SISTEMAS OPERACIONAIS

Prof. Dra Alba Cristina Magalhães Alves de Melo
Departamento de Ciência da Computação
Universidade de Brasília (UnB)

Informações Gerais

- Prof. Dra Alba Cristina Melo
  - Mestre em Ciência da Computação - UFRGS - 1991
  - PhD - especialidade Informática, sub-área sistemas operacionais paralelos - INPG - Grenoble, France – 1996
  - Pós-Doutorado na University of Ottawa, Canada, 2008
  - Cientista Visitante na Université Paris-Sud, France, 2011

- Bibliografia:
  - Livro-texto: A. S. Tanenbaum, *Sistemas Operacionais Modernos*, Prentice-Hall do Brasil, 3a edição, 2009.
  - Transparências da aula.

Programa do Curso

- Noções de Base
- Gerência de Processos
- Gerência de Memória
- Gerência de Arquivos
- Gerência de E/S

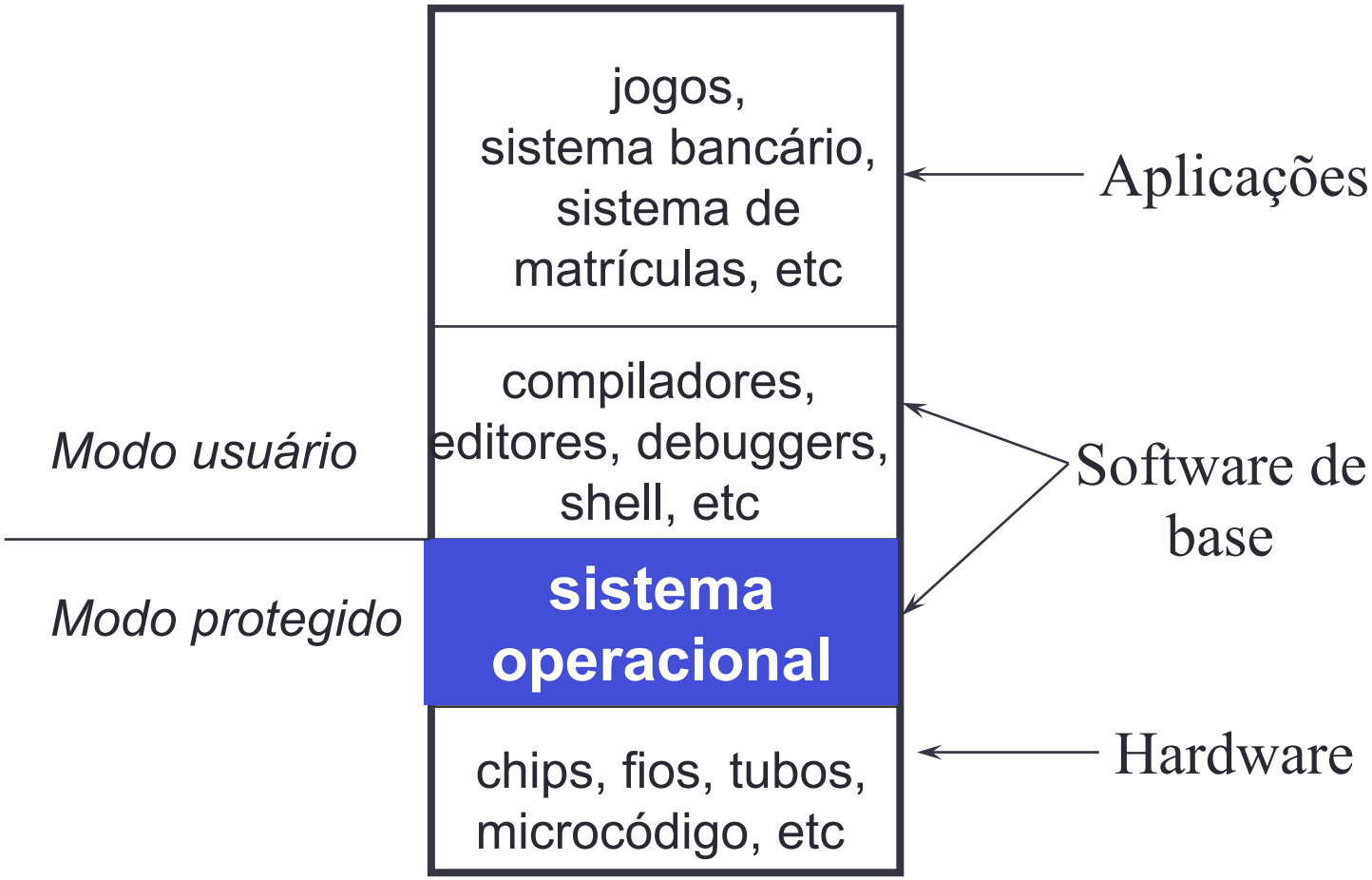
TÓPICO 1

Noções de Base

Noções de Base

- **Computador = hardware + software**
  - hardware: componentes físicos
  - software: conjunto de todos os programas
  - O sistema operacional é um programa ou conjunto de programas.

Noções de Base



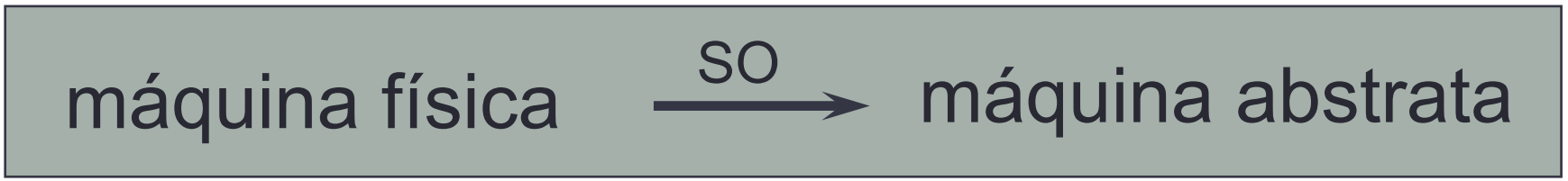
Funções do Sistema Operacional

- Qual a função do sistema operacional?
  - [Madnick74] *O sistema operacional é o gerenciador dos recursos da máquina*
  - [Fortier86] *O sistema operacional fornece ao usuário uma visão de sua interface com a máquina*

➔ ***Um sistema operacional possui duas grandes funções: criar para o usuário uma abstração do hardware e gerenciar os recursos da máquina [Krakowiack87] [Tanenbaum95-09]***

Função 1 - Máquina Estendida

- A primeira função de um sistema operacional é criar para o usuário uma máquina abstrata mais simples que a máquina real.
  - A máquina abstrata ou máquina estendida é geralmente equivalente ao hardware, porém muito mais simples de manipular.



Função 1 - Máquina Estendida

- Exemplo: leitura de um dado gravado em arquivo em disquete

Máquina Física	*Máquina Abstrata*
1) verificar se o motor está ligado 2)posicionar o braço mecânico(bloco,setor, trilha) 3)recuperar o dado 4) colocar o dado na posição indicada	1) abrir o arquivo fd=open(“arquivo”); 2) ler o arquivo read(fd,&dado);

Função 1 - Máquina Estendida

- O programador não quer tratar de todos estes detalhes
- O programador NÃO PODE tratar de todos estes detalhes

➔ A máquina estendida “esconde” a complexidade do hardware e protege os usuários

Função 2 - Gerente de Recursos

- O computador é um conjunto de recursos que serão compartilhados.
  - *recursos físicos*: processadores, memórias, discos, terminais, etc.
  - *recursos abstratos*: processos, arquivos, etc.
- Para todo recurso, o sistema operacional deve:
  - manter informações sobre o recurso (endereço, estado, etc).
  - decidir quem pode acessar o recurso
  - alocar e liberar o recurso

Função 2 - Gerente de Recursos

- Quanto à utilização de recursos, o SO deve:
  - ser eficiente (maximizar a utilização dos recursos)
  - possuir um tempo de resposta previsível

Histórico

- Os sistemas operacionais estão intimamente ligados às arquiteturas nas quais eles rodam. Por isso, analisamos a evolução dos sistemas operacionais em função da evolução das arquiteturas.
- O matemático inglês Charles Babbage (1792-1871) constrói o primeiro “computador”, que não funciona devido a problemas tecnológicos.

Histórico
Primeira Geração (1945-1954)

- *Hardware*: válvulas e painéis. No início dos anos 50 aparecem os cartões perfurados e as impressoras de linha.
- *Sistema operacional*: Não existe. O programador utiliza diretamente a console da máquina para executar seu programa (linguagem de máquina ou fios). Sistema mono-usuário. Um único grupo de pessoas concebia, construía, programava, utilizava e fazia a manutenção das máquinas.
- *Linguagem*: linguagem de máquina

Histórico
Segunda Geração (1955-1964)

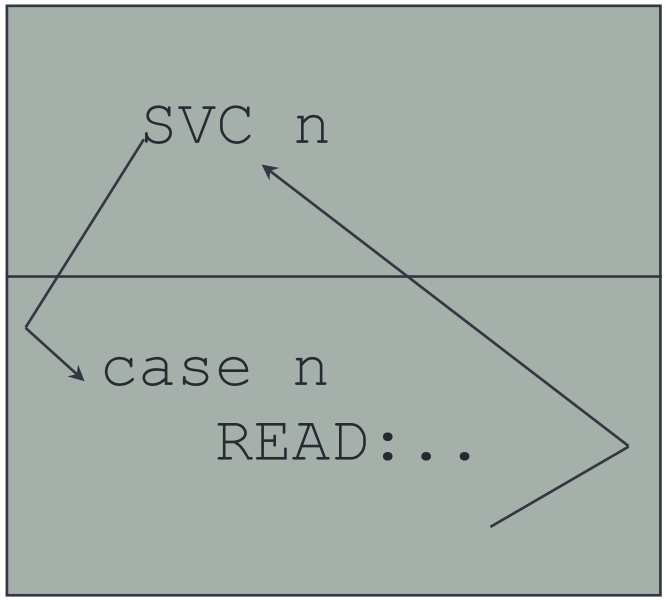
- *Hardware*: transistores. Os computadores começam a ser viáveis comercialmente. Aparição da fita magnética.
- *Sistema operacional:* É feita a separação entre conceptores, construtores, operadores, programadores e técnicos de manutenção. Existe um pseudo-SO que lê os cartões, executa o programa e escreve os resultados na impressora. Sistema mono-usuário.

Histórico
Segunda Geração (1955-1964)

- *Sistemas batch*: conceito de job (programa ou conjunto de programa com características similares). Cartões de início e fim de job. Pseudo-linguagem de controle de execução (wfl), transferência de controle automática entre jobs. A execução de um job pode ser feita ao mesmo tempo que a E/S de outros.

Histórico
Segunda Geração (1955-1964)

- *Modo Monitor/usuário*: é feita a proteção do SO contra o usuário. O SO é acessado através de chamadas ao sistema.



Histórico
Segunda Geração (1955-1964)

- *Temporizador:* O usuário executa durante um tempo pré-definido (objetivo: evitar loops infinitos)
- Menos interação entre o usuário e a máquina
- *Linguagens*: Fortran, Assembly, COBOL

Histórico
Terceira Geração (1965-1980)

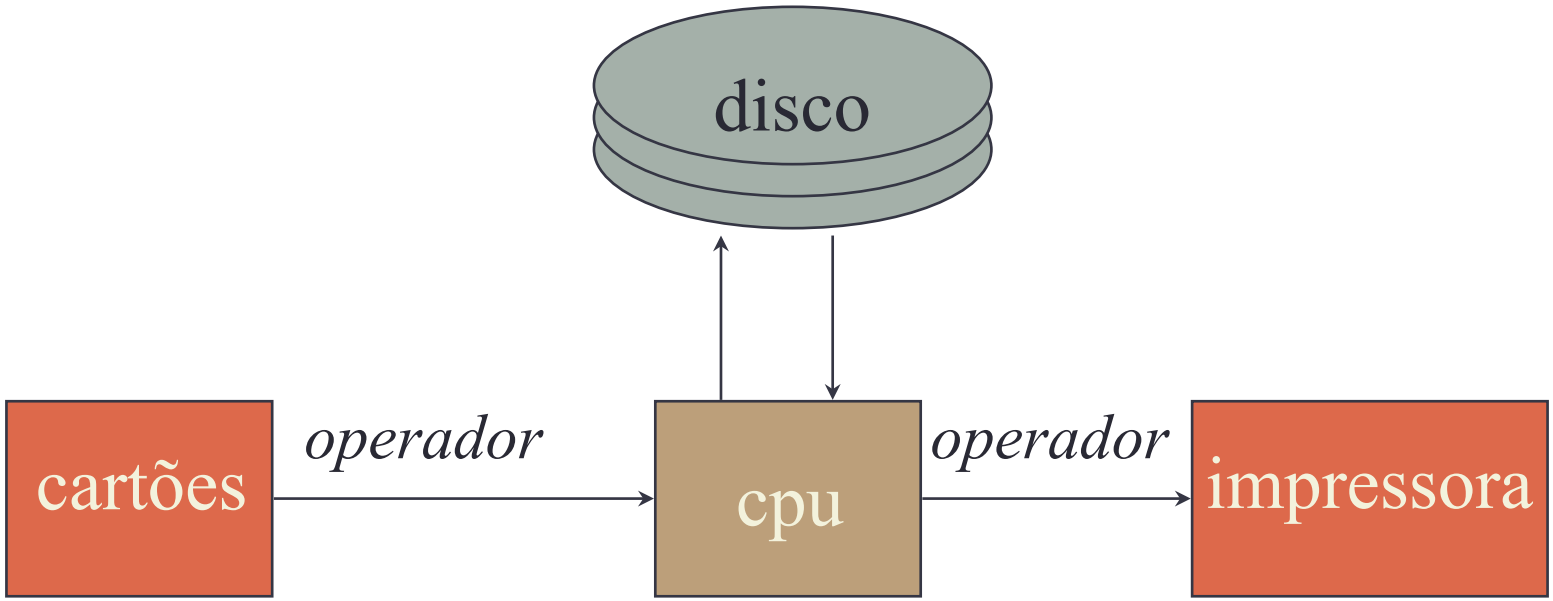
- *Hardware*: circuitos integrados. Aparição do disco. Surgem duas linhas diferentes de máquina: comercial e calculo científico. A IBM tenta conciliar as duas com a linha System/360. Surgem os minicomputadores (DEC PDP-1 com 4k de RAM e palavras de 18 bits - 1961).

Histórico
Terceira Geração (1965-1980)

- *Sistema Operacional*:
  - *Portabilidade*: IBM cria a linha 360 para rodar em várias configurações de máquinas. SO enorme e complexo.
  - *Independência de dispositivos*: execução de um mesmo programa com cartões e impressora ou com fita magnética. Dispositivos lógicos de E/S.

Histórico
Terceira Geração (1965-1980)

- *SPOOL* (Simultaneous Peripheral Operation On Line): leitura dos cartões e escrita em disco.



Histórico
Terceira Geração (1965-1980)

- *Multiprogramação*: Surgiu para evitar que a CPU fique ociosa enquanto o processador espera o término de uma operação de E/S. Vários jobs estão em execução ao mesmo tempo. Aparecem os processadores dedicados à E/S
- *Sistemas time-sharing*: visam garantir um tempo de resposta menor aos usuários. Surgem os terminais.

Histórico
Terceira Geração (1965-1980)

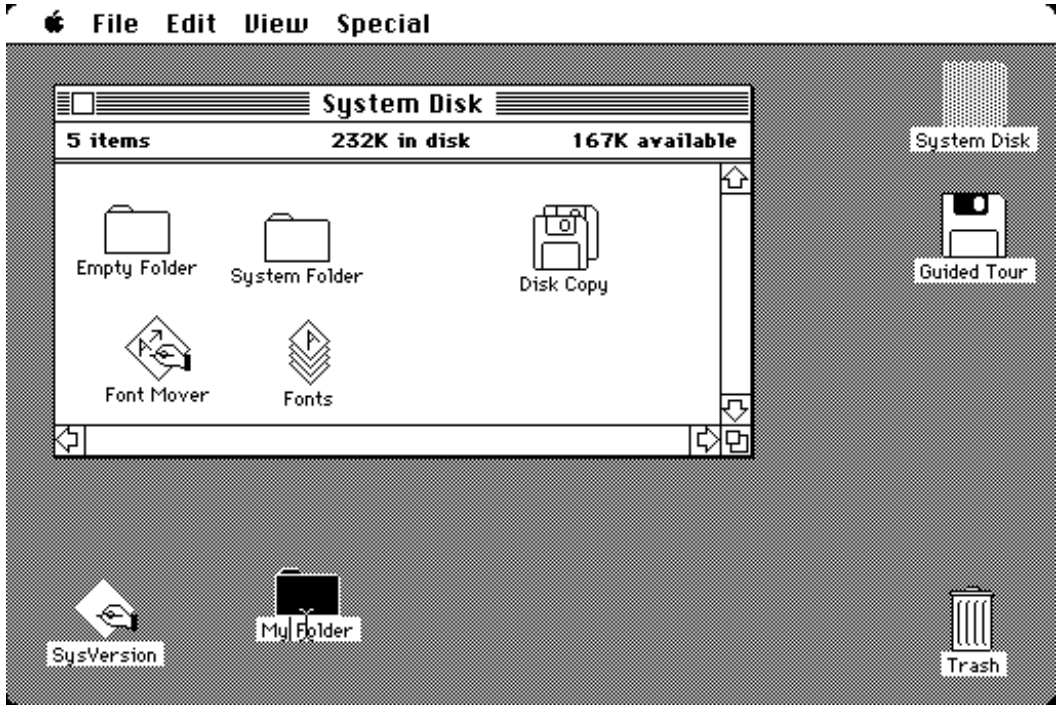
- *Gerência de memória*: para executar um programa que não cabe inteiramente na memória, são propostas duas soluções: overlay e memória virtual.
- *Projeto do Sistema Multics*, cujo objetivo é fornecer bastante poder computacional à centenas de usuários. Projeto conjunto do MIT, GE e Bell. Enorme sucesso acadêmico. Enorme fracasso comercial.

Histórico
Terceira Geração (1965-1980)

- *Nascimento do Unix*, sistema operacional baseado no MULTICS escrito em C.
- *Linguagens:* PL/I, APL, Algol, B, C, etc.

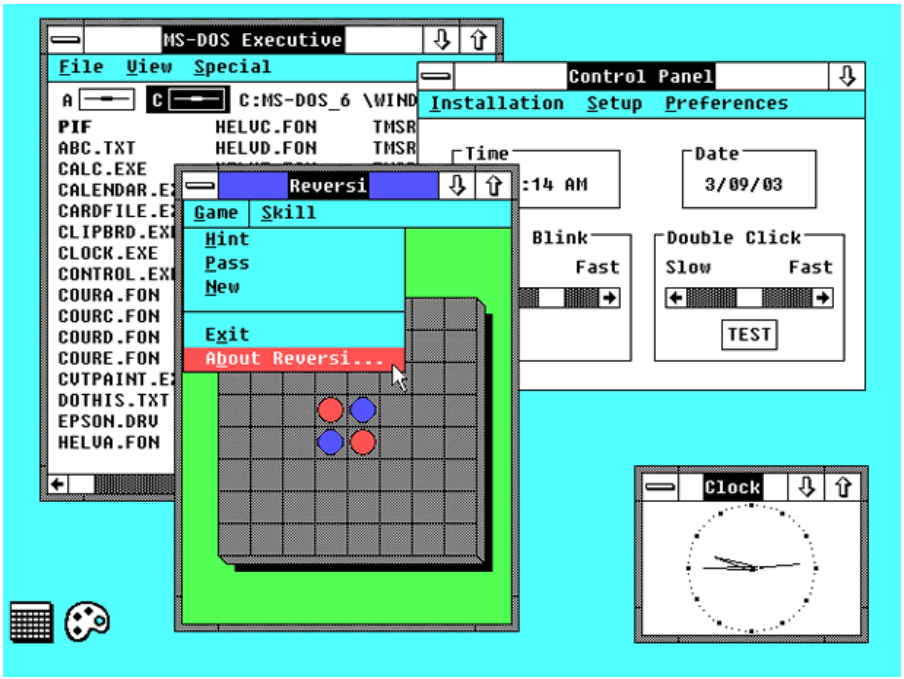
Histórico
Quarta Geração (1981-1990)

- *Hardware*: circuitos LSI (large scale integration) fazem com que o custo do computador caia bastante. Disseminação dos microcomputadores. Aparição do mouse.
- Ênfase em interface amigável: aparição do Macintosh (1984).



Histórico
Quarta Geração (1981-1990)

- *Sistema operacional:* Leva-se em conta a interface com o usuário na concepção dos softwares e do sistema operacional. Dois sistemas dominam o mercado: MS-DOS (Windows 2.0x) e Unix-likes.



Histórico
Quarta Geração (1981-1990)

- *Redes de computadores*: os usuários estão conscientes de que várias máquinas compõem o sistema e endereçam solicitações explicitamente a elas: Apollo/Domain, Appletalk, TokenRing, etc.
- *Sistemas distribuídos*: as diferentes máquinas que compõem o sistema são percebidas pelo usuário como uma única máquina virtual.

Histórico
Quarta Geração (1991-??)

- Grandes redes de computadores (WAN)
- Máquinas paralelas e massivamente paralelas
- Computação Móvel
- Explosão do uso da Internet
- Mercado concentrado nas plataformas Windows e Linux
- Sistemas embarcados
- Computação em nuvem
- Etc...

Estrutura dos Sistemas Operacionais

- Sistemas Monolíticos
- Sistemas em Camadas
- Máquinas Virtuais
- Micro-kernel (Cliente/Servidor)
- Exo-kernel

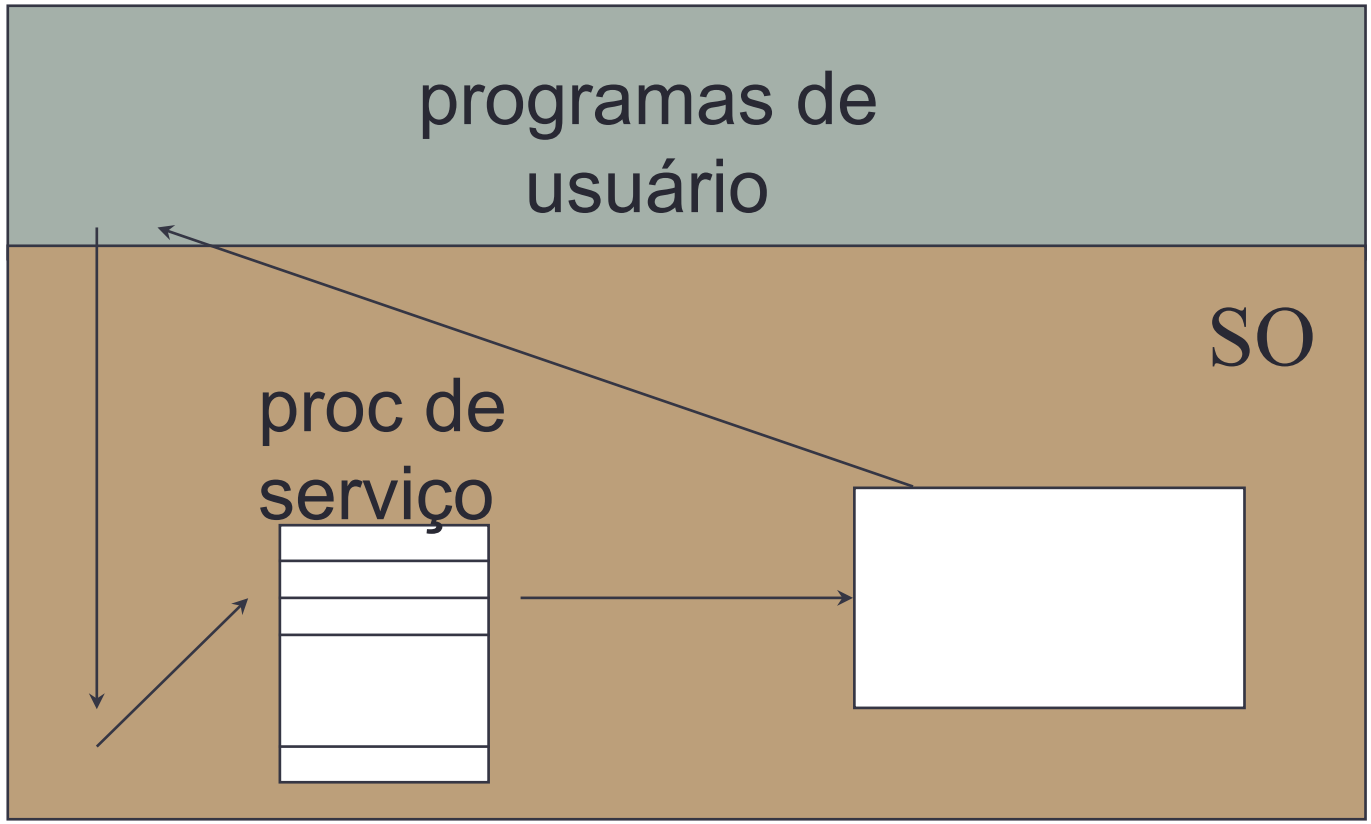
Sistemas Monolíticos

- É a organização de SO mais comum. Não há estrutura. O sistema é um conjunto de procedimentos que chamam um ao outro.
- Quanto ao tempo de resposta, é a organização mais eficaz.
- O sistema operacional roda em modo kernel enquanto os demais programas rodam em modo usuário.

Sistemas Monolíticos

- Pode-se obter um mínimo de estruturação se os procedimentos são forçados à fazer uma SVC (supervisor call) ou gerar um trap. É a organização do Unix comercial.
- Possui geralmente três “camadas”: um procedimento principal que chama os procedimentos de serviço, procedimentos de serviço que tratam as chamadas ao sistema e procedimentos que ajudam os procedimentos de serviço.

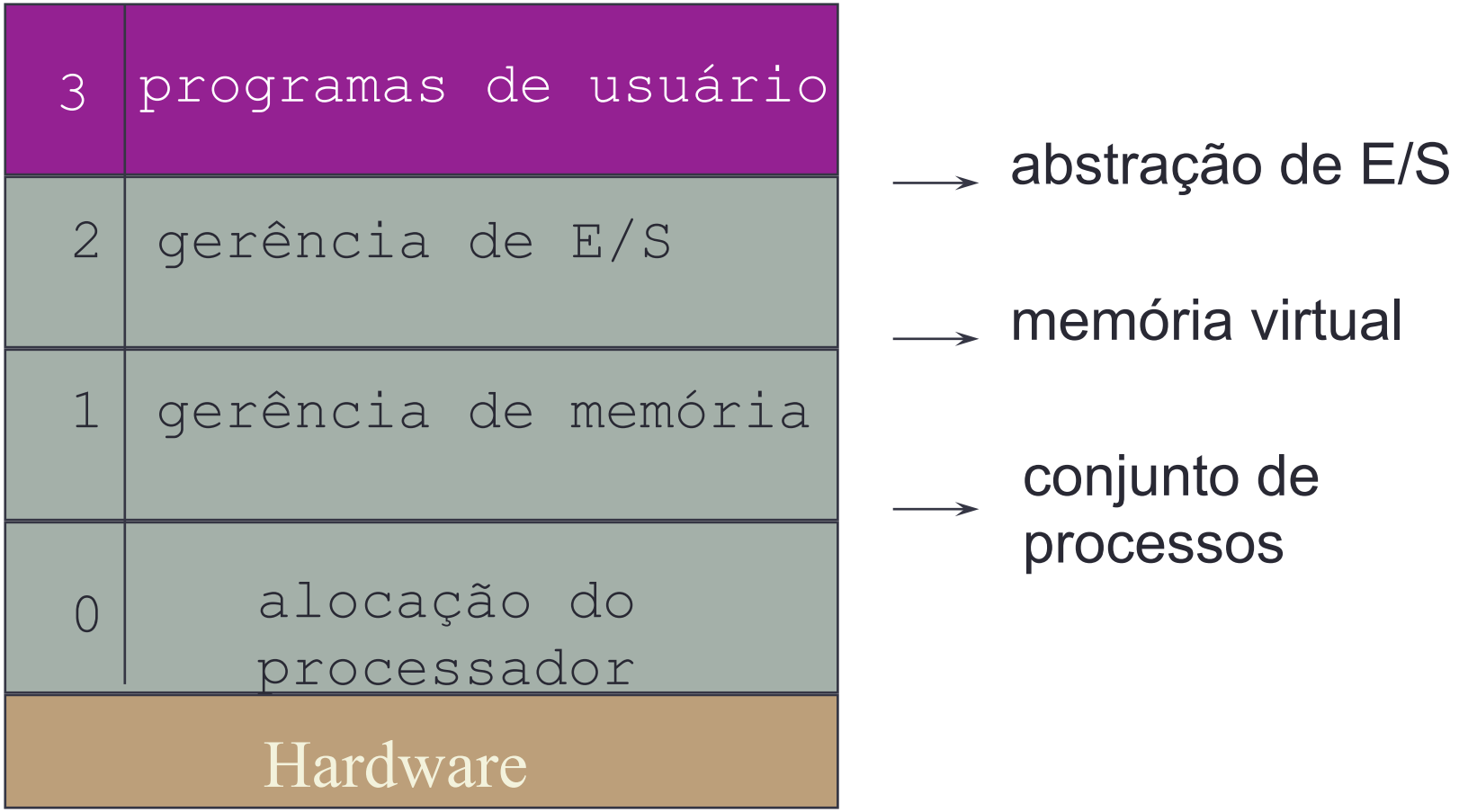
Sistemas Monolíticos



Sistemas em Camadas

- O sistema é organizado em camadas funcionais. Cada camada só faz chamada à camada imediatamente inferior.
- Exemplos: THE (1968 - Dijkstra), MULTICS (BELL, MIT)
- A noção de camadas é fortemente reforçada pelo hardware.

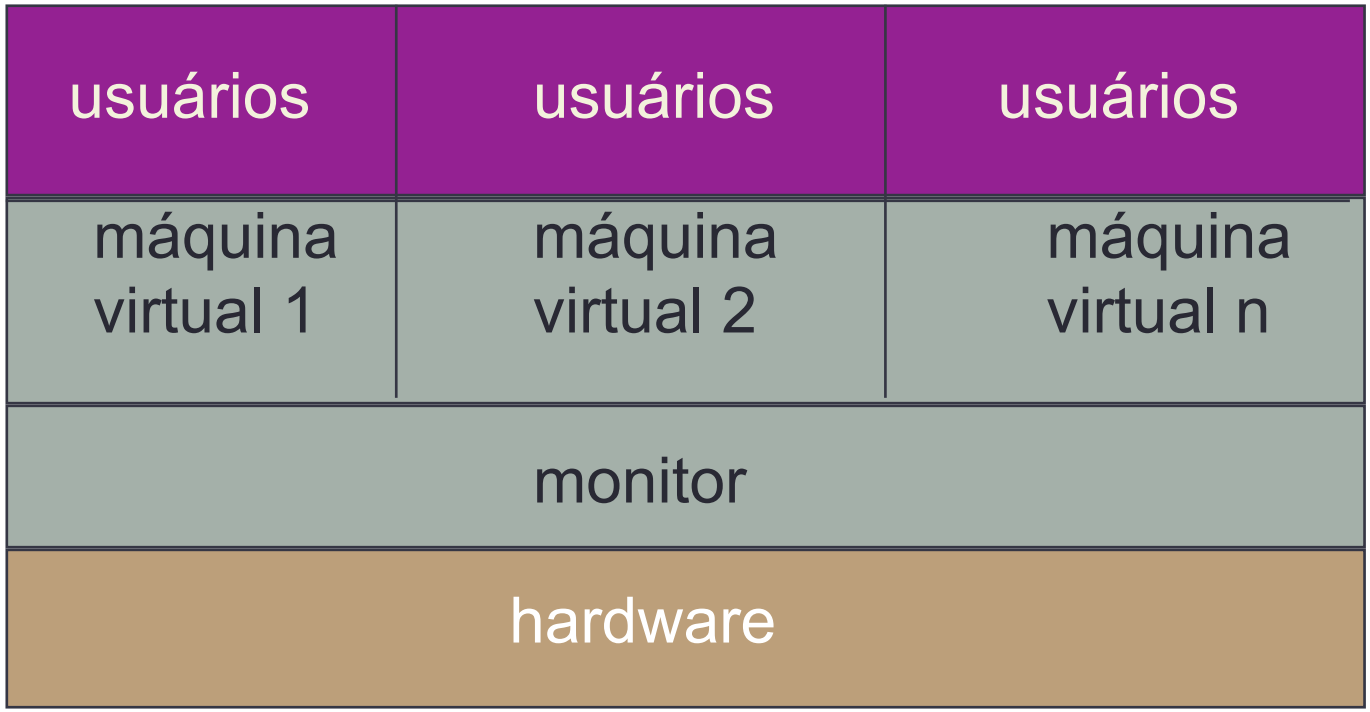
Sistemas em Camadas



Máquinas Virtuais

- Os sistemas operacionais estruturados como máquinas virtuais possuem, no mais baixo nível, um monitor da máquina virtual, que simplesmente implementa a multiprogramação.
- Em cima do monitor, várias máquinas virtuais podem ser utilizadas.
- As máquinas virtuais implementam uma cópia fiel do hardware, com modo kernel/usuário, E/S, interrupções, etc.

Máquinas Virtuais

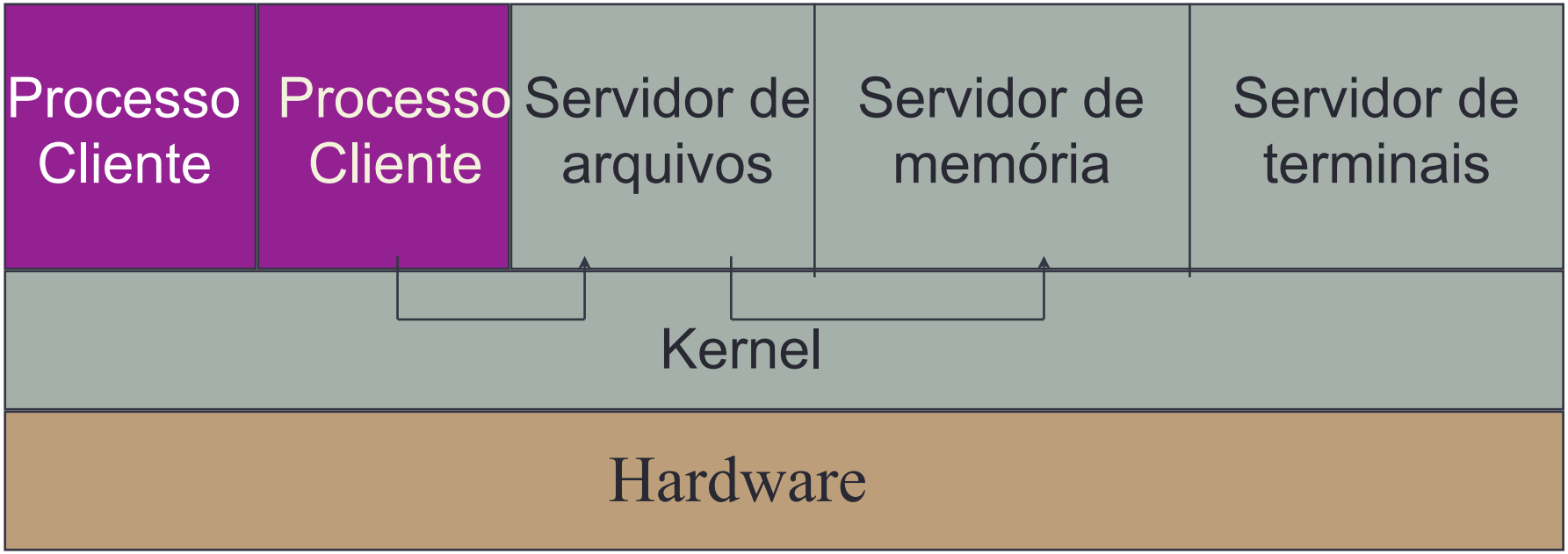


Exemplo: VM/370 da IBM

Micro-kernel

- A maior parte das funções do SO é implementada à nível de usuário (processos clientes).
- O kernel funciona como um servidor para os processos clientes e implementa simplesmente a abstração de processos e a comunicação entre eles.
- Esta estrutura, além de tornar mais simples o projeto do SO, faz com que ele fique mais confiável. Uma pane em um servidor de arquivos, e.g, não derruba o kernel.
- Os processos interagem com o sistema operacional através de troca de mensagens.

Modelo Cliente/Servidor



Exemplo: sistema operacional distribuído Mach

Exo-kernels

- Concentra-se na multiplexação segura de hardware.
- Torna o hardware visível às aplicações (expõe o hardware).
- De posse das primitivas básicas de hardware, podem ser implementadas em modo usuário as abstrações tradicionalmente oferecidas pelo SO.

Exo-kernels

- O conceito tradicional de sistema operacional é dividido em duas partes:
  - Exokernel: faz a multiplexação segura entre recursos de hardware, protegendo os mesmos.
  - LibOS (sistemas operacionais biblioteca): conjunto de bibliotecas que gerenciam recursos e oferecem as abstrações de alto nível para as aplicações.