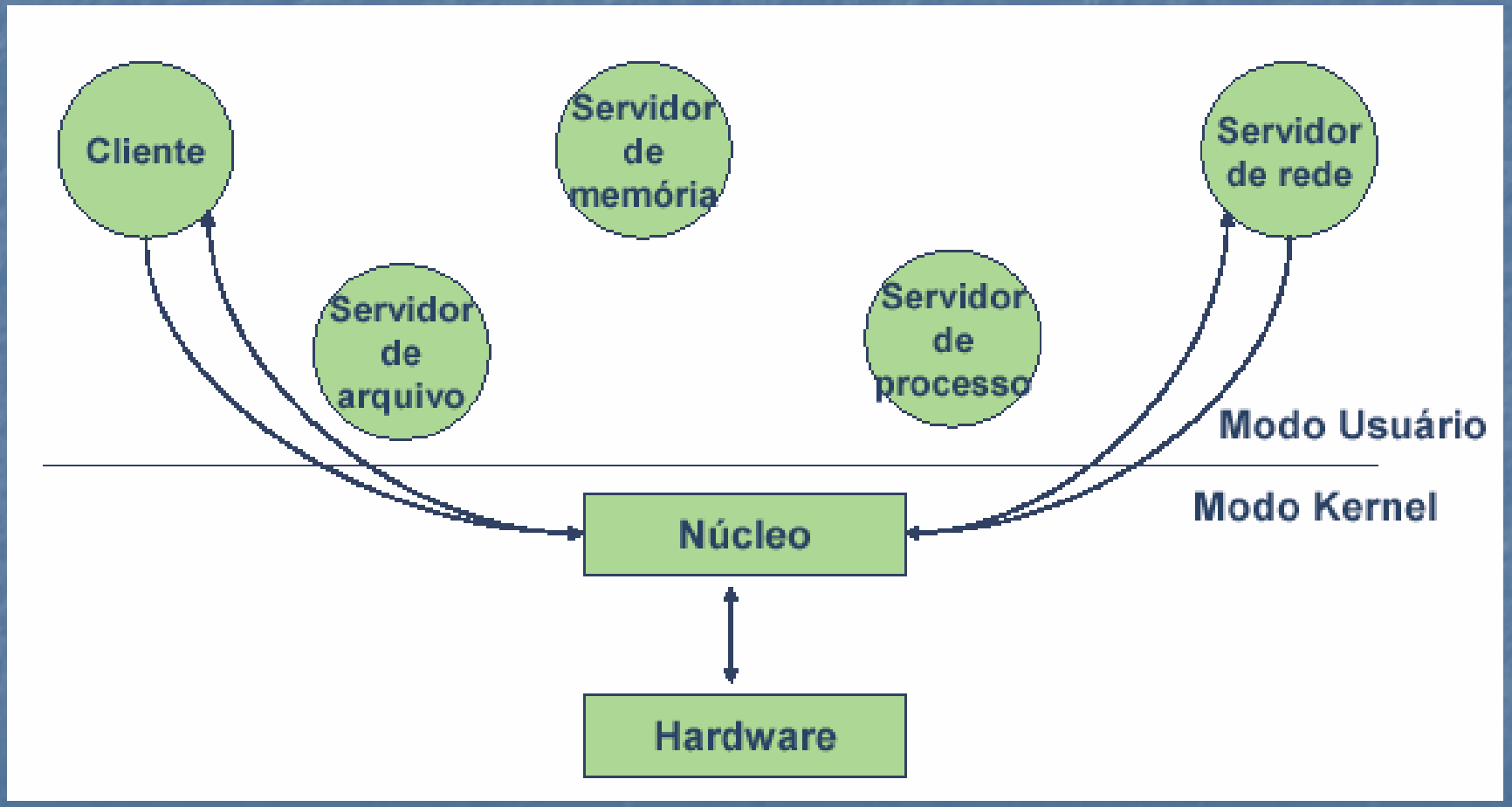
Microkernels

- A utilização deste modelo permite que os servidores executem em modo usuário
- Apenas o núcleo do sistema, responsável pela comunicação entre clientes e servidores, execute o modo kernel
- O sistema operacional passa a ser de mais fácil manutenção
- Não importa se o serviço esta sendo processado em um único processador, com múltiplos processadores (fortemente acoplado) ou em sistema distribuído (fracamente acoplado)

Microkernels

- Em ambiente distribuído permite que um cliente solicite um serviço e a resposta seja processada remotamente
- Sua implementação é difícil e mais usualmente é implantado uma combinação do modelo de camadas com o cliente-servidor
- O núcleo do sistema passa a incorporar o escalonamento e gerência de memória além das funções de *device drivers* (controladores de dispositivos).

Microkernels

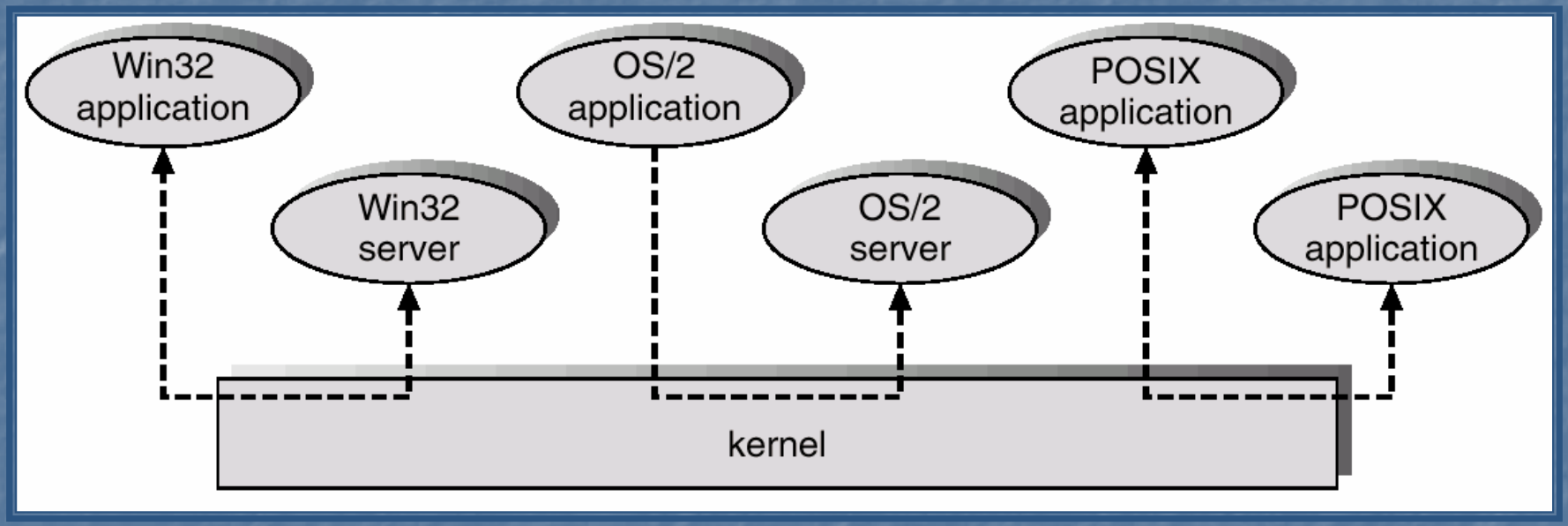


Microkernels

- **Exemplo**
  - **UNIX**
- **Windows NT**
  - **Estrutura híbrida**
  - **Projetado para executar várias aplicações (Win32, OS/2 e POSIX)**
  - **Fornece servidor que executa no espaço do usuário**
  - **Kernel coordena troca de mensagens entre as aplicações cliente e servidores de aplicação**

Windows NT

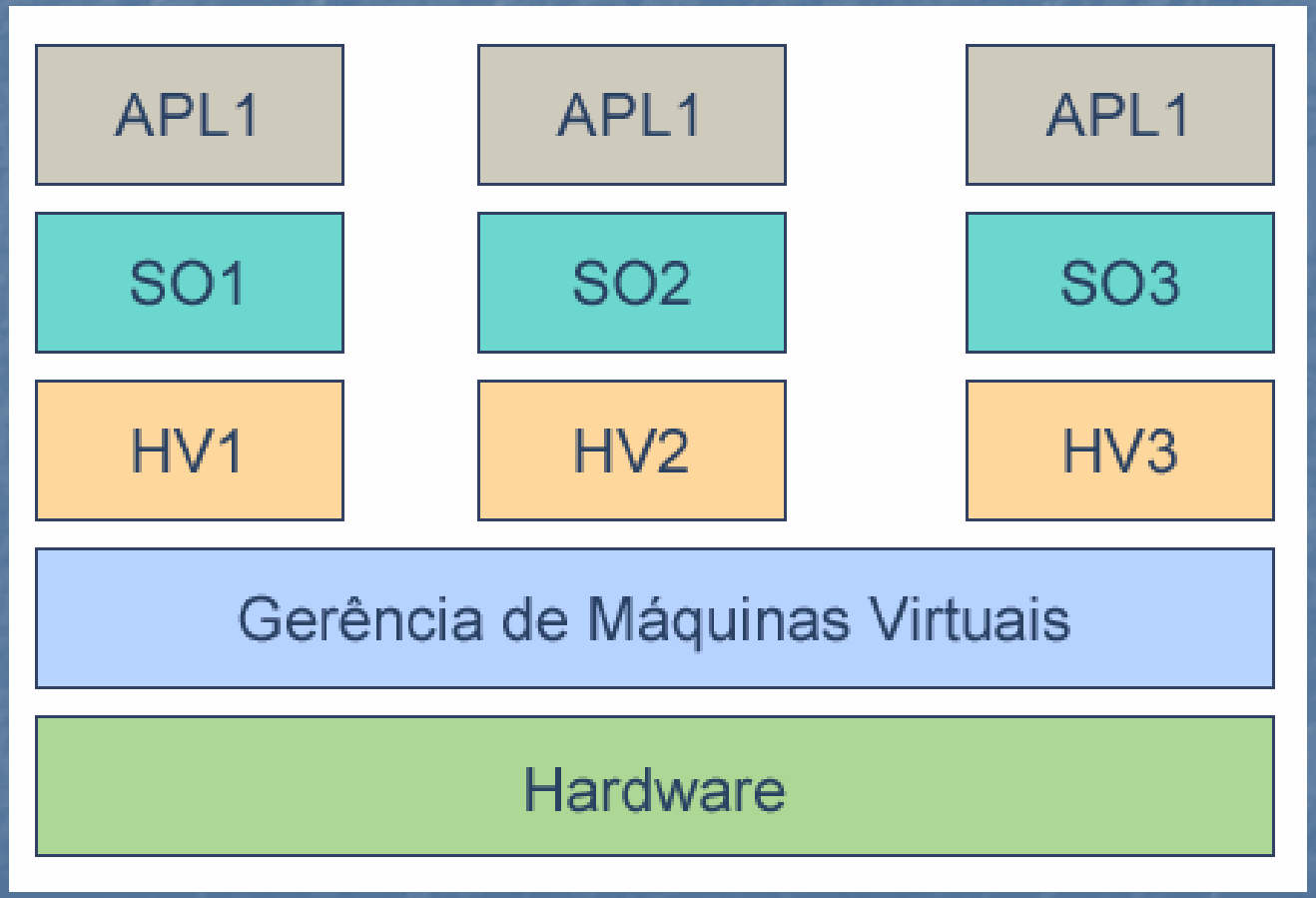
Estrutura Cliente-Servidor



Máquina Virtual

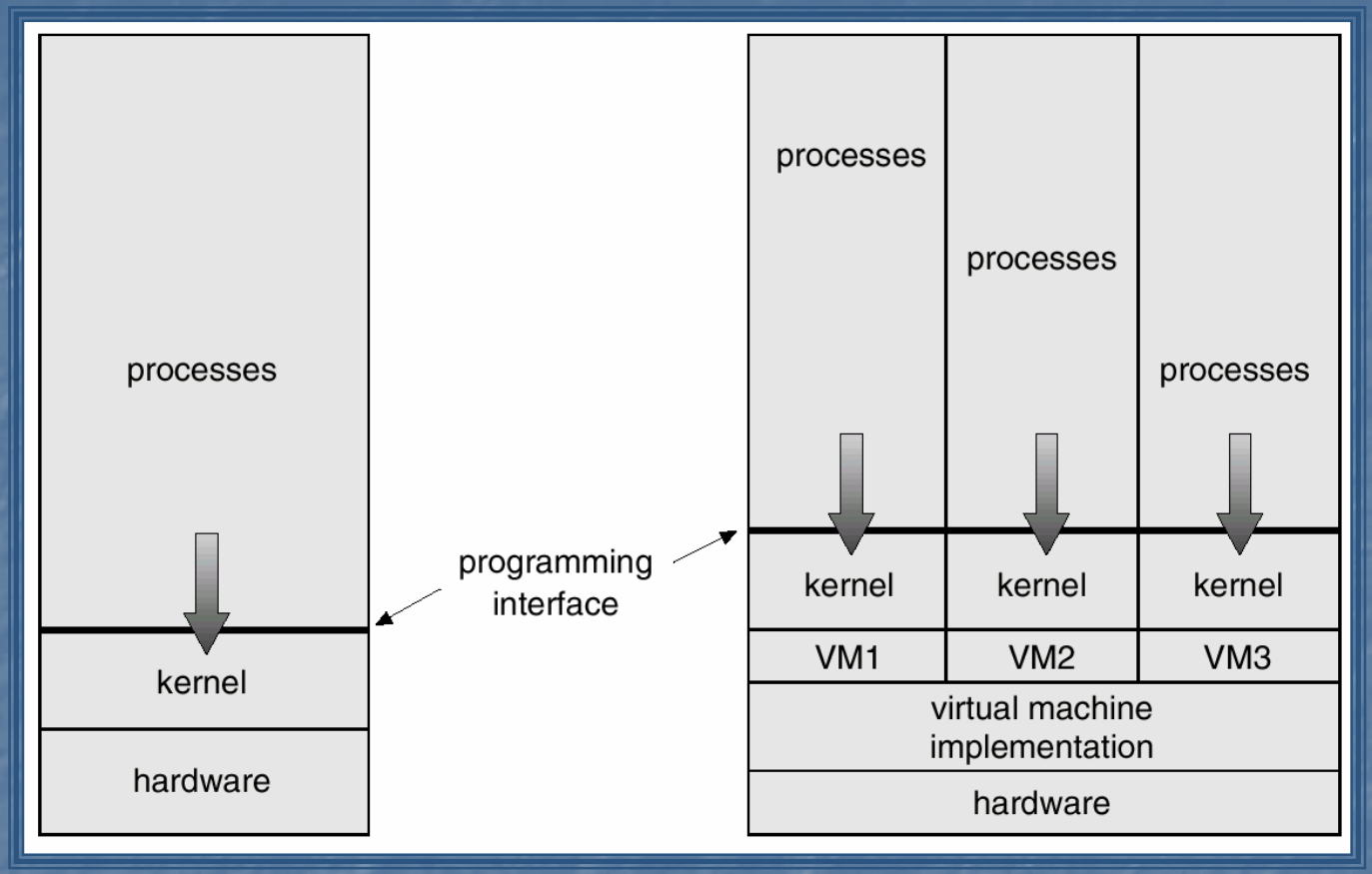
- **Conceitualmente, um sistema de computação é formado por camadas**
  - **O hardware é o nível mais baixo**
  - **Kernel executa no próximo nível**
  - **Programas do sistema estão acima do kernel**
    - **não distinguem entre hardware e Kernel**
- **Máquina Virtual:** *aplicativos acessam a máquina através dos programas de sistema*

Máquina Virtual



Máquina Virtual

- A máquina virtual fornece uma interface idêntica ao hardware
- O SO cria a ilusão de múltiplos processos, cada um executando em seu próprio processador com sua memória (virtual) específica (Ex: VM - IBM)
- Os recursos do computador físico são compartilhados para criar a máquina virtual
  - O escalonamento da CPU pode compartilhar a CPU para criar a ilusão de que os usuários têm seus próprios processadores
  - *Spooling* pode fornecer impressoras virtuais
  - um terminal de usuário compartilhado fornece a função de console do operador da máquina virtual



Máquina Não-Virtual Máquina Virtual

Vantagens e Desvantagens da Máquina Virtual

- O conceito de MV fornece uma completa proteção dos recursos do sistema, pois isola cada MV das demais. Este isolamento, porém, impede o compartilhamento direto de recursos
- Um sistema de MV é perfeito para pesquisa e desenvolvimento de SO. O desenvolvimento do SO é feito sobre a MV e não na máquina física, evitando interrupções no funcionamento normal do sistema
- O conceito de MV é difícil de implementar devido ao esforço requerido para criar o efeito *exato* da máquina física

Exercícios

- Processadores de 64 bits são hoje uma realidade. Vale a pena ter um sistema operacional de 64 bits hoje em dia? Analise a questão do Microsoft Windows e do Linux.
- Qual a principal vantagem da abordagem em camadas ao projeto do sistema ?
- Qual a principal vantagem da abordagem microkernel ao projeto do sistema?
- Qual a principal vantagem de um projetista de SO utilizar uma arquitetura de máquina virtual?
- Qual a principal vantagem da abordagem máquina virtual para o usuário ?