Aula 8: Serviços do Sistema Operacional

Professor(a): Virgínia Fernandes Mota
http://www.dcc.ufmg.br/∼virginiaferm

OCS (TEORIA) - SETOR DE INFORMÁTICA

Tipos de serviço do SO

- O S.O. fornece um ambiente para a execução de programas através de serviços para os programas e para os usuários desses programas
- Apesar da forma como esses serviços serviços são oferecidos variar de sistema para sistema existem algumas classes de serviços que são comuns a todos os sistemas operacionais

Tipos de serviço do SO

- Serviços mais comuns gerenciados pelo S.O.:
  - Execução de programas;
  - Operações de entrada/saída;
  - Manipulação de sistema de arquivos;
  - Detecção de erros;
  - Alocação de recursos;
  - Proteção.

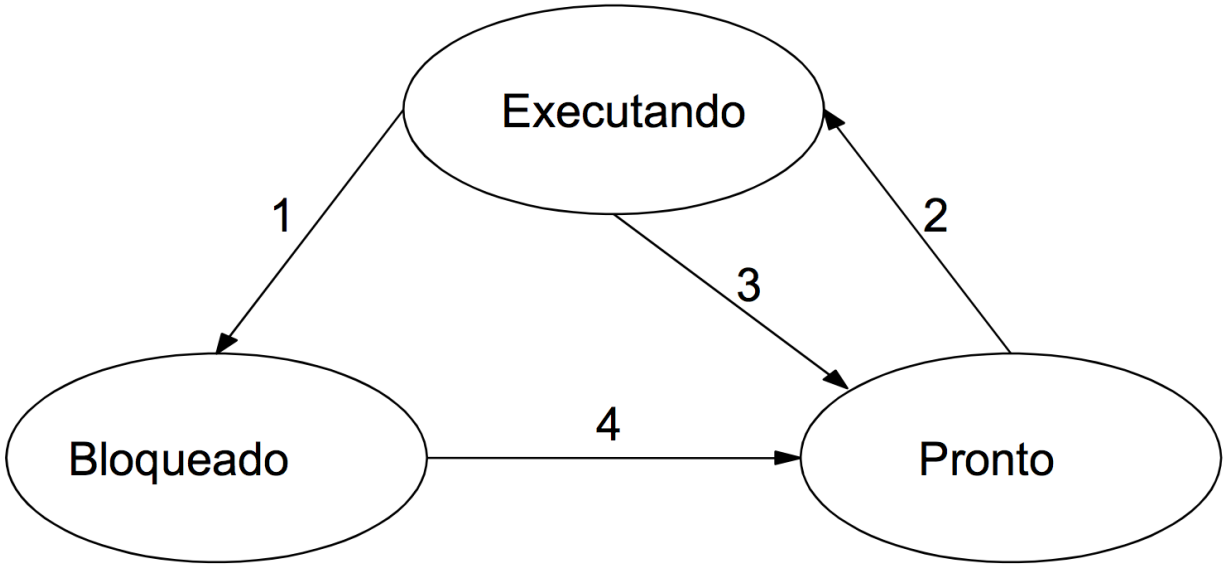
Conceitos Básicos de Sistemas Operacionais

- Principais conceitos:
  - Processo;
  - Memória;
  - Chamadas de Sistema;

Processos

- Processo: chave do SO;
- Caracterizado por programas em execução;
- Cada processo possui:
  - Um espaço de endereço;
  - Uma lista de alocação de memória (mínimo, máximo);
  - Um conjunto de registradores (contador de programa);
- O Sistema Operacional controla todos os processos;

- Estados básicos de um processo



- Ex.: processo bloqueado (suspenso)
- Quando o SO suspende um processo P1 temporariamente para executar um processo P2, o processo P1 deve ser reiniciado exatamente no mesmo estado no qual estava ao ser suspenso. Para tanto, todas as informações a respeito do processo P1 são armazenadas em uma tabela de processos (process table). Essa tabela é um vetor ou uma lista encadeada de estruturas.

Processos

- Um processo pode resultar na execução de outros processos, chamados de processos-filhos. Características para a hierarquia de processos:
  - Comunicação (Interação) e Sincronização;
  - Segurança e proteção;
  - Uma árvore de no máximo três níveis;
- Escalonadores de processos - processo que escolhe qual será o próximo processo a ser executado $\rightarrow$ Diversas técnicas para escalonamento de processos;

- Comunicação e sincronismo entre processos - possíveis soluções:
  - Semáforos;
  - Monitores;
  - Instruções especiais em hardware;
  - Troca de mensagens;

Gerenciamento de Memória

- Gerenciamento elementar (década de 60)
  - Sistema monoprogramado;
  - Sem paginação: Apenas um processo na memória; Acesso a toda a memória;
- Gerenciamento mais avançado (atualidade)
  - Sistema multiprogramado;
  - Mais de um processo na memória;
  - Chaveamento de processos: por entrada/saída ou por limite de tempo (sistema de tempo compartilhado);

Compartilhamento de Memória

- Partições Fixas
  - Cada processo é alocado em uma dada partição da memória (pré-definida);
  - Partições são liberadas quando o processo termina;
- Partições Variáveis
  - Memória é alocada de acordo com o tamanho e número de processos;
  - Otimiza o uso da memória;

Comunicação entre usuário e o SO

- **Chamadas ao Sistema** (system calls) fornecem uma interface entre um programa em execução e o S.O. Estão, geralmente, disponíveis como instruções nas linguagens de baixo nível ou até mesmo em linguagens de alto nível, como C.
- Podem ser classificadas em duas categorias:
  - Controle de processos.
  - Gerenciamento de arquivos e de dispositivos de E/S.

Chamadas ao Sistema

- Modos de Acesso:
  - Modo usuário;
  - Modo kernel ou Supervisor ou Núcleo;
- São determinados por um conjunto de bits localizados no registrador de status do processador: PSW (program status word);
  - Por meio desse registrador, o hardware verifica se a instrução pode ou não ser executada pela aplicação;
- Protege o próprio kernel do Sistema Operacional na RAM contra acessos indevidos;

Chamadas ao Sistema - Modo usuário

- Aplicações não têm acesso direto aos recursos da máquina, ou seja, ao hardware;
- Quando o processador trabalha no modo usuário, a aplicação só pode executar instruções sem privilégios, com um acesso reduzido de instruções;
- Por que? Para garantir a segurança e a integridade do sistema;

Chamadas ao Sistema - Modo kernel

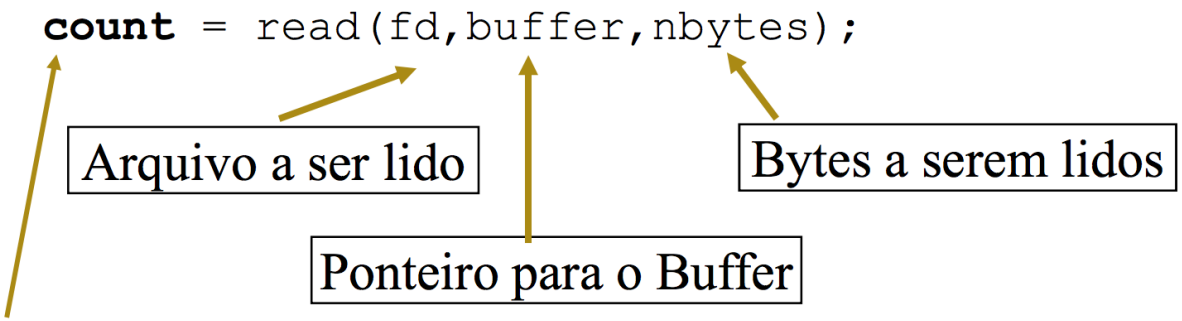
- Aplicações têm acesso direto aos recursos da máquina, ou seja, ao hardware;
- Operações com privilégios;
- Quando o processador trabalha no modo kernel, a aplicação tem acesso ao conjunto total de instruções;
- Apenas o SO tem acesso às instruções privilegiadas;

Alteração entre modos

- Se uma aplicação precisa realizar alguma instrução privilegiada, ela realiza uma chamada ao sistema (system call), que altera do modo usuário para o modo kernel;
- Chamadas de sistemas são a porta de entrada para o modo Kernel;
  - São a interface entre os programas do usuário no modo usuário e Sistema Operacional no modo kernel;
  - As chamadas diferem de SO para SO, no entanto, os conceitos relacionados às chamadas são similares independentemente do SO;

Execução de chamadas ao sistema

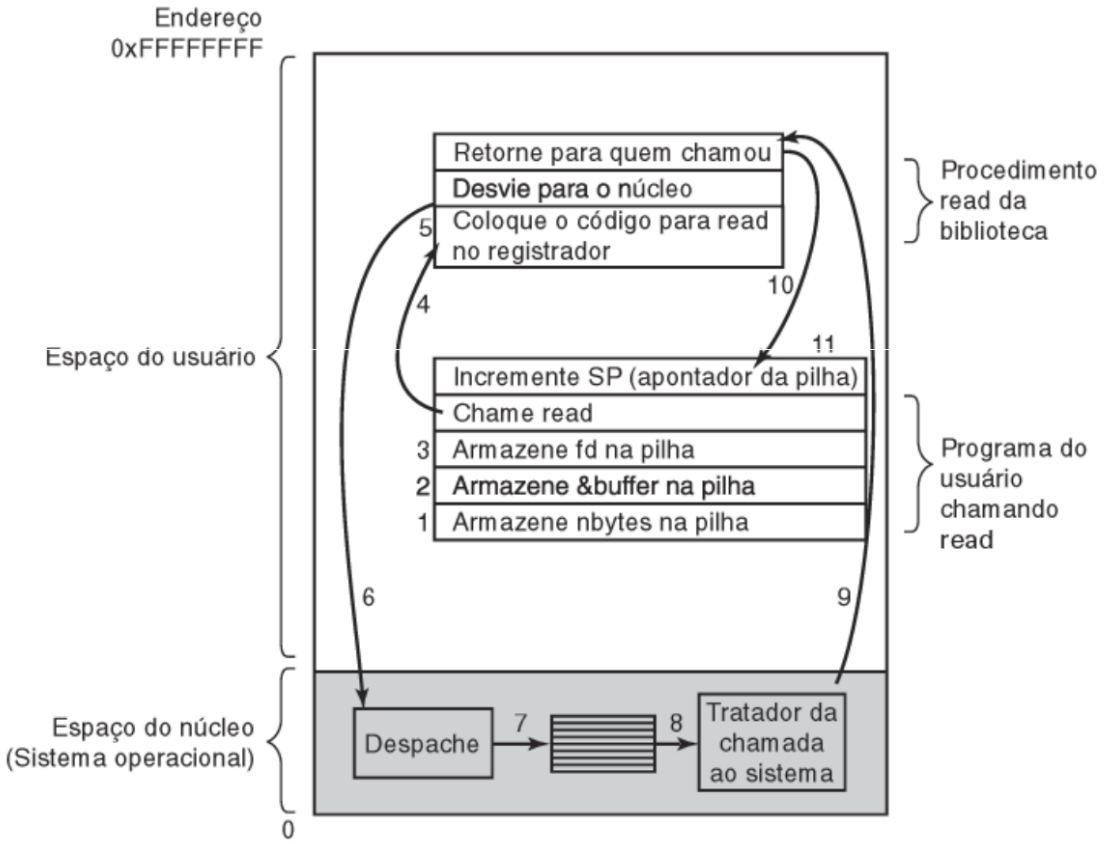
- TRAP: instrução que permite o acesso ao modo kernel;
- Exemplo: Instrução do UNIX



O programa sempre deve checar o retorno da chamada de sistema para saber se algum erro ocorreu!!!

Execução de chamadas ao sistema

Os 11 passos para fazer uma chamada ao sistema para o comando "*read (arq, buffer, nbytes)*"



Após o passo 5, é executado um TRAP, passando do modo usuário para o modo sistema

Exemplos de chamadas ao sistema

Gerenciamento de processos

Chamada	Descrição
pid = fork( )	Crie um processo filho idêntico ao processo pai
pid = waitpid(pid, &statloc, options)	Aguarde um processo filho terminar
s = execve(name, argv, environp)	Substitua o espaço de endereçamento do processo
exit(status)	Termine a execução do processo e retorne o estado

Gerenciamento de arquivos

Chamada	Descrição
fd = open(file, how, ...)	Abra um arquivo para leitura, escrita ou ambas
s = close(fd)	Feche um arquivo aberto
n = read(fd, buffer, nbytes)	Leia dados de um arquivo para um buffer
n = write(fd, buffer, nbytes)	Escreva dados de um buffer para um arquivo
position = lseek(fd, offset, whence)	Mova o ponteiro de posição do arquivo
s = stat(name, &buf)	Obtenha a informação de estado do arquivo

Exemplos de chamadas ao sistema

Gerenciamento do sistema de diretório e arquivo

Chamada	Descrição
s = mkdir(name, mode)	Crie um novo diretório
s = rmdir(name)	Remova um diretório vazio
s = link(name1, name2)	Crie uma nova entrada, name2, apontando para name1
s = unlink(name)	Remova uma entrada de diretório
s = mount(special, name, flag)	Monte um sistema de arquivo
s = umount(special)	Desmonte um sistema de arquivo

Diversas

Chamada	Descrição
s = chdir(dirname)	Altere o diretório de trabalho
s = chmod(name, mode)	Altere os bits de proteção do arquivo
s = kill(pid, signal)	Envie um sinal a um processo
seconds = time(&seconds)	Obtenha o tempo decorrido desde 1º de janeiro de 1970

Exemplos de chamadas ao sistema

Unix	Win32	Descrição
fork	CreateProcess	Crie um novo processo
waitpid	WaitForSingleObject	Pode esperar um processo sair
execve	(none)	CrieProcesso = fork + execve
exit	ExitProcess	Termine a execução
open	CreateFile	Crie um arquivo ou abra um arquivo existente
close	CloseHandle	Feche um arquivo
read	ReadFile	Leia dados de um arquivo
write	WriteFile	Escreva dados para um arquivo
lseek	SetFilePointer	Mova o ponteiro de posição do arquivo
stat	GetFileAttributesEx	Obtenha os atributos do arquivo
mkdir	CreateDirectory	Crie um novo diretório
rmdir	RemoveDirectory	Remova um diretório vazio
link	(none)	Win32 não suporta ligações (link)
unlink	DeleteFile	Destrua um arquivo existente
mount	(none)	Win32 não suporta mount
umount	(none)	Win32 não suporta mount
chdir	SetCurrentDirectory	Altere o diretório de trabalho atual
chmod	(none)	Win32 não suporta segurança (embora NT suporte)
kill	(none)	Win32 não suporta sinais
time	GetLocalTime	Obtenha o horário atual

Estruturas dos Sistemas Operacionais

- Monolíticos;
- Em camadas;
- Máquinas Virtuais;
- Arquitetura Micro-kernel;
- Cliente-Servidor;

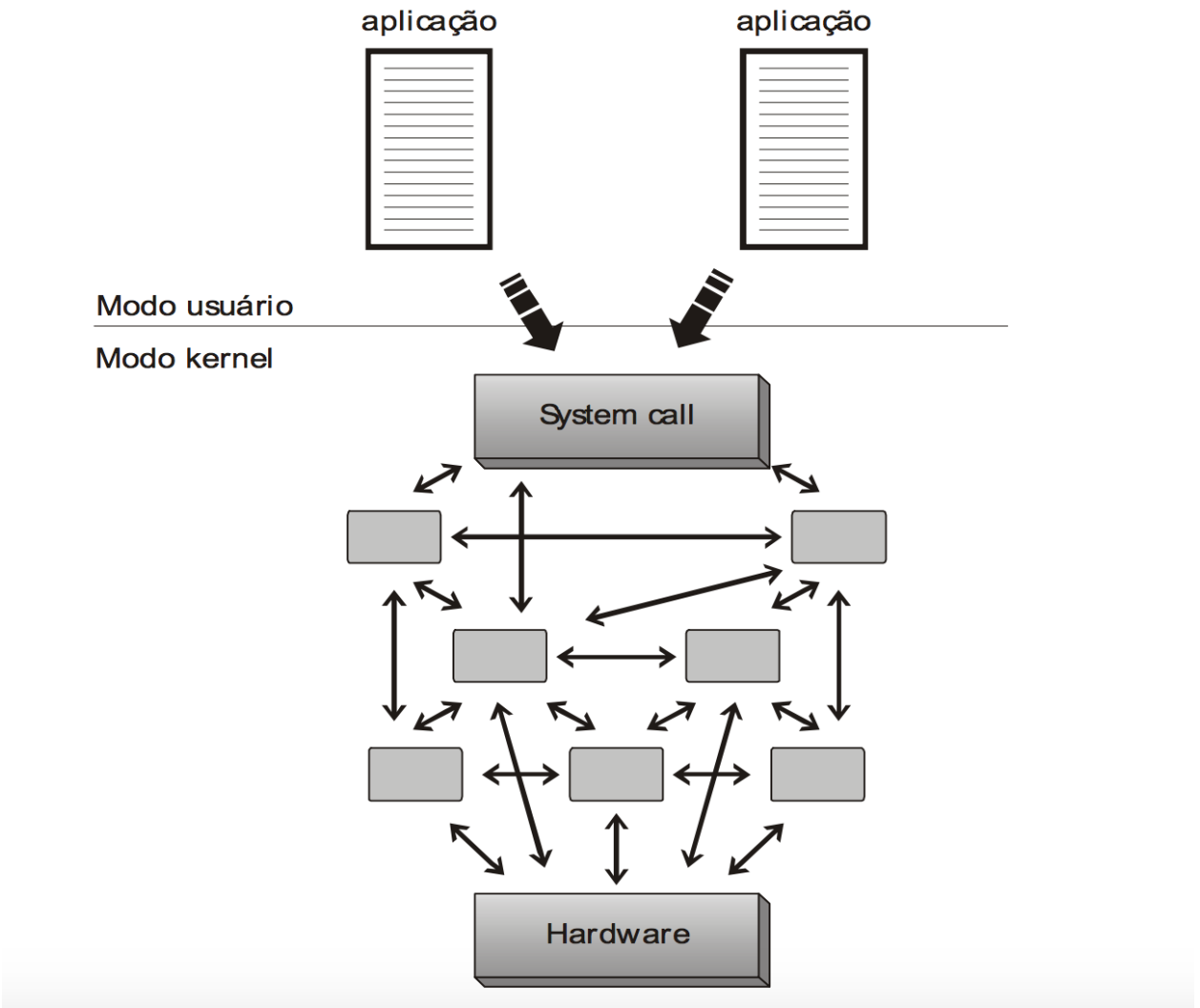
Estruturas dos Sistemas Operacionais - Monolíticos

- Todos os módulos do sistema são compilados individualmente e depois ligados uns aos outros em um único arquivo-objeto;
- O Sistema Operacional é um conjunto de processos que podem interagir entre si a qualquer momento sempre que necessário;
- Cada processo possui uma interface bem definida com relação aos parâmetros e resultados para facilitar a comunicação com os outros processos;
- Simples;
- Primeiros sistemas UNIX e MS-DOS;

Estruturas dos Sistemas Operacionais - Monolíticos

- Os serviços (chamadas) requisitados ao sistema são realizados por meio da colocação de parâmetros em registradores ou pilhas de serviços seguida da execução de uma instrução chamada TRAP;

Estruturas dos Sistemas Operacionais - Monolíticos



Estruturas dos Sistemas Operacionais - Em camadas

- Possui uma hierarquia de níveis;
- Primeiro sistema em camadas: THE (idealizado por E.W. Dijkstra): Possuía 6 camadas, cada qual com uma função diferente; Sistema em batch simples;
- Vantagem: isolar as funções do sistema operacional, facilitando manutenção e depuração.
- Desvantagem: cada nova camada implica uma mudança no modo de acesso.

Estruturas dos Sistemas Operacionais - Máquina Virtual

- Idéia em 1960 com a IBM VM/370;
- Modelo de máquina virtual cria um nível intermediário entre o SO e o Hardware;
- Esse nível cria diversas máquinas virtuais independentes e isoladas, onde cada máquina oferece um cópia virtual do hardware, incluindo modos de acesso, interrupções, dispositivos de E/S, etc.;
- Cada máquina virtual pode ter seu próprio SO;

- Evolução do OS/360 para o TSS/360:
  - Compartilhamento de tempo (TimeSharing);
  - Tanto a multiprogramação quanto a interface com o hardware eram realizadas pelo mesmo processo - sobrecarga gerando alto custo;
- Surge o CP/CMS $\rightarrow$ VM/370 (Mainframes IBM)
  - Duas funções distintas em processos distintos: Ambiente para multiprogramação; Máquina estendida com interface para o hardware;

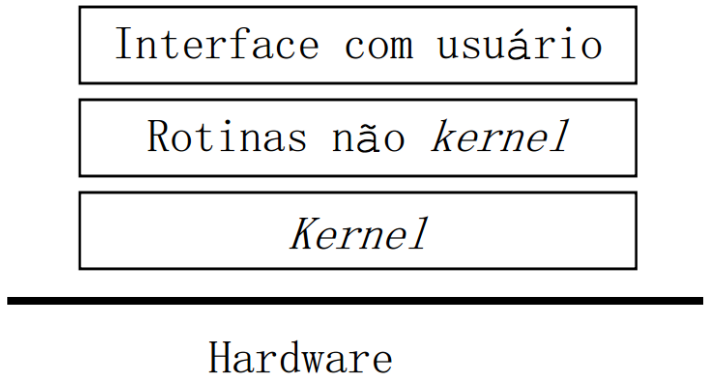
Estruturas dos Sistemas Operacionais - Máquina Virtual

- Monitor da Máquina Virtual (VMM): roda sobre o hardware e implementa multiprogramação fornecendo várias máquinas virtuais é o coração do sistema;
- CMS (Conversational Monitor System): TimeSharing; Executa chamadas ao Sistema Operacional;
- Máquinas virtuais são cópias do hardware, incluindo os modos kernel e usuário;
- Cada máquina pode rodar um Sistema Operacional diferente;

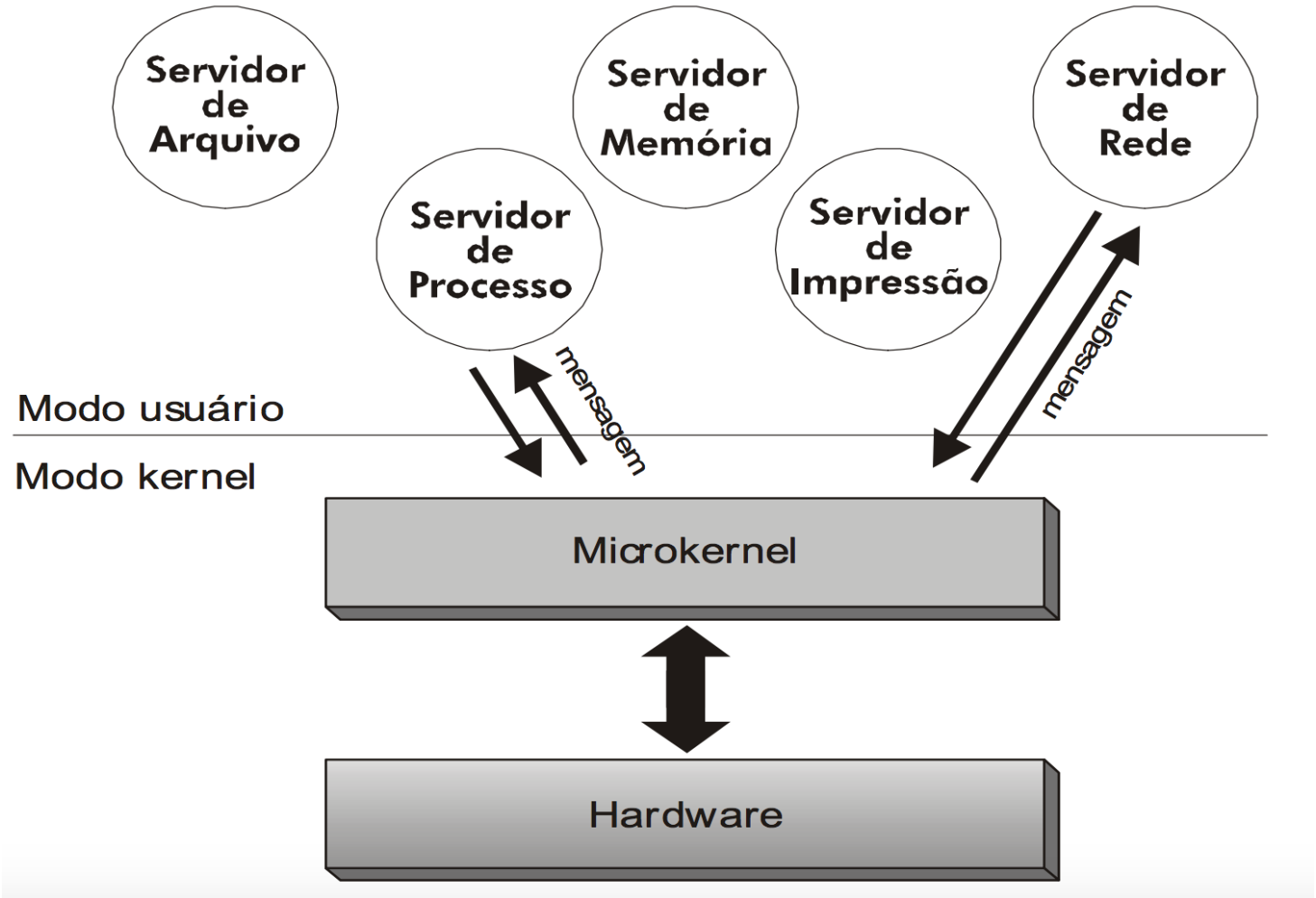
- Vantagens: Flexibilidade;
- Desvantagem: Simular diversas máquinas virtuais não é uma tarefa simples $\rightarrow$ sobrecarga;

Estruturas dos Sistemas Operacionais - Baseados em kernel

- Kernel é o núcleo do Sistema Operacional
- Provê um conjunto de funcionalidades e serviços que suportam várias outras funcionalidades do SO
- O restante do SO é organizado em um conjunto de rotinas não-kernel



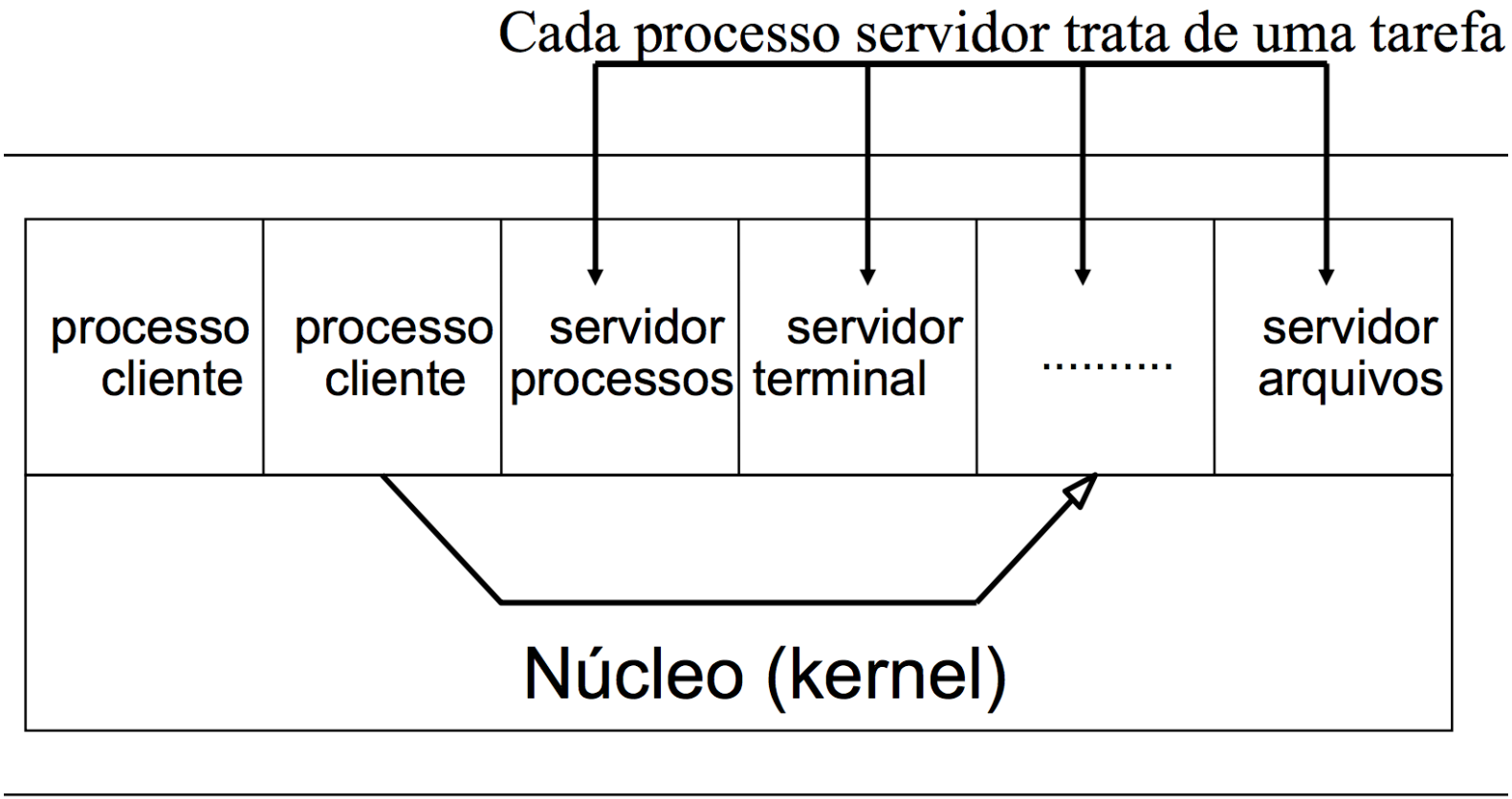
Estruturas dos Sistemas Operacionais - Micro-kernel

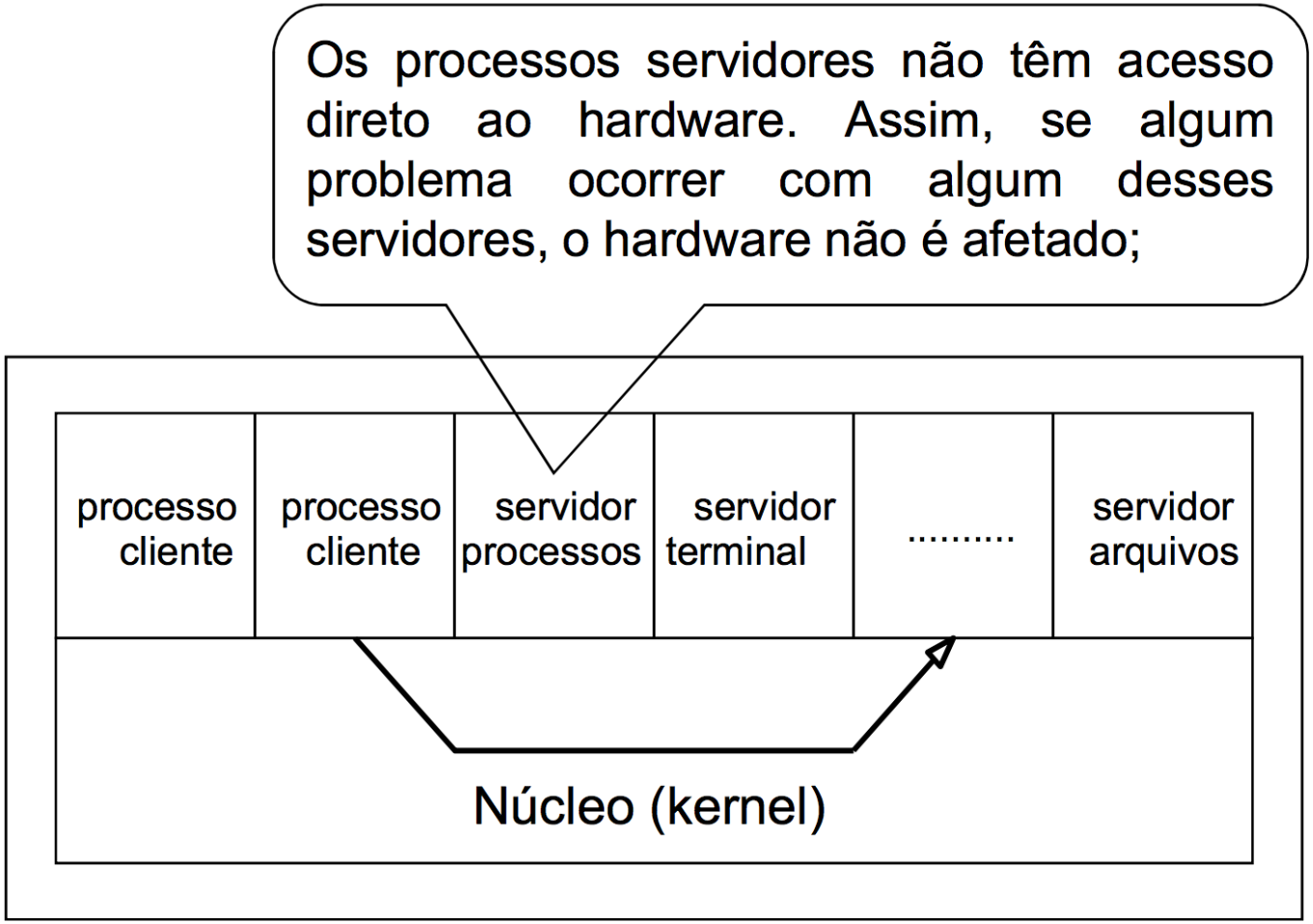


Estruturas dos Sistemas Operacionais - Cliente/Servidor

- Reduzir o Sistema Operacional a um nível mais simples:
- Kernel: implementa a comunicação entre processos clientes e processos servidores $\rightarrow$ Núcleo mínimo;
- Maior parte do Sistema Operacional está implementado como processos de usuários (nível mais alto de abstração);
- Sistemas Operacionais Modernos;

Estruturas dos Sistemas Operacionais - Cliente/Servidor




Os processos servidores não têm acesso direto ao hardware. Assim, se algum problema ocorrer com algum desses servidores, o hardware não é afetado;
processo cliente
processo cliente
servidor processos
servidor terminal
.....
servidor arquivos
Núcleo (kernel)

Próxima aula

Sistema de Arquivos