Chapter 2
Operating System Structures

OS Services

- User Interface
- Program Execution
- I/O Operation
- File System manipulation
- Communication
- Error detection

- Resource Allocation
- Accounting
- Protection/Security

User Interface to the OS

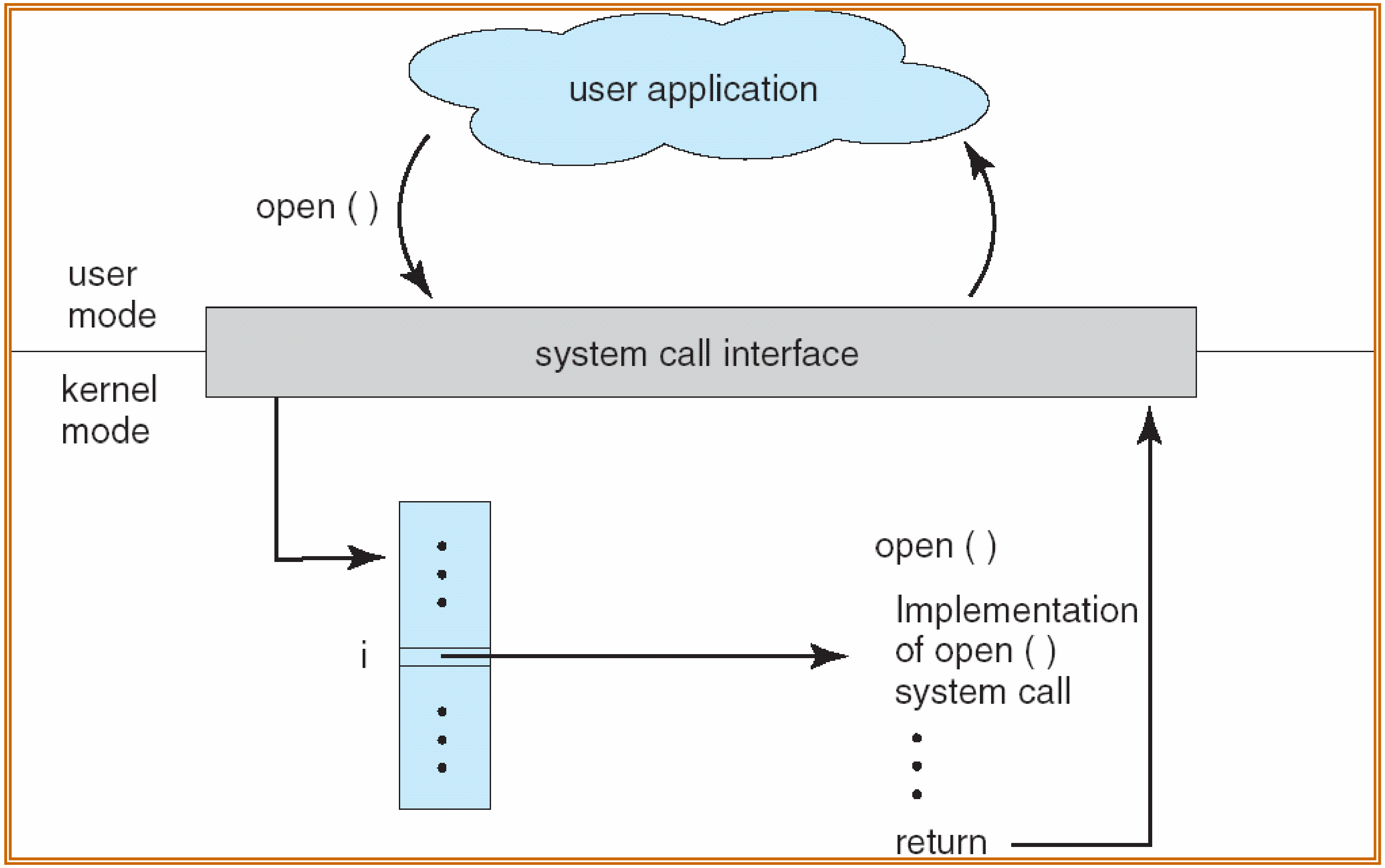
- Command Interpreter
  - Command line
  - Unix Shell
  - C:\
  - Mac Terminal
- GUI
  - Xerox PARC
  - Mac OS
  - Windows
  - X-Windows
  - KDE/GNOME/XFCE

Often these are application programs and not part of the OS.

The true interface to the OS is via system calls

System Calls

- Interface to OS (kernel) services
- Wrapped in API (API = ?)
  - POSIX
    libc.so
    libgcc.so
  - Win32
  - Java API
  - why?



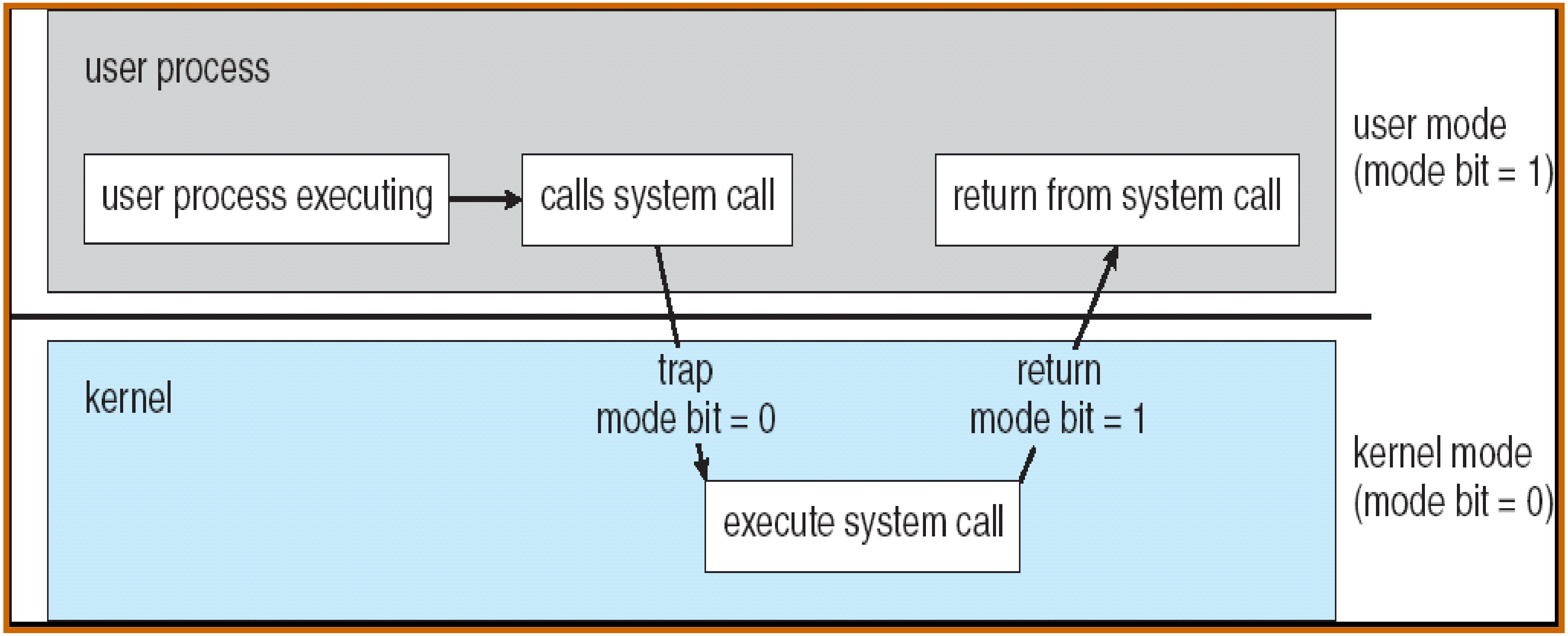
Operation

- Dual Mode
  - Kernel mode
    - { Supervisor | System | Privileged } mode
    - Hardware bit
    - Privileged instructions
      - Based on CPU type
      - I/O control
      - Interrupt management
      - Stop/Halt
      - Memory management
  - User mode
    - **System calls**

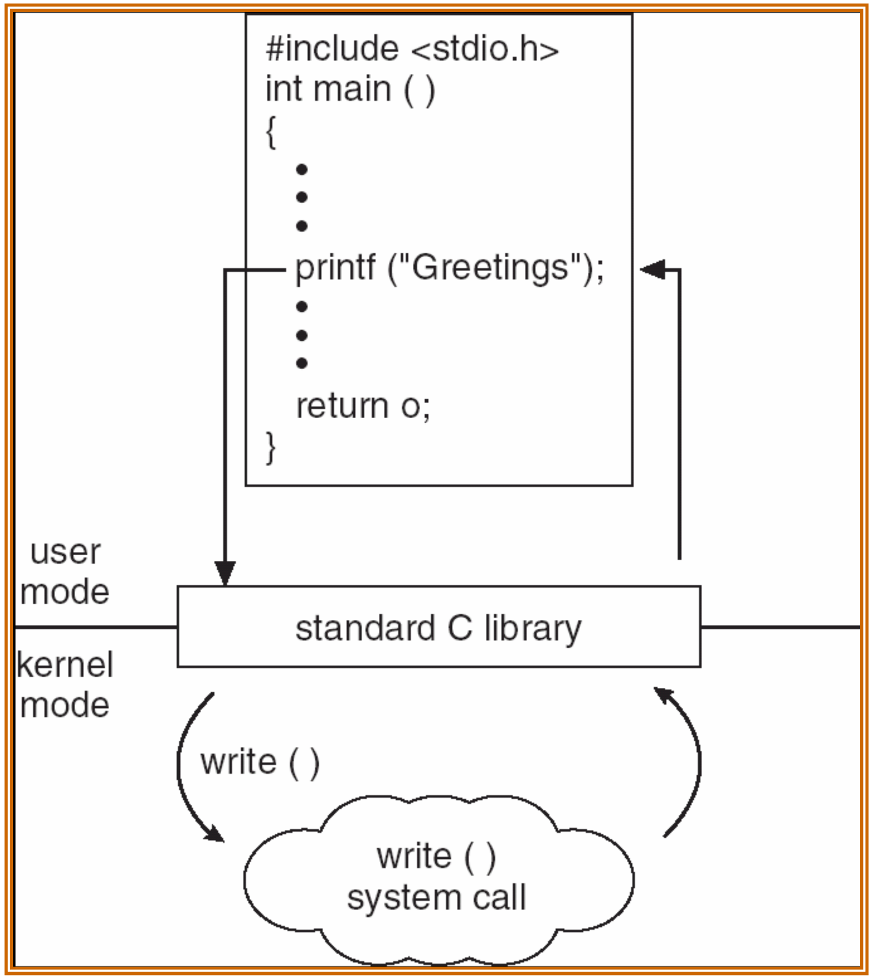
No mode bit on the original Intel 8088 chip

Hence, MS-DOS originally not dual mode!

Dual-Mode, in action



System Call via Library

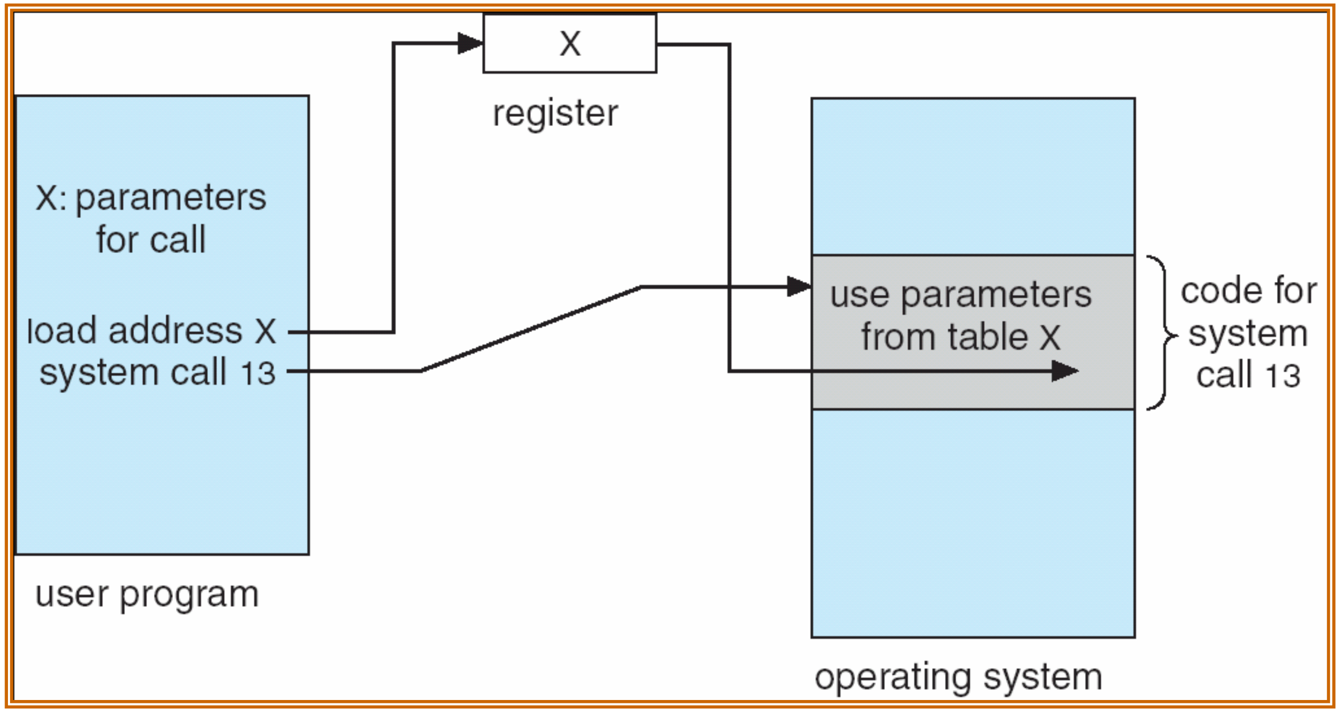


Linux
- glibc.so
- libpthread.so
- libcap.so

Solaris/SunOS
- libsocket.so

Systems Calls: Data

- Passing data to a system call
  - Registers
  - Block of memory
  - Stack
  - Advantages/ Disadvantages?



Types of System Calls

- Process Control
  - How does GDB work?
- File access
- Device access
- Information maintenance
- Communications

Process Control

- What are some process control system calls?
  - fork() / exec()

- How does GDB work?
  - the ptrace API
  - what does GDB need to do?

More System Calls....

- File Management

- Device Management
  - How is this different from File Management?
  - When would you use this?

Even More....

- Information Maintenance
  - Date
  - Time
- Communication
  - Message passing
    - pipes
  - Shared memory
  - Networking

“System Programs”

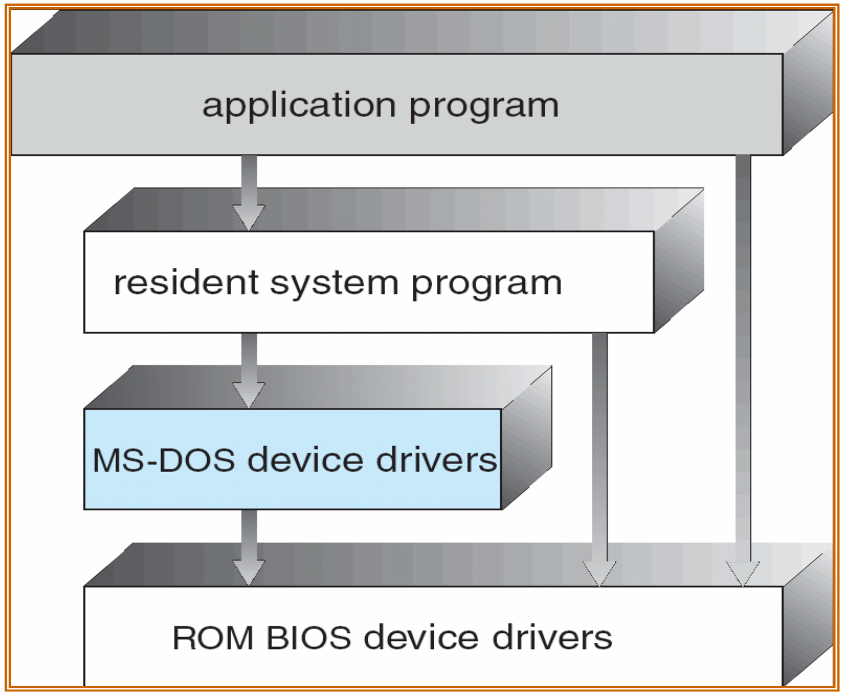
- System Utilities
  - common applications shipped with the OS/kernel
  - not necessarily part of the OS
  - often a wrapper around a system call
  - cp, mv, rm, cat
  - compilers
- Loaders

Operating System Design

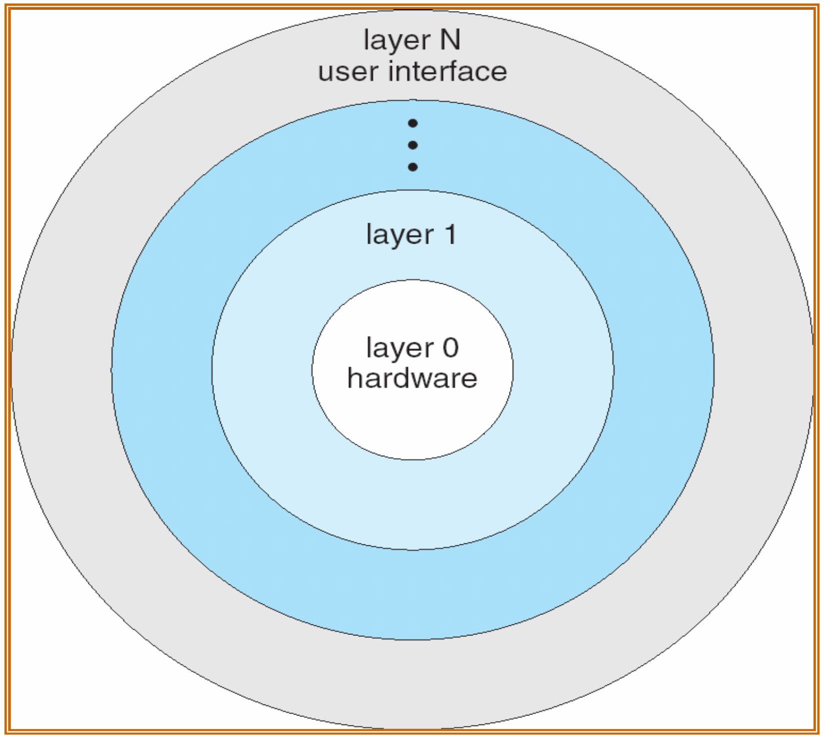
- Design Goals
- Mechanism vs Policies
- Implementations
  - Assembly vs C
    - advantages/disadvantages?

OS Structure

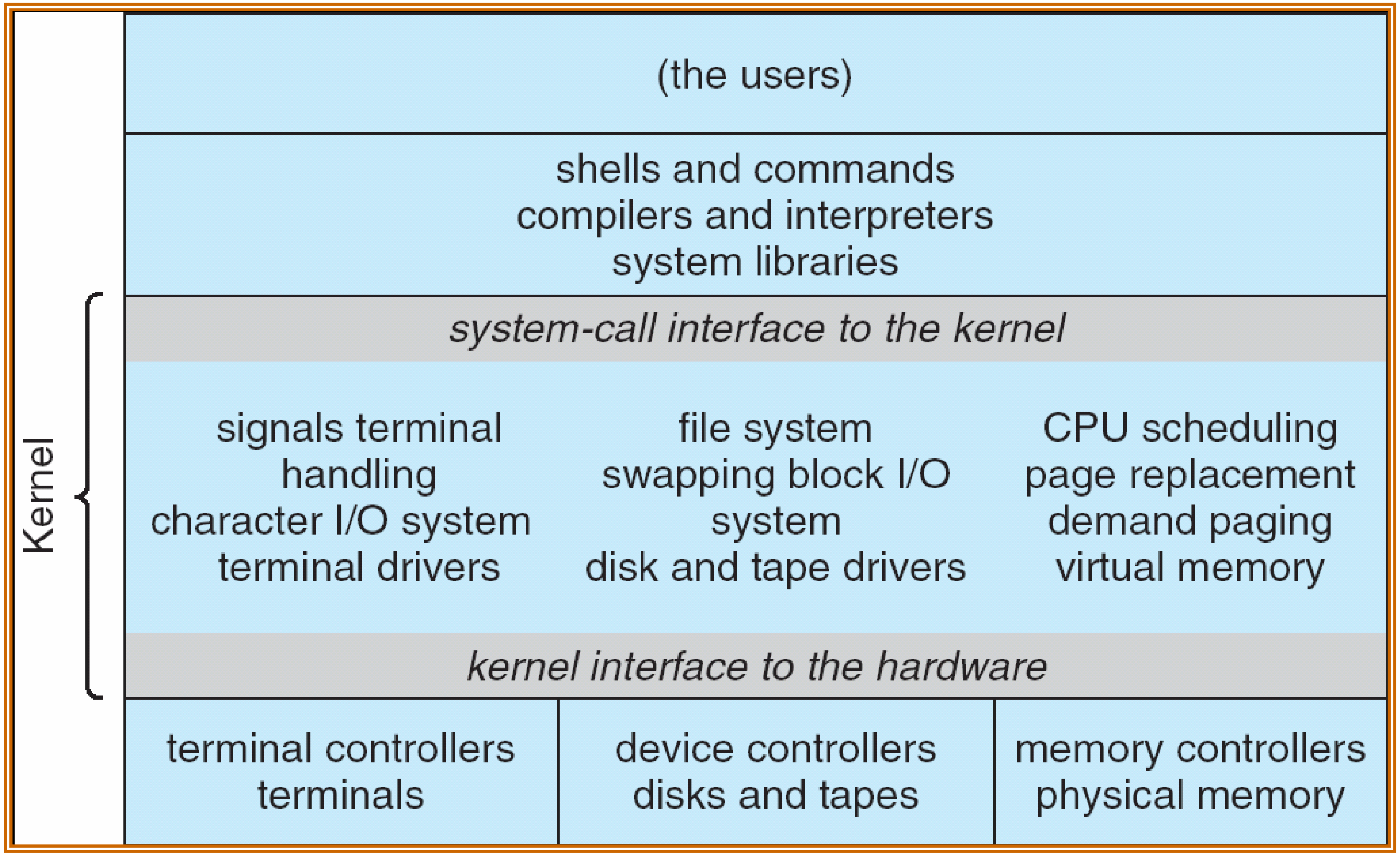
- Simple
  - MS DOS
  - Monolithic



- Layered



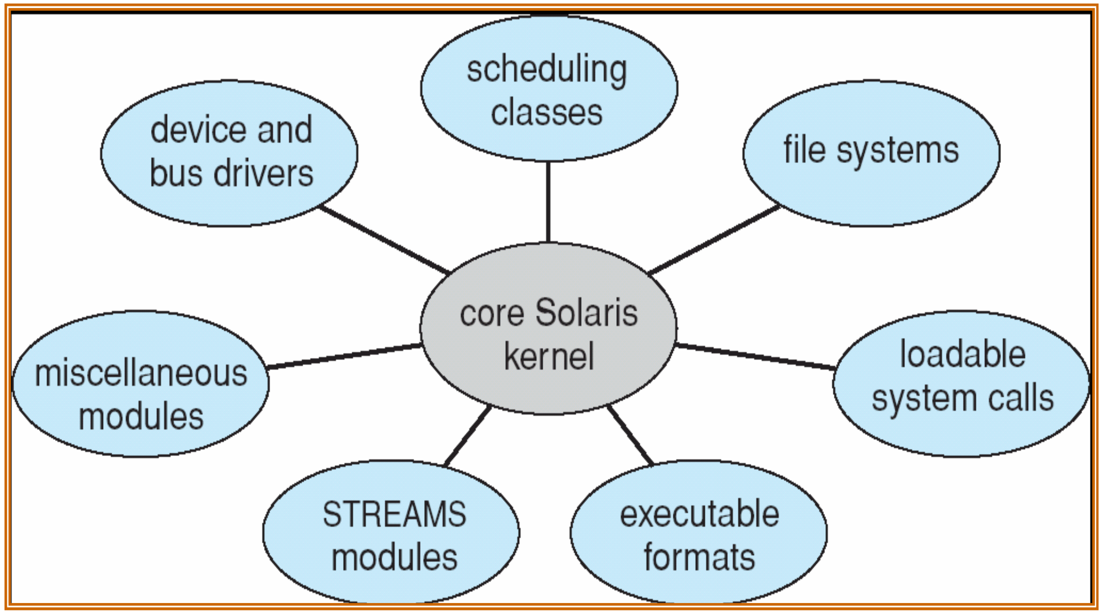
Old Unix



- Really Big Layers

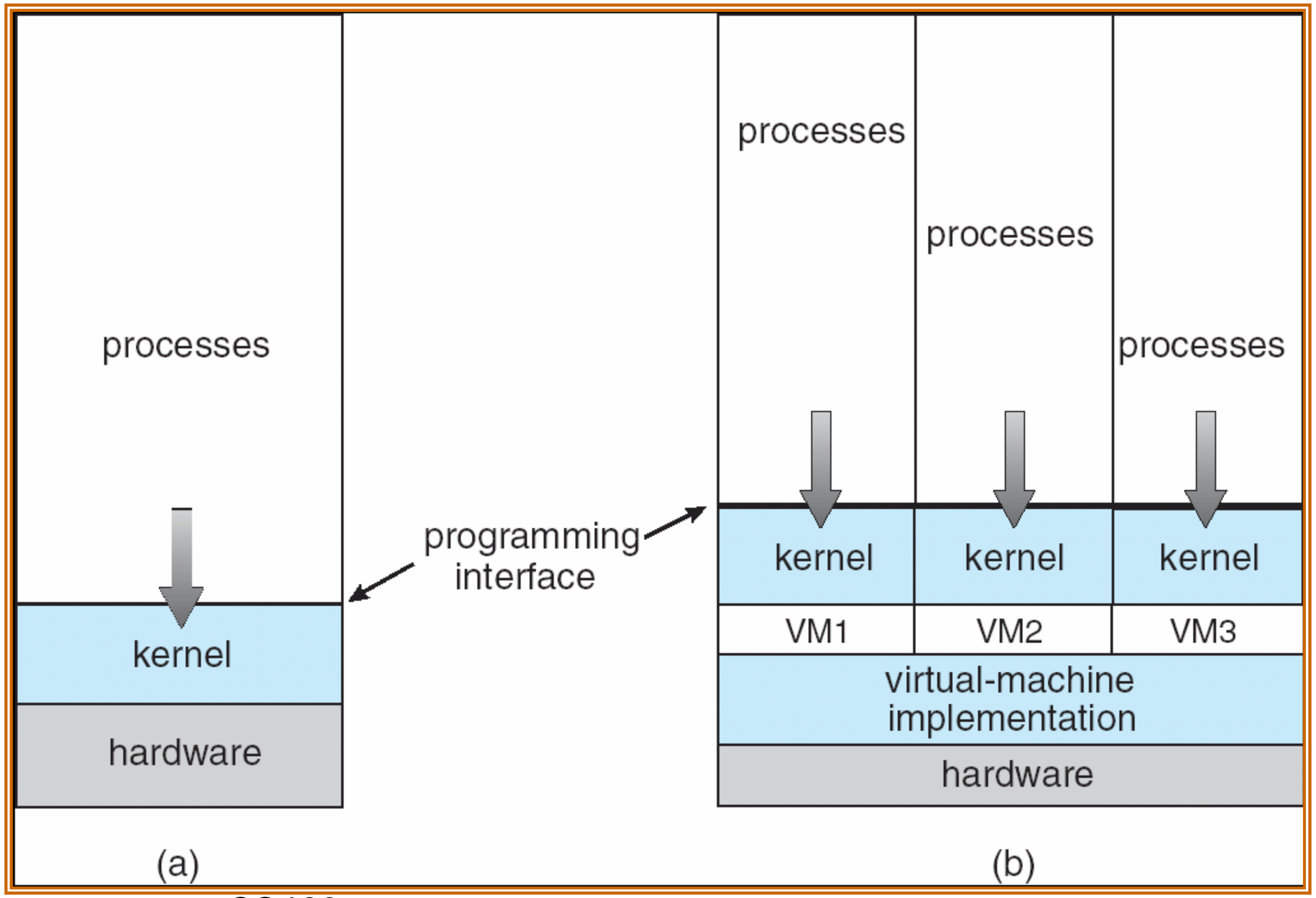
Structure

- Microkernel
  - Mach/MacOS
- Modular
- Monolithic with Modules
  - Modern Linux



Virtual Machines (VM)

- Abstract away the hardware
  - Real or imagined hardware
  - Parallels
  - VMWare/Bochs
  - VirtualBox
  - Java VM
  - .Net


processes
kernel
hardware
(a)
programming interface
processes
processes
processes
kernel
kernel
kernel
VM1
VM2
VM3
virtual-machine implementation
hardware
(b)