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

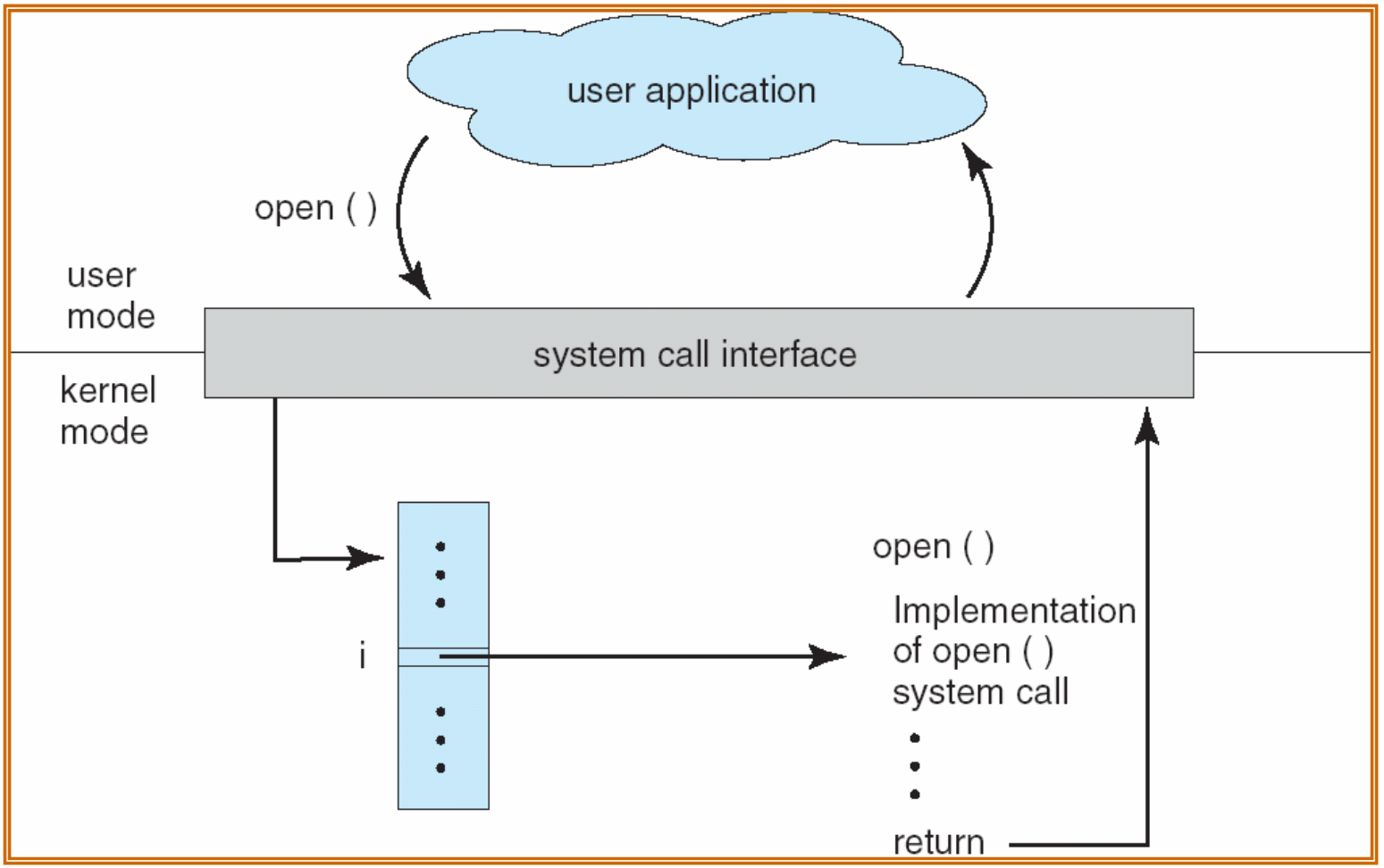
- Command Interpreter
  - Command line
  - Unix Shell
  - C:\
  - Mac Terminal

- GUI
  - Xerox PARC
  - Mac OS
  - Windows
  - X-Windows
  - KDE/GNOME

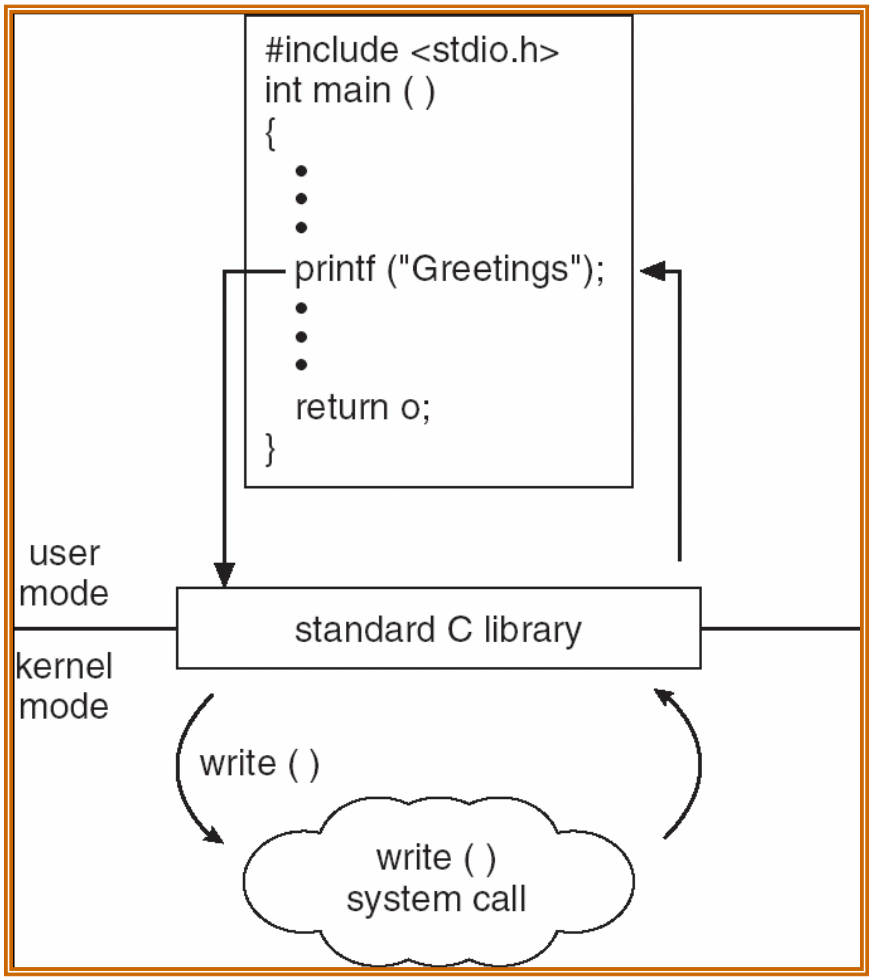
System Calls

- Interface to OS (kernel) services

- Wrapped in API (API = ?)
  - POSIX
    libc.so
    libgcc.so
  - Win32
  - Java API
  - why?

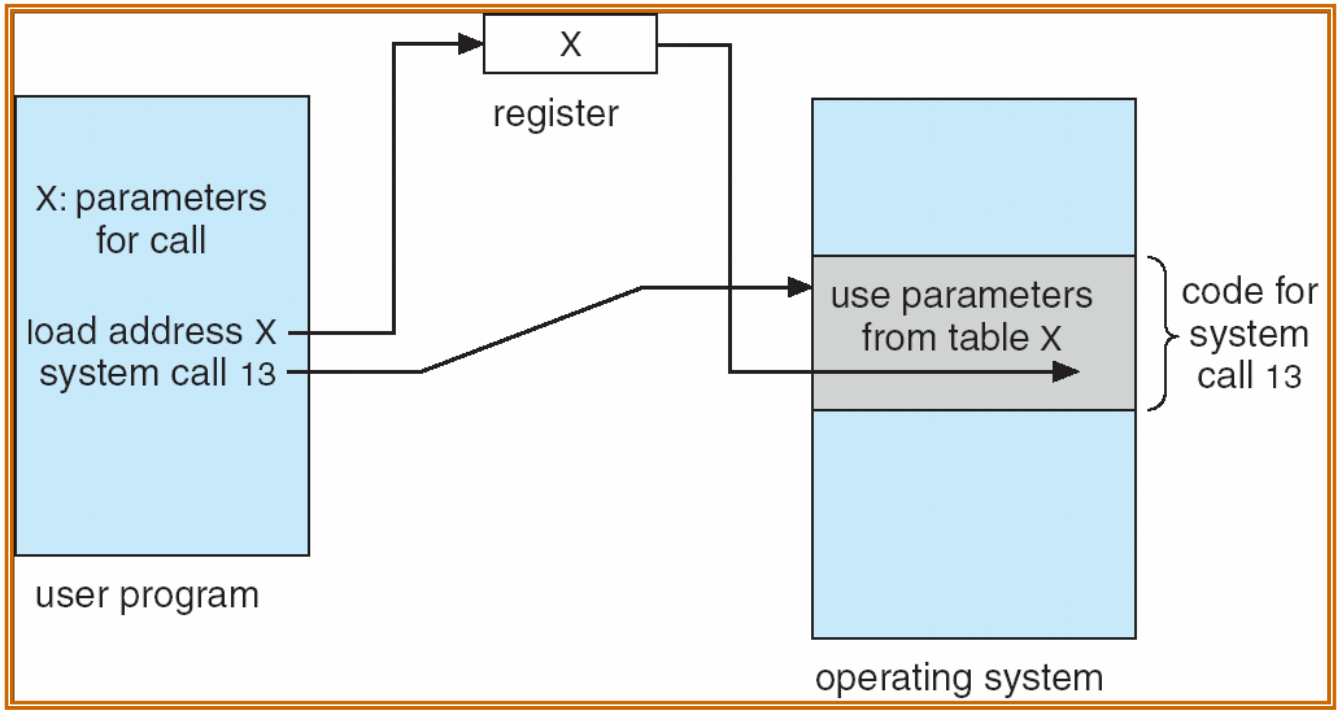


System Call via Library



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?
  -
  -
  -
  -
- 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....

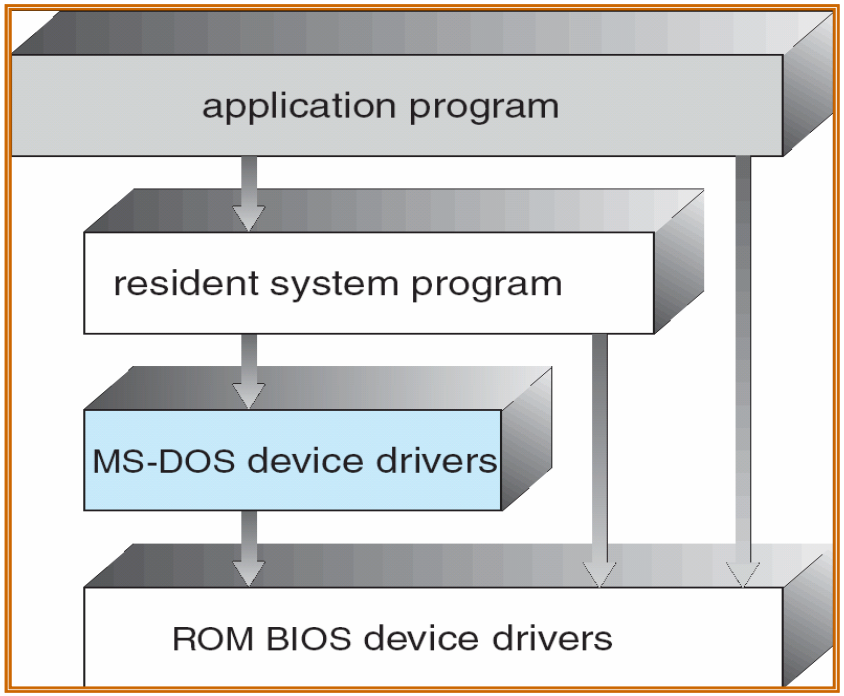
- Information Maintenance
  - Date
  - Time
- Communication
  - Message passing
    - pipes
  - Shared memory
  - Networking

Operating System Design

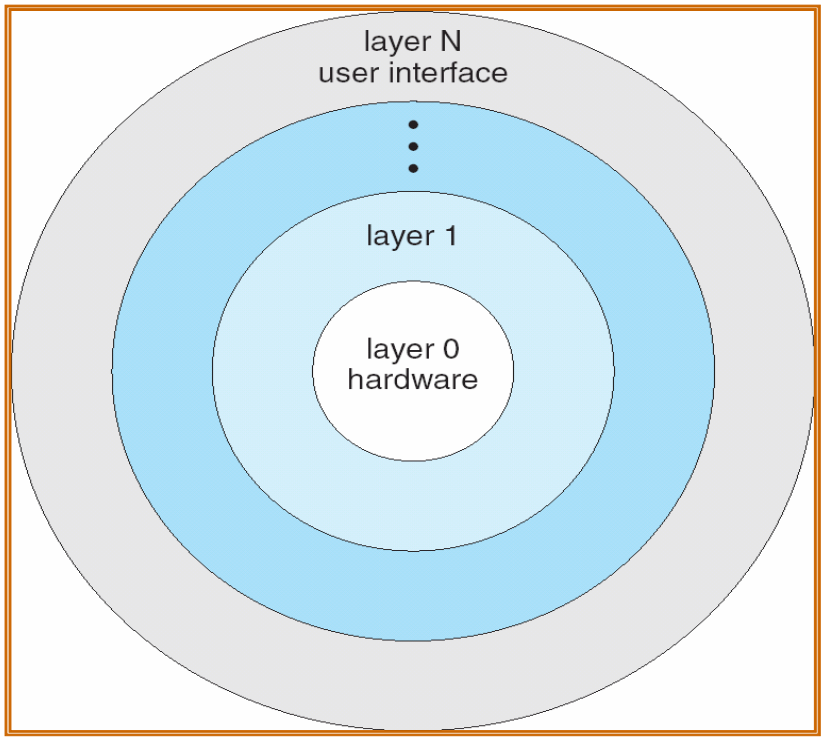
- Design Goals
- Mechanism vs Policies
- Implementations
  - Assembly vs C
    - advantages/disadvantages?

OS Structure

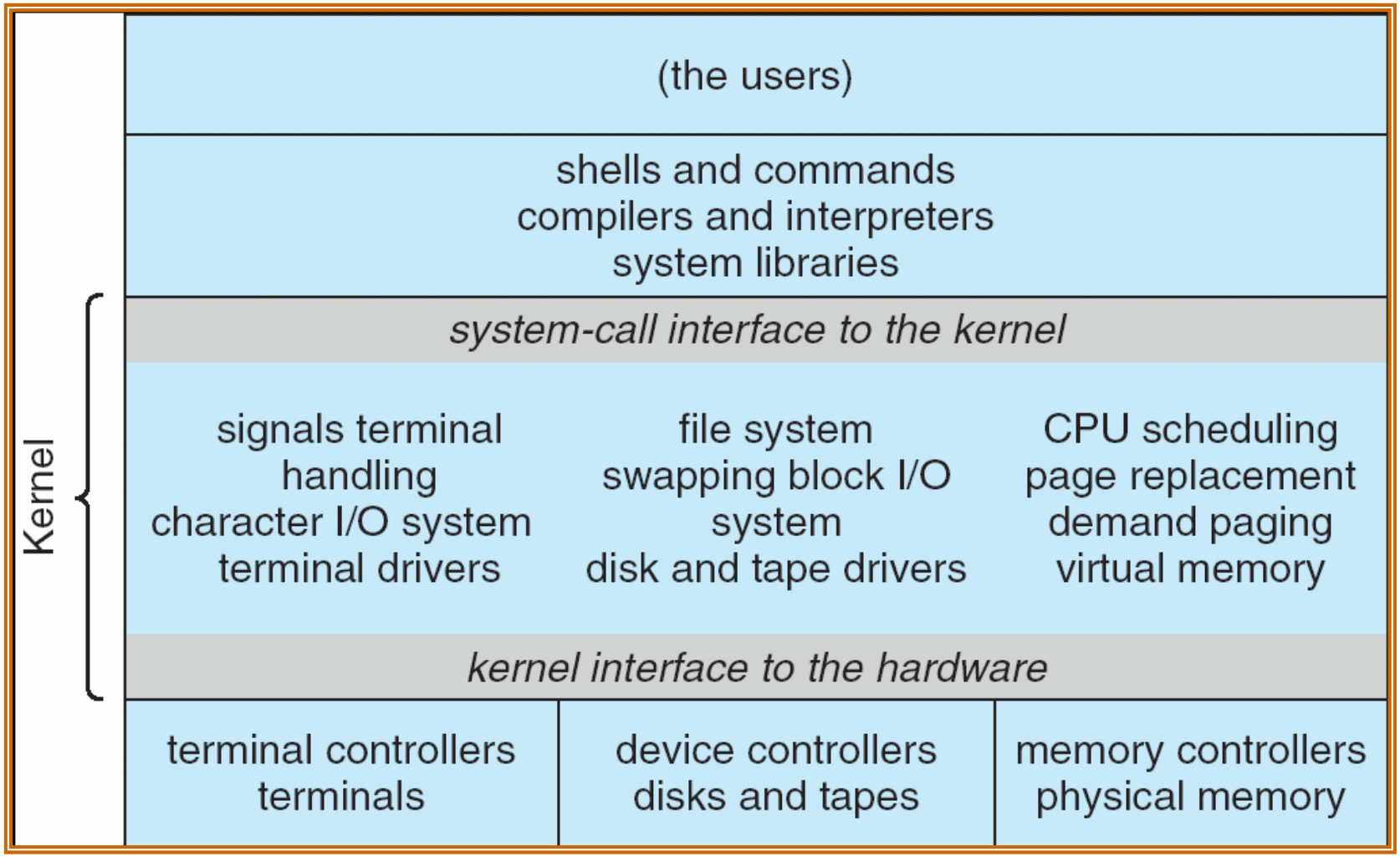
- Simple
  - MS DOS
  - Monolithic



- Layered



Old Unix

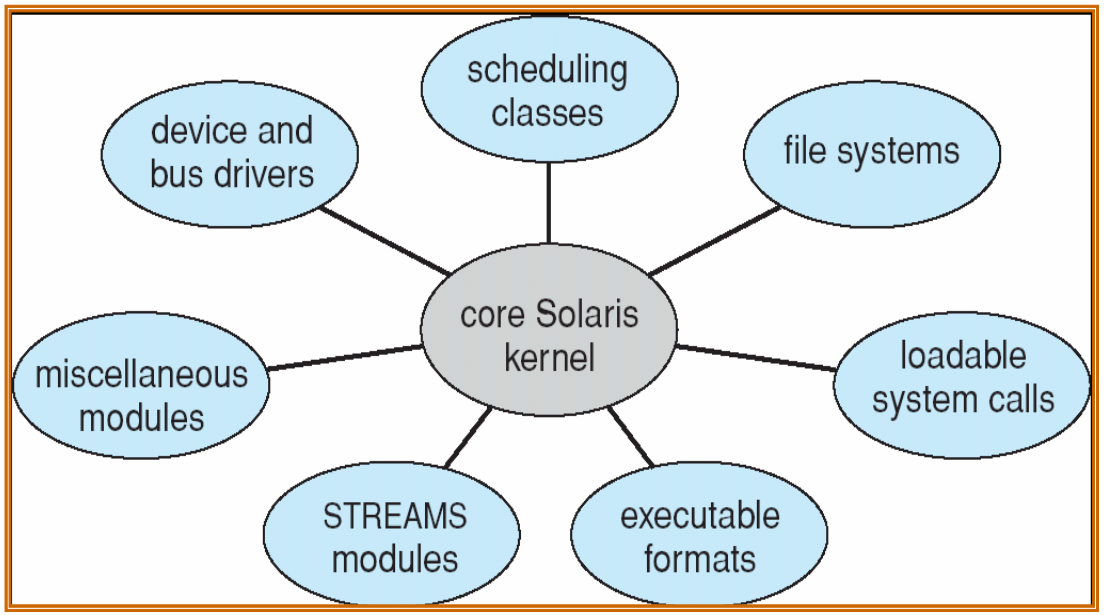


- Really Big Layers

Structure

- Microkernel
  - Mach/MacOS

- Modular
  - Modern Linux/Unix



Virtual Machines (VM)

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