Sistemas Operacionais

Introdução

Prof. André D'Amato

andredamato@utfpr.edu.br

Sistemas de computação

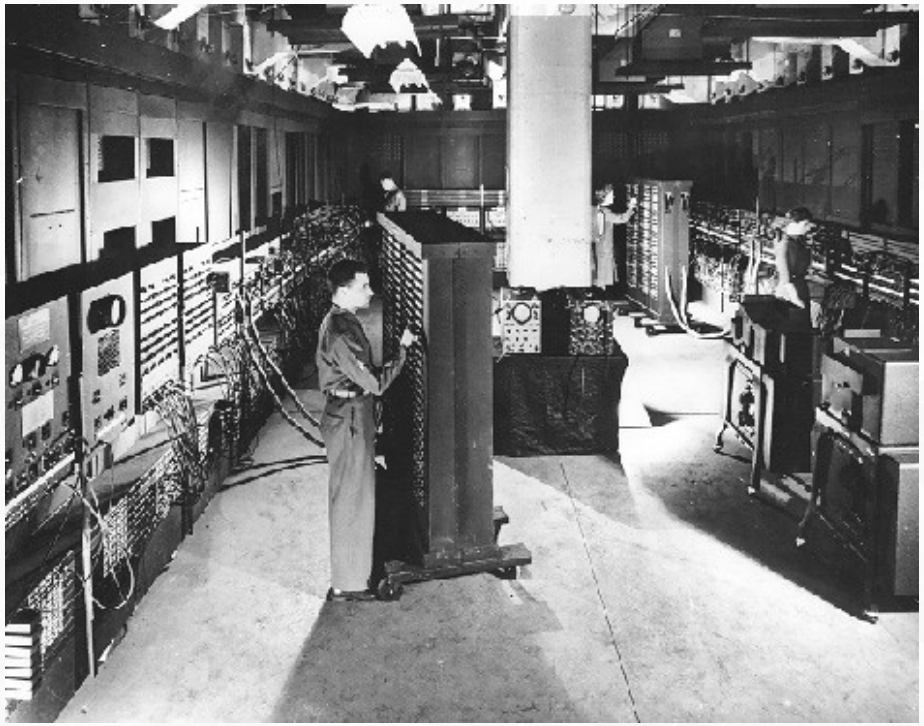
- Hardware
  - CPU + memória + dispositivos de E/S
- Sistemas Operacionais
- Aplicações
  - Objetivo real dos sistemas de computação
  - Bancos de dados, automação, jogos, etc
- Usuários
  - Definem problemas de computação a serem resolvidos
  - Pessoas, máquinas, outros computadores

Sistemas Operacionais

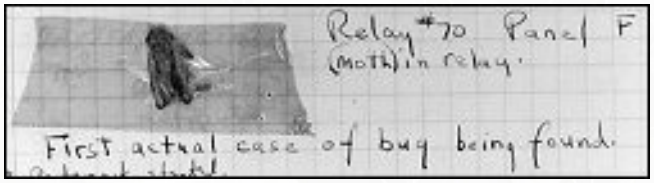
- Perspectiva da máquina virtual
  - SO extende o hardware até implementar uma interface de alto nível para aplicações
- Perspectiva do gerenciador de recursos
  - SO gerencia os recursos do sistema (processadores, memória, discos, etc) para comodidade das aplicações

Perspectiva histórica

- Primeira geração (1945 - 1955)
  - Tubos de vácuo
  - Nenhum software
  - Operado por conexão de cabos e chaves



ENIAC (1946)



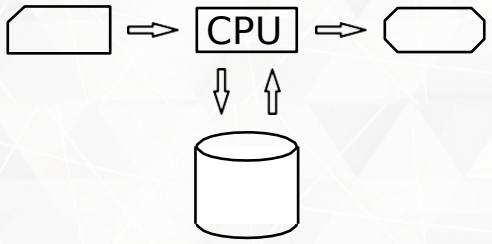
Primeiro bug 'pego' por
Grace Murray Hopper, 1945.

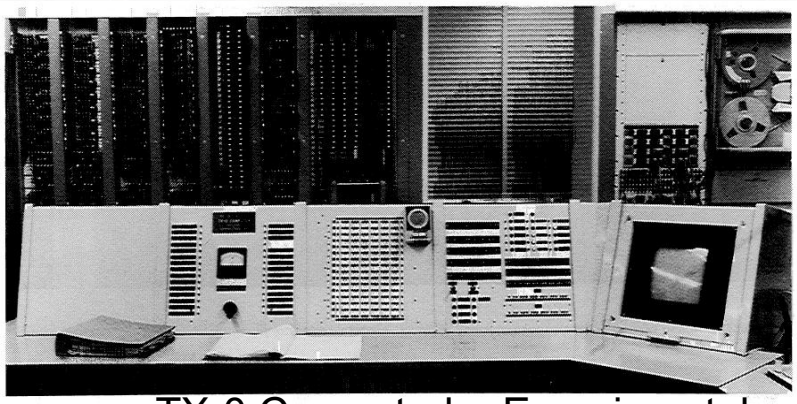
Perspectiva histórica

- Segunda geração (1955 - 1965)
  - Transistores
  - Device drivers
  - Primeiras linguagens de programação (Fortran)
  - Monitor (leitor de cartões perfurados)

  - Batch (offline)

    - Spooler (*Simultaneous Peripheral Operation On-Line*)





TX-0 Computador Experimental Transistorizado (1956)

Perspectiva histórica

- Terceira geração (1965 - 1980)
  - Circuitos Integrados (TI IC/CI)
  - Primeiro SO (IBM OS/360)
  - Multiprogramação (CPU/IO overlap)
  - Time-sharing (MIT CTSS)
  - MULTICS (MIT, BELL, GE)
  - PDP-11 (DEC)
  - UNIX (BELL)

PDP-11/20 (1970)

Perspectiva histórica

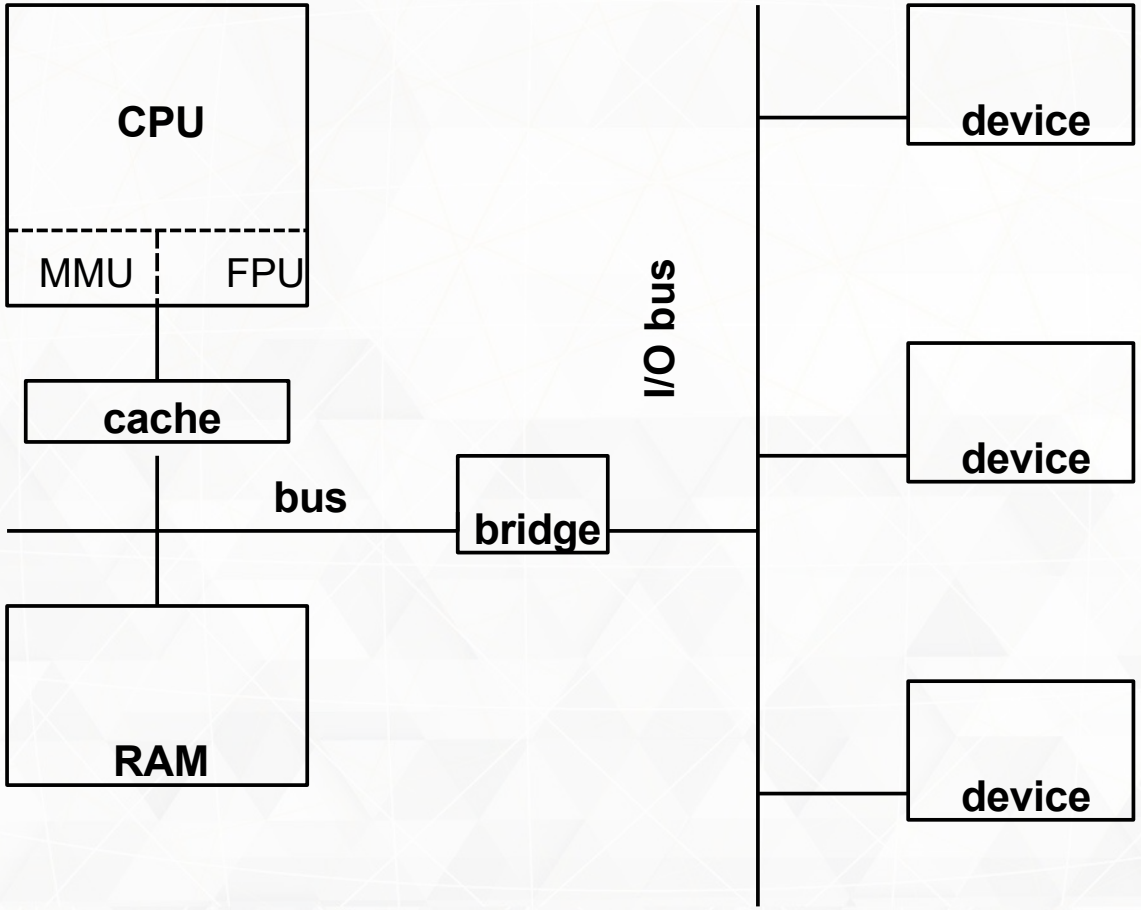
- Quarta geração (1980 - ?)
  - Microprocessador
  - MS-DOS, UNIX
  - Sistemas com rede
  - Sistemas distribuídos
  - Sistemas de tempo-real

Apple MacIntosh SE/30
(1972)

Perspectiva histórica

- Quinta geração (?)
  - Hardware
    - Ubíquo!
    - Paralelo
    - Embarcado
  - Software
    - Interface Homem-máquina revolucionária!
    - Inteligência artificial???
    - Nuvem

Um computador típico



Estruturas de Sistemas de Computação

- Motivação
  - Intermediar operações de CPU e E/S para melhorar performance
  - Evitar interferência inter-processos
- Interrupções
  - Evita espera-ocupada (*busy-waiting*)
  - Dispositivo de E/S recebe uma requisição de serviço e gera uma interrupção quando a requisição for completada
  - Transparente aos processos
  - Direct Memory Access (DMA)
    - Transferência de dados entre dispositivos de E/S e memória principal sem assistência da CPU

Estruturas de Sistemas de Computação

- Proteção de recursos
  - Permite ao SO definir políticas
  - Violações causam uma *trap* para dentro do SO
- CPU
  - Operação em múltiplos modos
    - Modo supervisor: todas as instruções, restrito ao SO
    - Modo usuário: instruções não-privilegiadas (ex: sem E/S)
  - Timer
    - Interrupções do Timer transferem controle ao SO periodicamente
- Memória
  - Memory Management Unit (MMU)
    - Isolamento do SO
    - Espaço de endereçamento separado para cada processo
    - Proteção dos registradores dos dispositivos de E/S

Serviços de sistemas operacionais

- Gerenciamento de Processos
  - Criação e destruição de processos
  - Alocação e liberação de recursos
  - Escalonamento de CPU (e contabilização de processos)
  - Sincronização de processos
  - Comunicação entre processos
  - Tratamento de dead-lock
- Gerenciamento de Memória
  - Alocação e liberação de memória
  - Manutenção da integridade (o que pertence a quem)
  - Swapping
  - Memória virtual

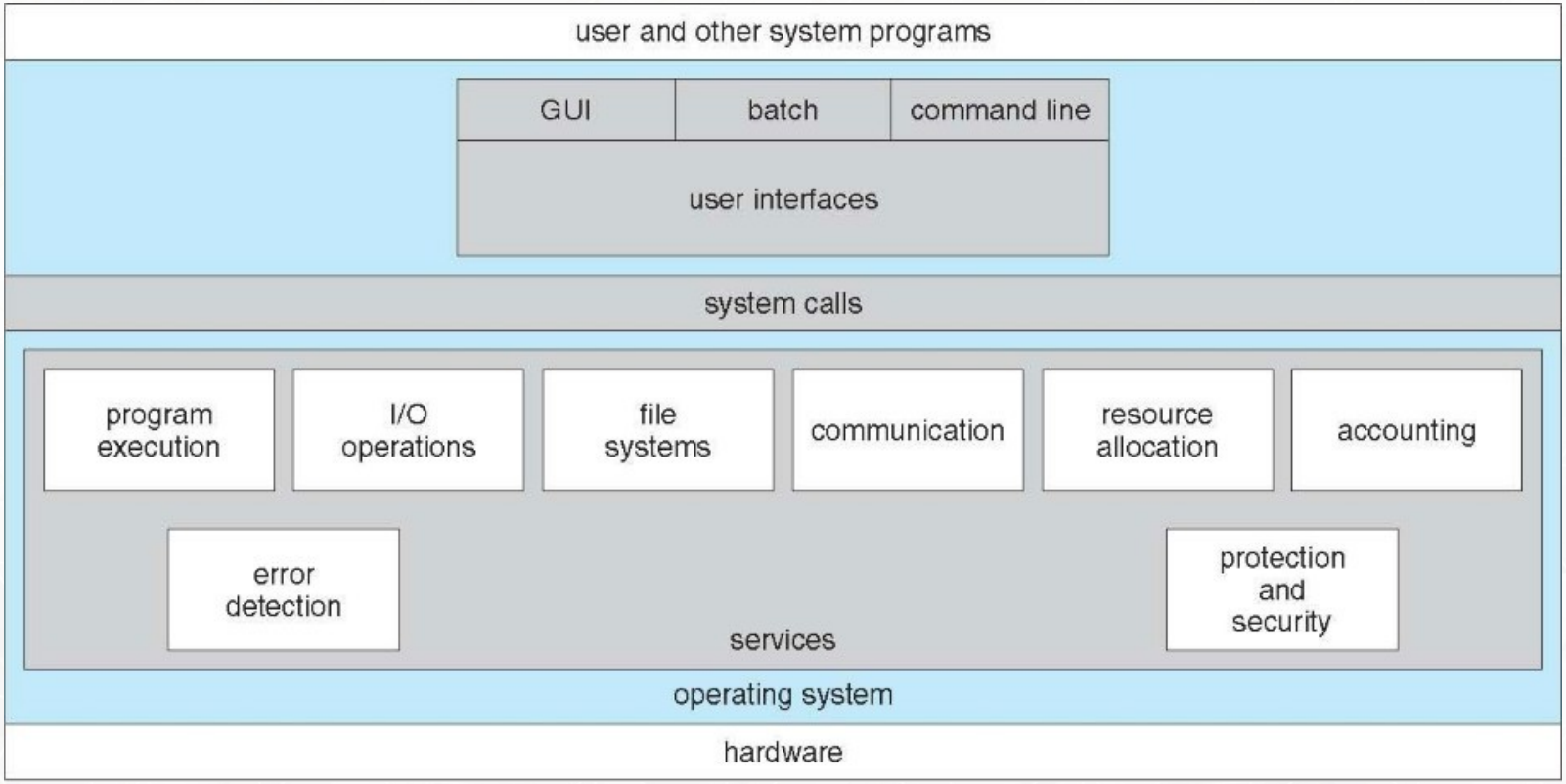
Serviços de sistemas operacionais

- Gerenciamento de I/O
  - Buffering/caching
  - Escalonamento (e.g. disco, rede)
  - Device drivers
- Gerenciamento de arquivos
  - Criação, manipulação e remoção de arquivos
  - Criação, manipulação e remoção de diretórios
  - Mapeamento de arquivos em discos
- Rede
  - Roteamento, acesso ao meio e segurança de mensagens
  - Heterogeneidade
  - Interface de usuário

Serviços de sistemas operacionais

- Proteção
  - Controle de acesso aos recursos
  - Logging
  - Validação de procedimentos
- Interface
  - SO provê serviços para aplicações por meio de APIs (Application Program Interface)
    - Se o SO está em um domínio de proteção diferente do das aplicações (e.g. Kernel), uma system call é usada
  - Interação com usuário
    - Interpretador de comandos (*shell*)
    - Interface gráfica com o usuário (GUI)

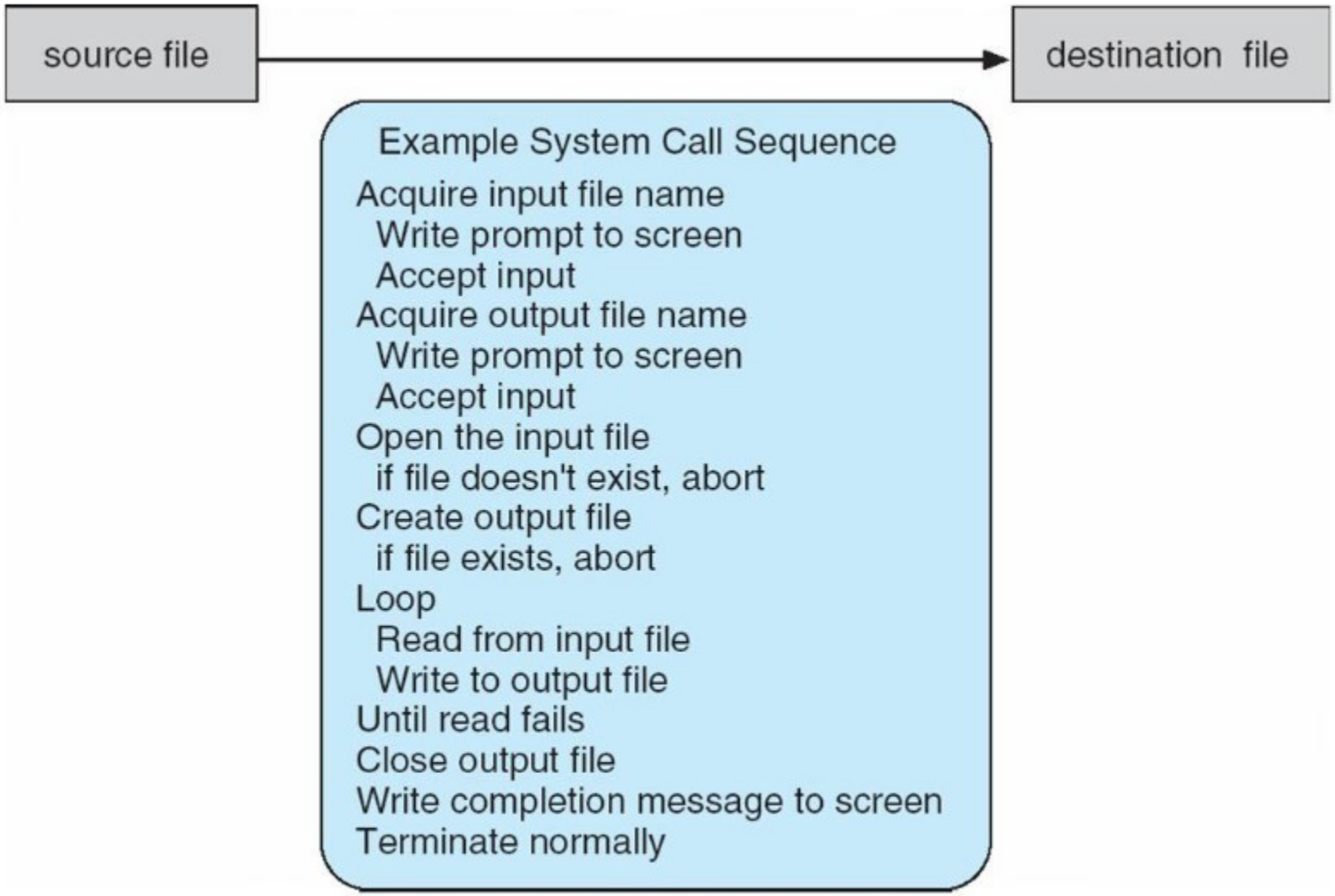
Serviços de sistemas operacionais



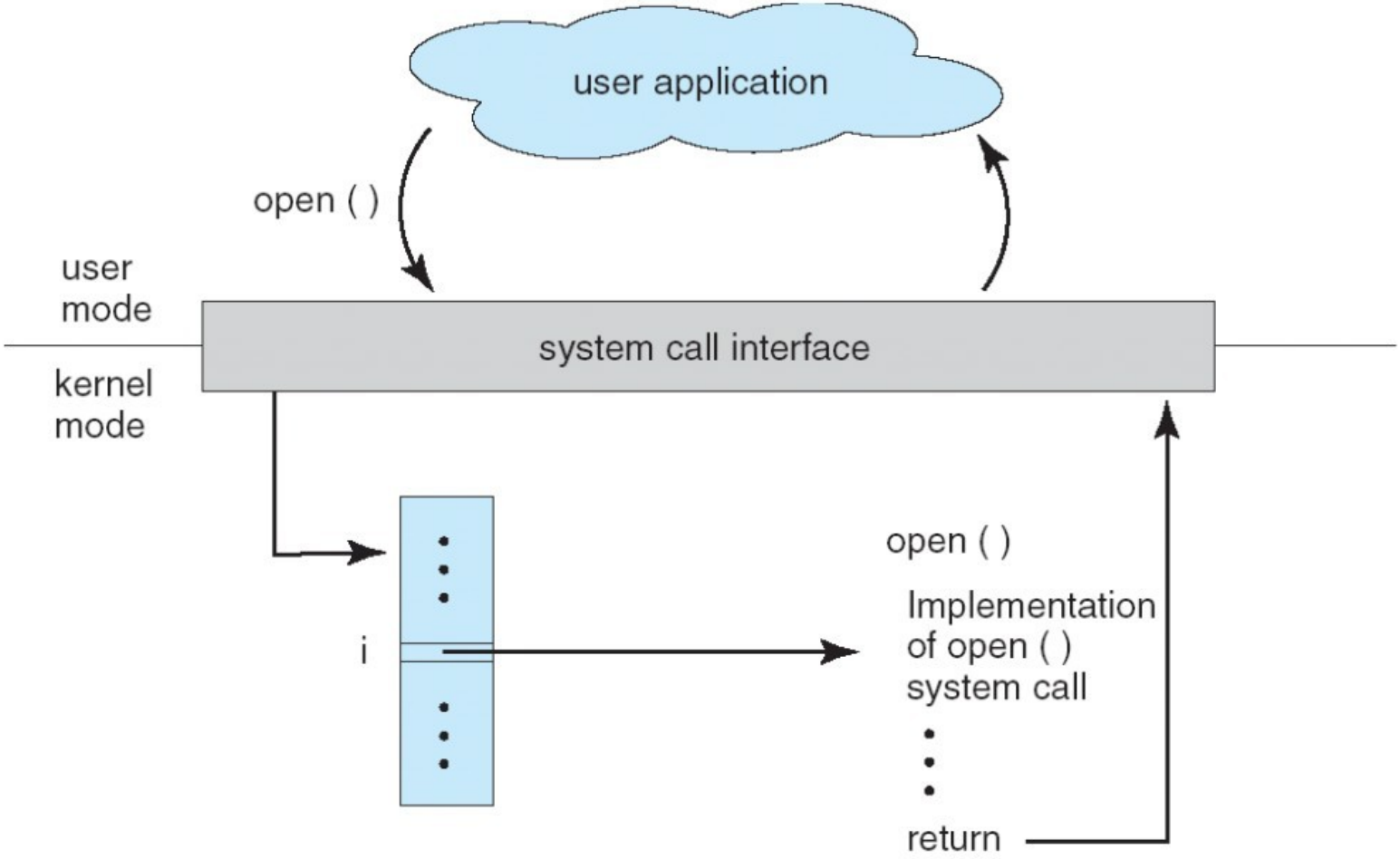
Chamadas de Sistema (*System Call*)

- Interface de programação para acesso aos serviços do SO
- Geralmente acessadas via APIs e bibliotecas
- As três mais comuns
  - Win32 (Windows)
  - POSIX (Unix, Linux, Mac OS X e similares)
  - API JAVA (JVM)
- Por que utilizar APIs ao invés das SysCall direto?

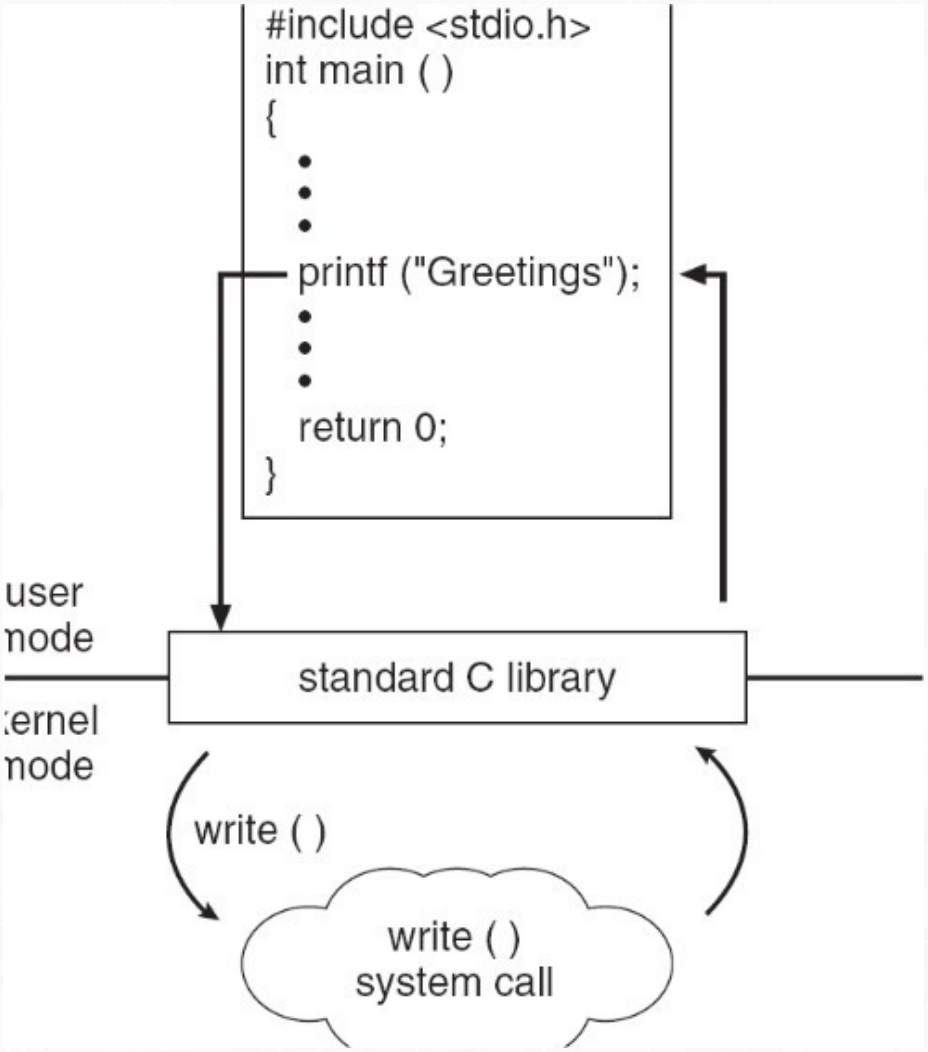
Chamadas de Sistema (*System Call*)



API – System Call

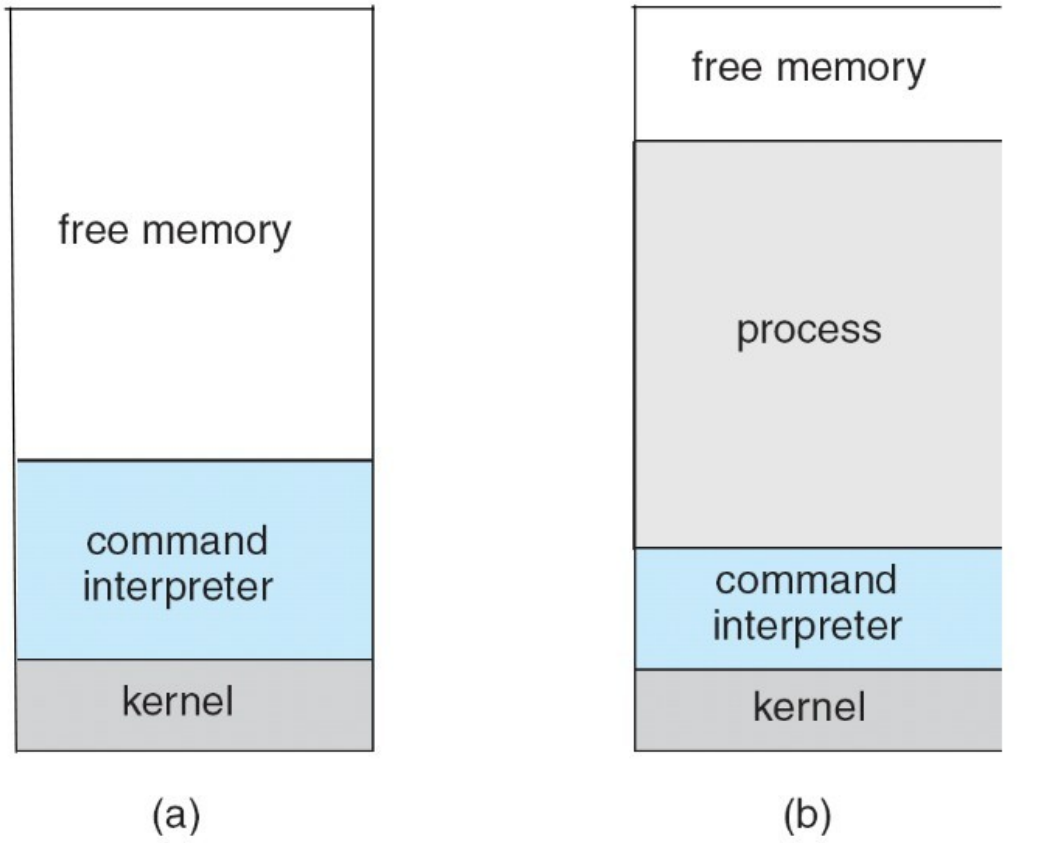


Exemplo da API em C (POSIX)



Controle de Processos

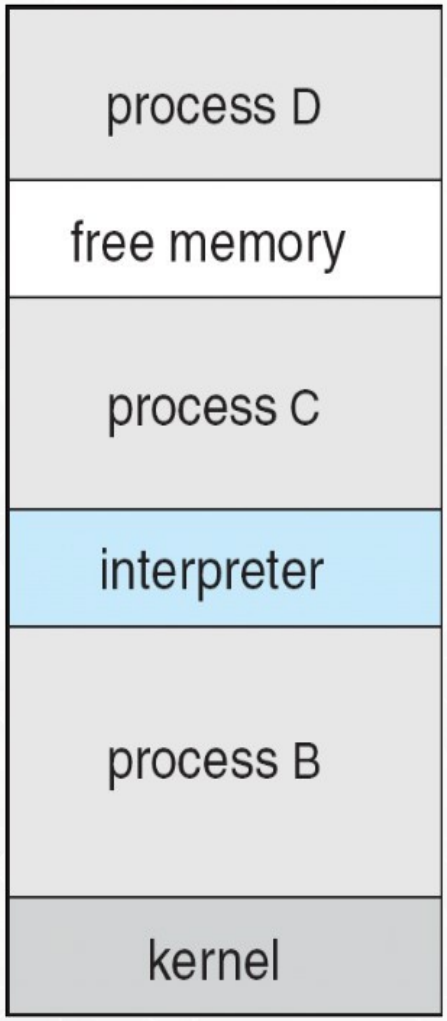
Processos no MS-DOS



(a) Logo após o boot (b) Programa em execução

Controle de Processos

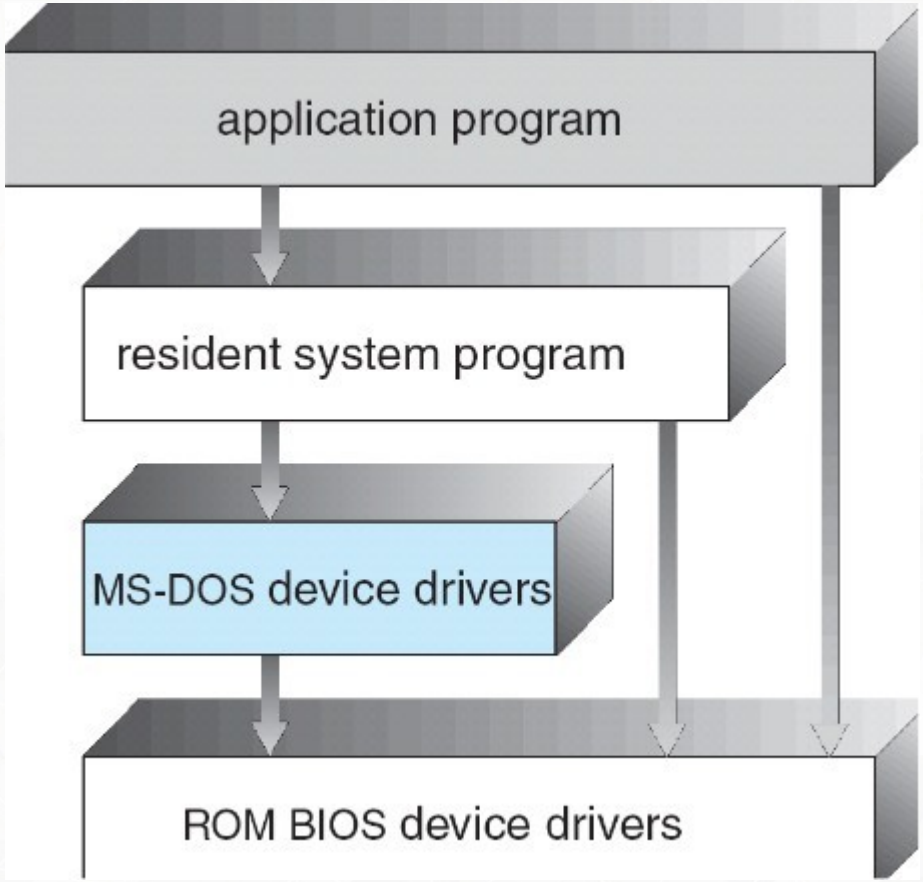
Processos no Linux



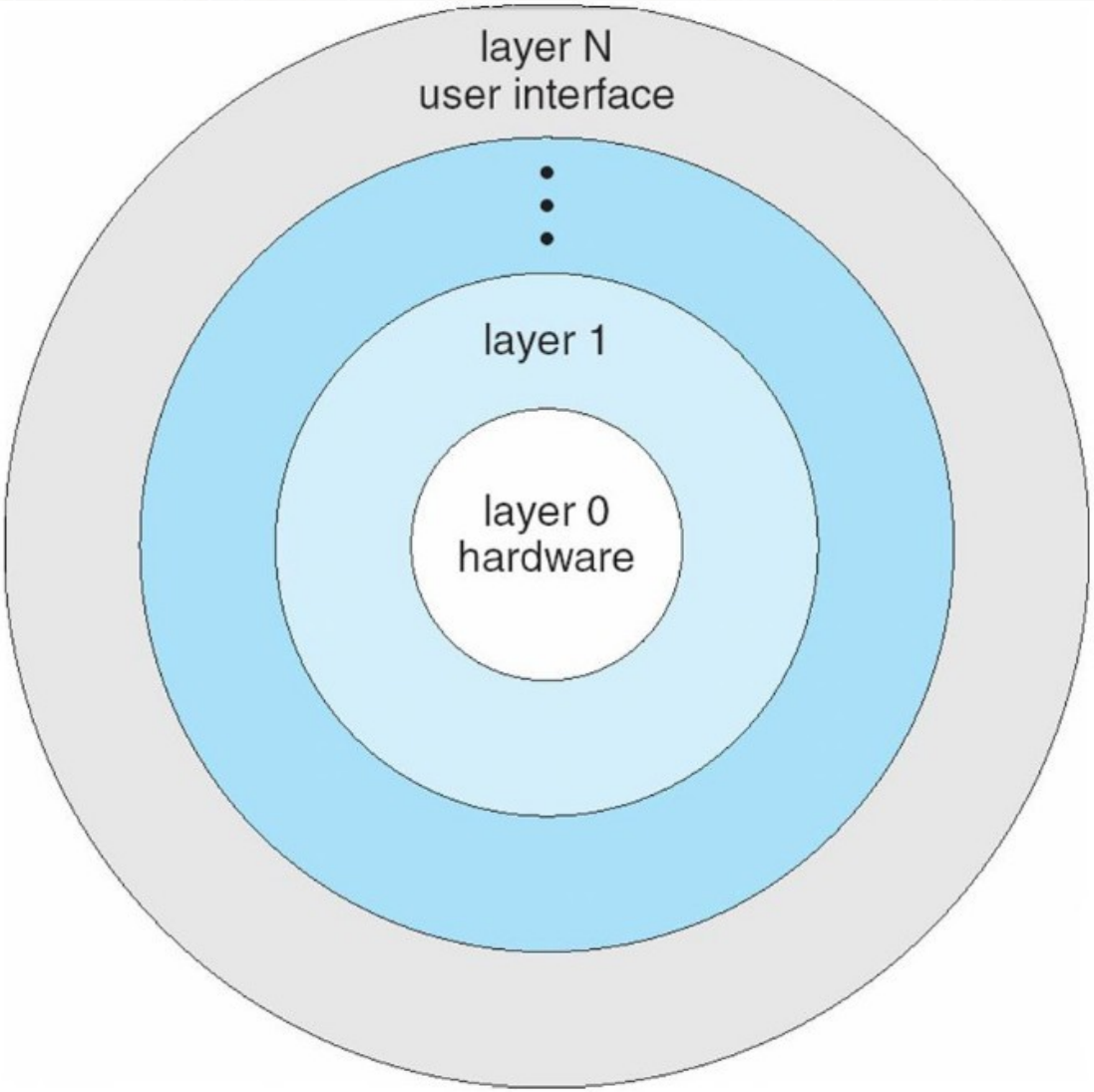
Arquiteturas de Sistemas Operacionais

- Monolítico
  - Todo SO é um único programa, complexo, responsável por todos os serviços
- Máquina Virtual
  - Serviços de SO são entregues como máquinas virtuais privadas para cada processo de aplicação
- Kernel + servidores
  - Partes cruciais do SO, responsáveis por serviços fundamentais, são mantidos em um kernel protegido
  - Serviços avançados são delegados a servidores que operaram como um processo comum

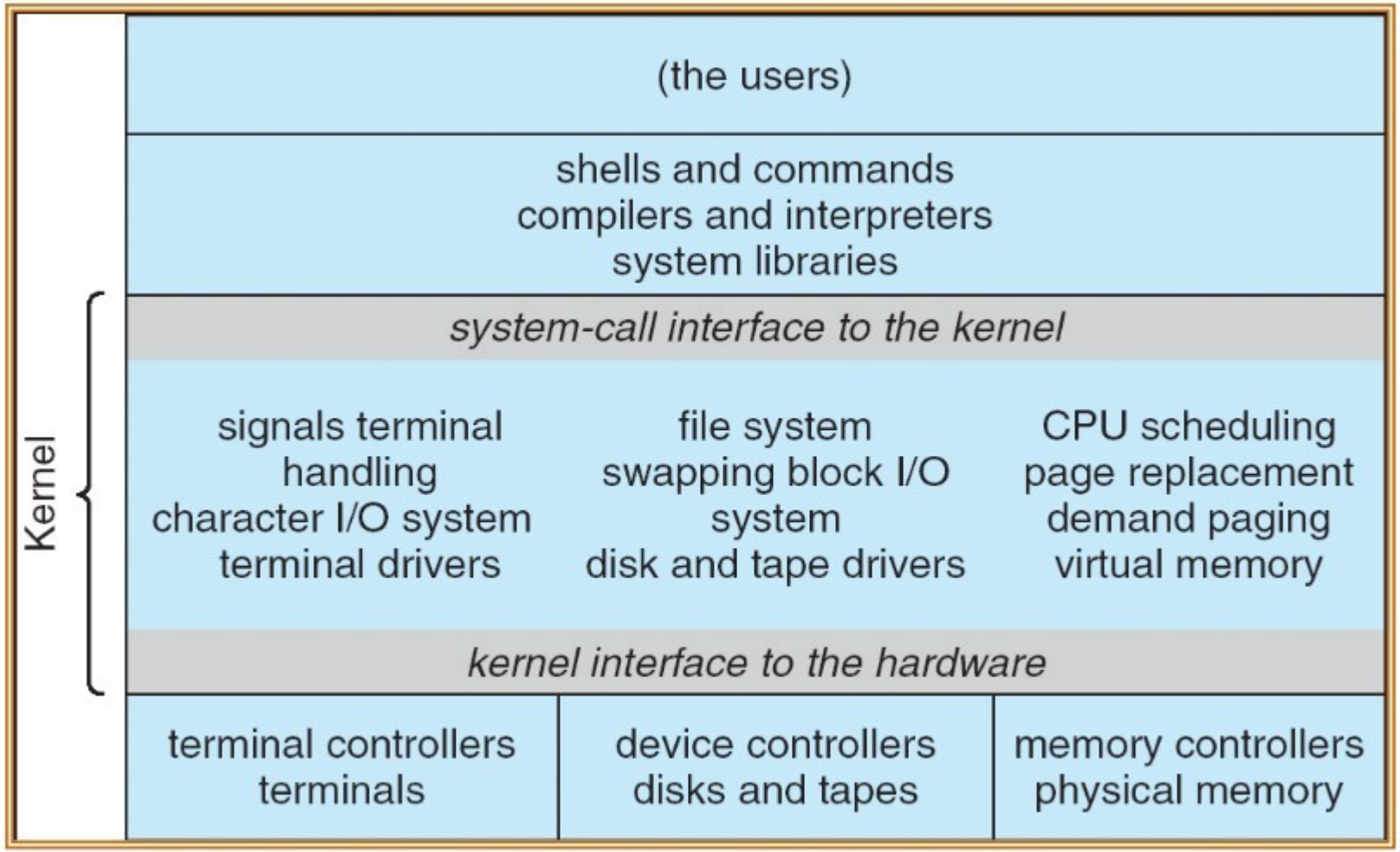
Monolítico


application program
resident system program
MS-DOS device drivers
ROM BIOS device drivers

Em camadas



Camadas do Unix (década 80)



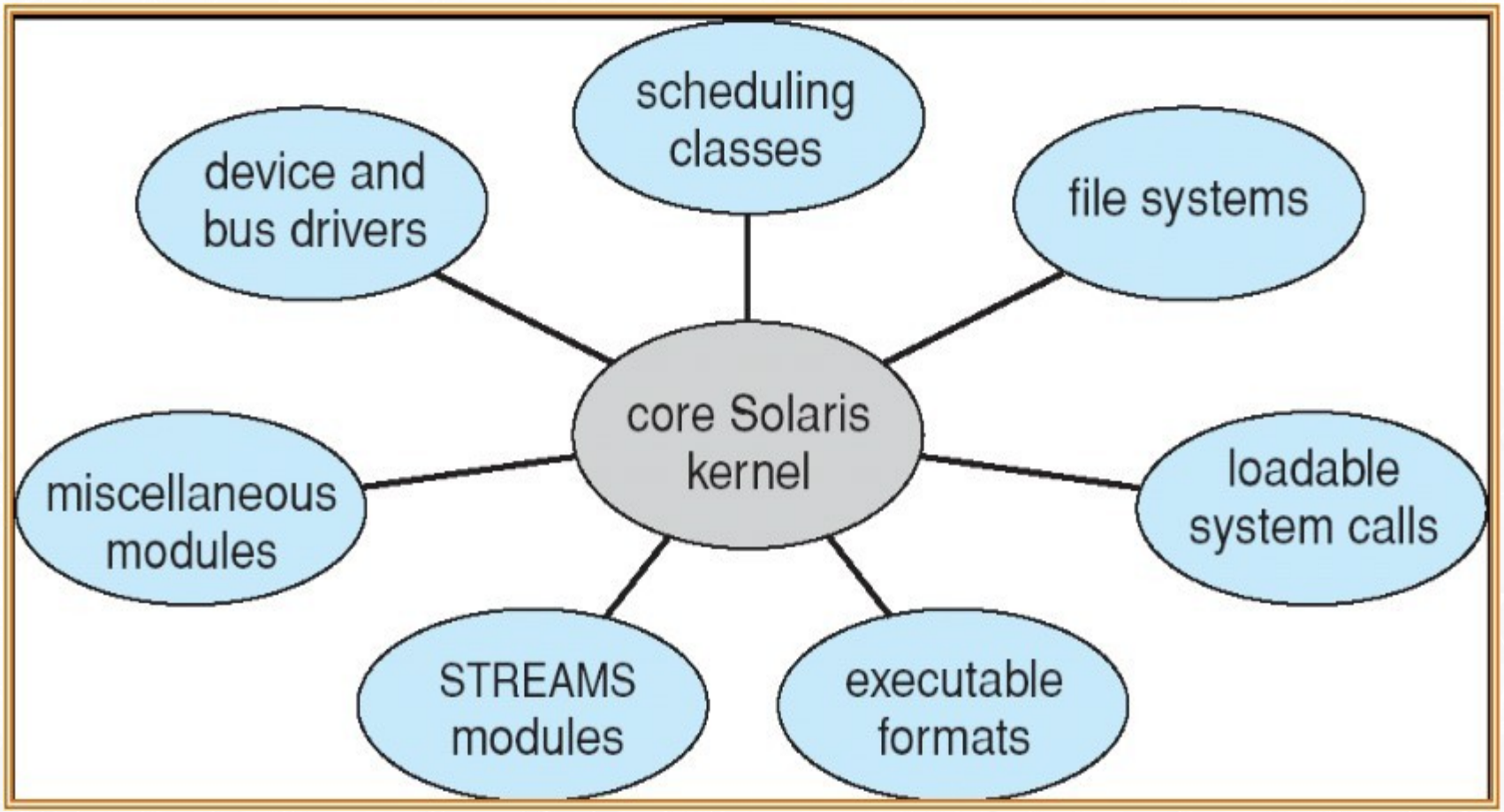
Arquiteturas de Sistemas Operacionais

- Microkernel + servidores
  - O kernel contém apenas serviços necessários à operação dos servidores
- Exokernel + bibliotecas
  - Recursos físicos (CPU, memória, cache) são exportados de modo seguro para serem tratados pelas aplicações
  - Bibliotecas implementam serviços típicos de SO
- Embutido na aplicação
  - Normalmente utilizados em sistemas com apenas uma aplicação
  - Apenas os serviços de SO necessários à aplicação são linkados no binário final

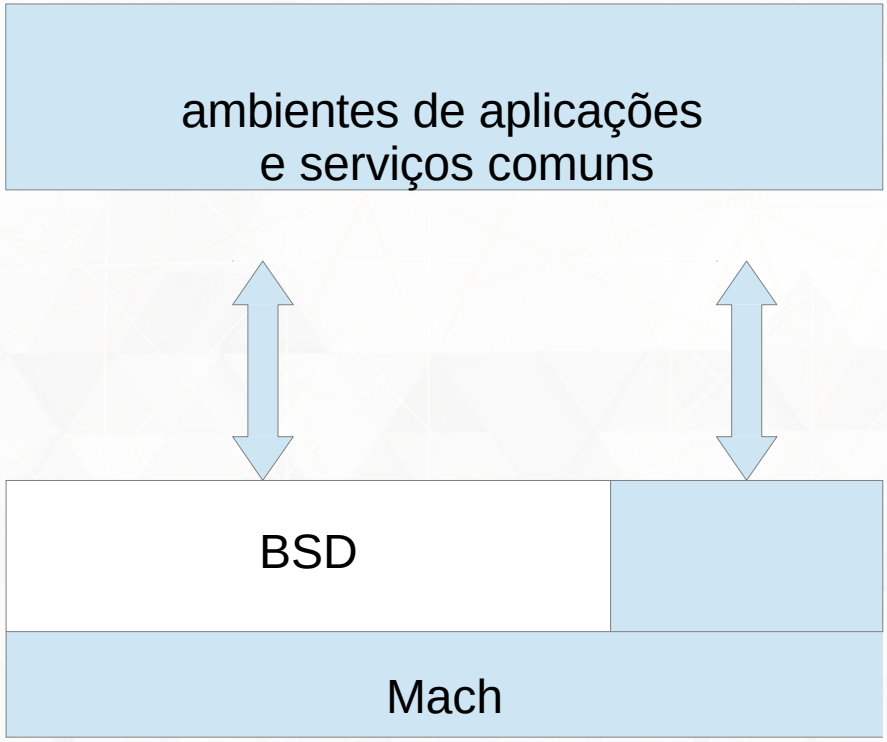
Engenharia de Sistemas Operacionais

- Estruturado
  - SO é decomposto em conjuntos de procedimentos/funções
  - Modificações implicam em recompilar todo o sistema
- Modular
  - SO é decomposto em conjuntos de módulos (ex.: subsistemas, classes de serviços, etc)
  - Permite “replugar” módulos
  - Originalmente implementado no Solaris
    - Presente nas versões modernas do Unix, Linux e Mac OS X
- Orientado a Objeto
  - Similar ao modular, mas utilizando técnicas mais eficientes de engenharia de software
- Baseados em componentes
  - SO é decomposto em conjuntos de componentes reusáveis (disponibilizando interfaces públicas apenas)

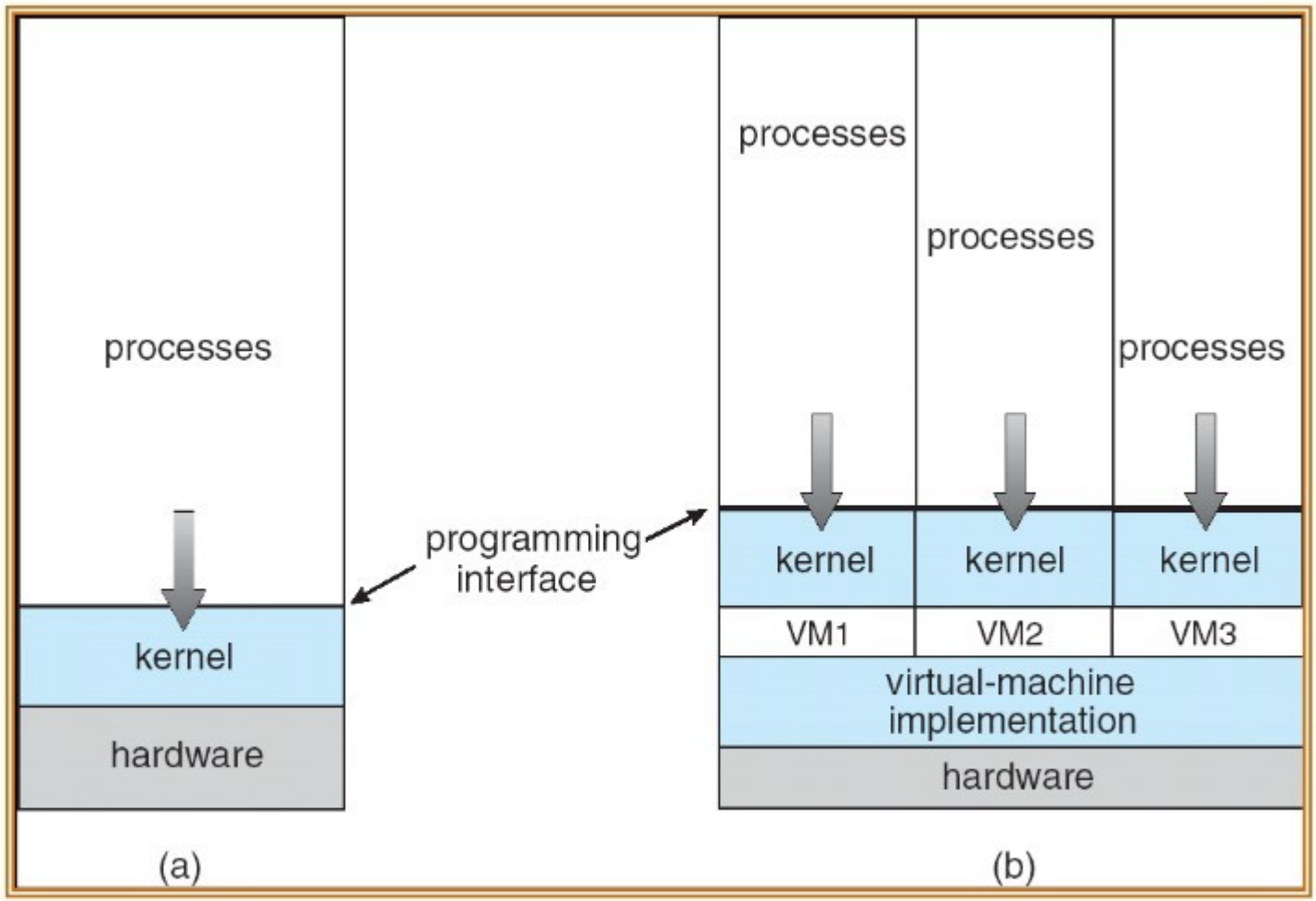
Estrutura Modular do Solaris



Mac OS X: estrutura híbrida
Modular + Microkernel



Máquinas Virtuais



Máquina não virtual

Máquina virtual