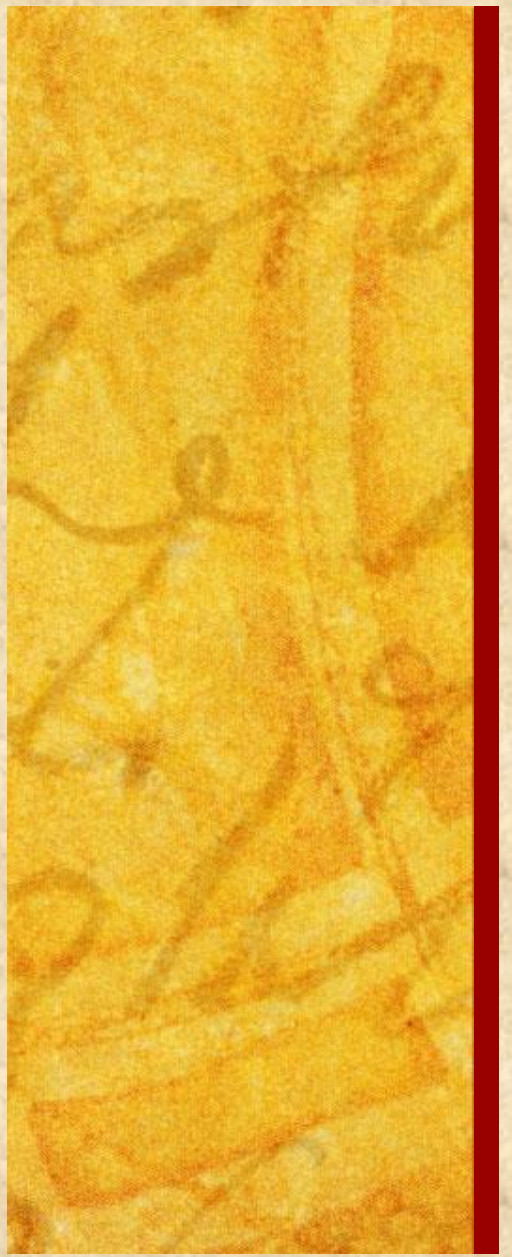


Chapter 3 Operating System Structures

Operating System Services

- User interface - Almost all operating systems have a user interface (UI)
- Program execution – The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error).
- I/O operations – Since user programs cannot execute I/O operations directly, the operating system must provide some means to perform I/O.
- File-system manipulation – Program capabilities to read and write files and directories, create and delete them, search them, list file Information, permission management.

Operating System Services

- Communications – Processes may exchange information, on the same computer or between computers over a network
  - Communications may be via shared memory or through message passing (packets moved by the OS)

- Error detection – Ensure correct computing by detecting errors in the CPU and memory hardware, in I/O devices, or in user programs.

Additional Operating System Functions

Additional functions exist not for helping the user, but rather for ensuring efficient system operations.

- Resource allocation – allocating resources to multiple users or multiple jobs running at the same time.
- Accounting – keep track of and record which users use how much and what kinds of computer resources for account billing or for accumulating usage statistics.
  - ✓ May include the amount of processor time and clock used, time limits, account numbers and so on.
- Protection and security - The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.

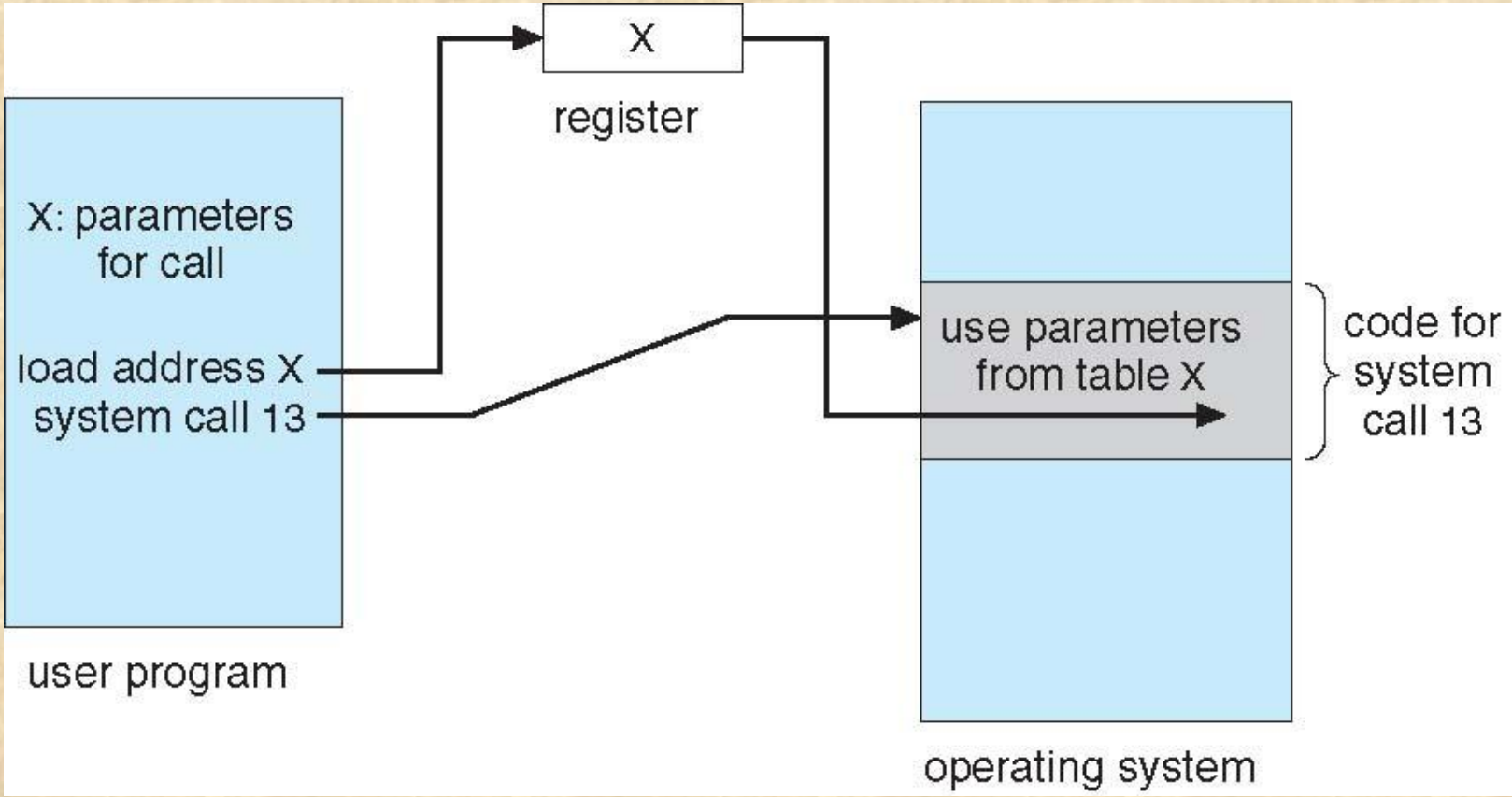
System Calls

- System calls provide the interface between a running program and the operating system.
  - Generally available as assembly-language instructions.
  - Languages defined to replace assembly language for systems programming allow system calls to be made directly (e.g., C, C++).

System Calls

- Three general methods are used to pass parameters between a running program and the operating system.
  - Pass the parameters in registers.
  - Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register.
  - Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system.

Parameter Passing via Table

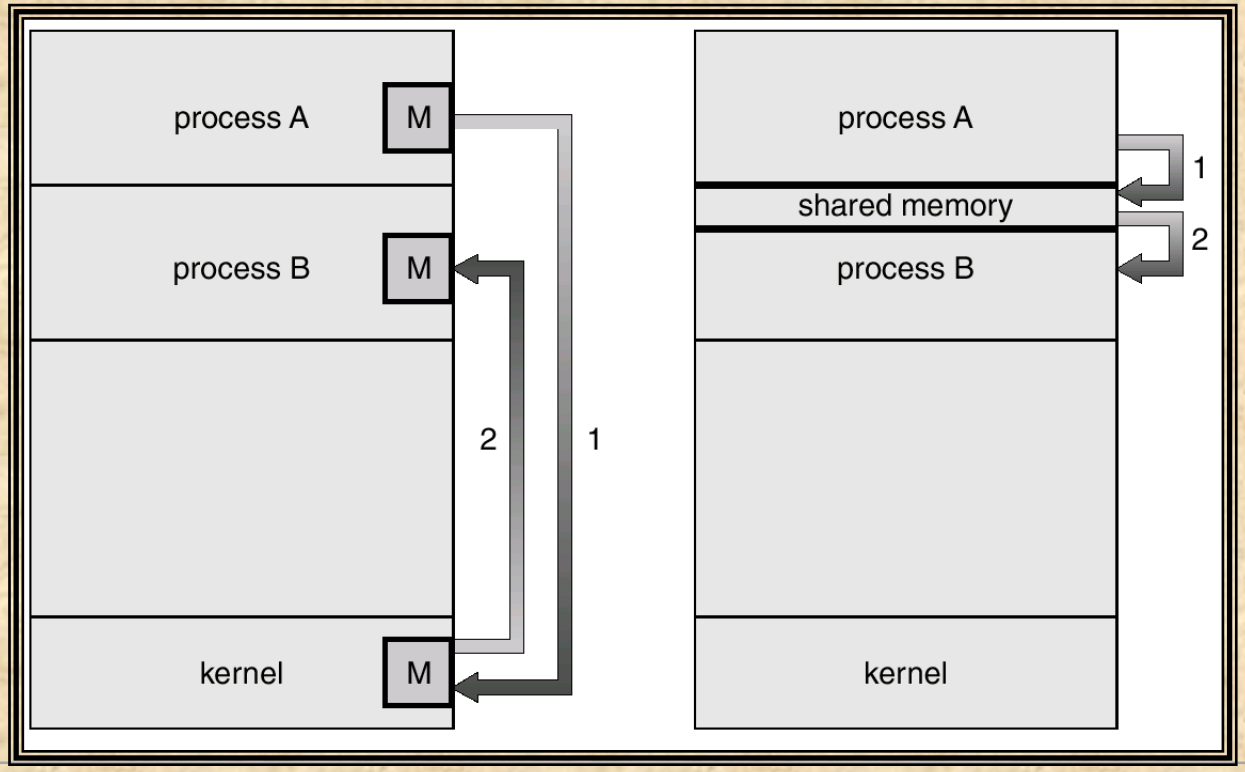


Types of System Calls

- Process control
- File management
- Device management
- Information maintenance
- Communications
- Protection

Communication Models

- Communication may take place using either message passing or shared memory.



Layered Approach

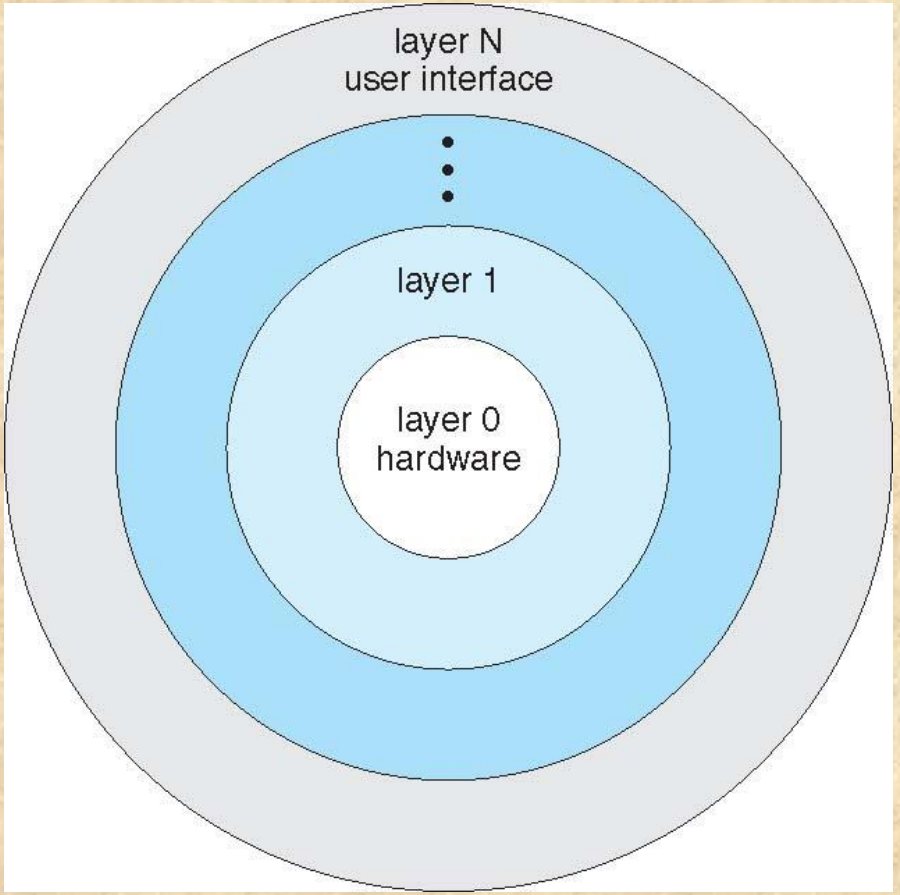
- In layered approach, the operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.

- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers.

Layered Approach

- Advantage
  - Simplicity of construction and debugging
- Disadvantages
  - The careful definition and interaction of the layers
  - Less efficient

Layered Operating System



Microkernel

- Small operating system core
- Contains only essential core operating systems functions
- Many services traditionally included in the operating system are now external subsystems
  - Device drivers
  - File systems
  - Virtual memory manager
  - Windowing system
  - Security services

Microkernel vs. Layered Kernel

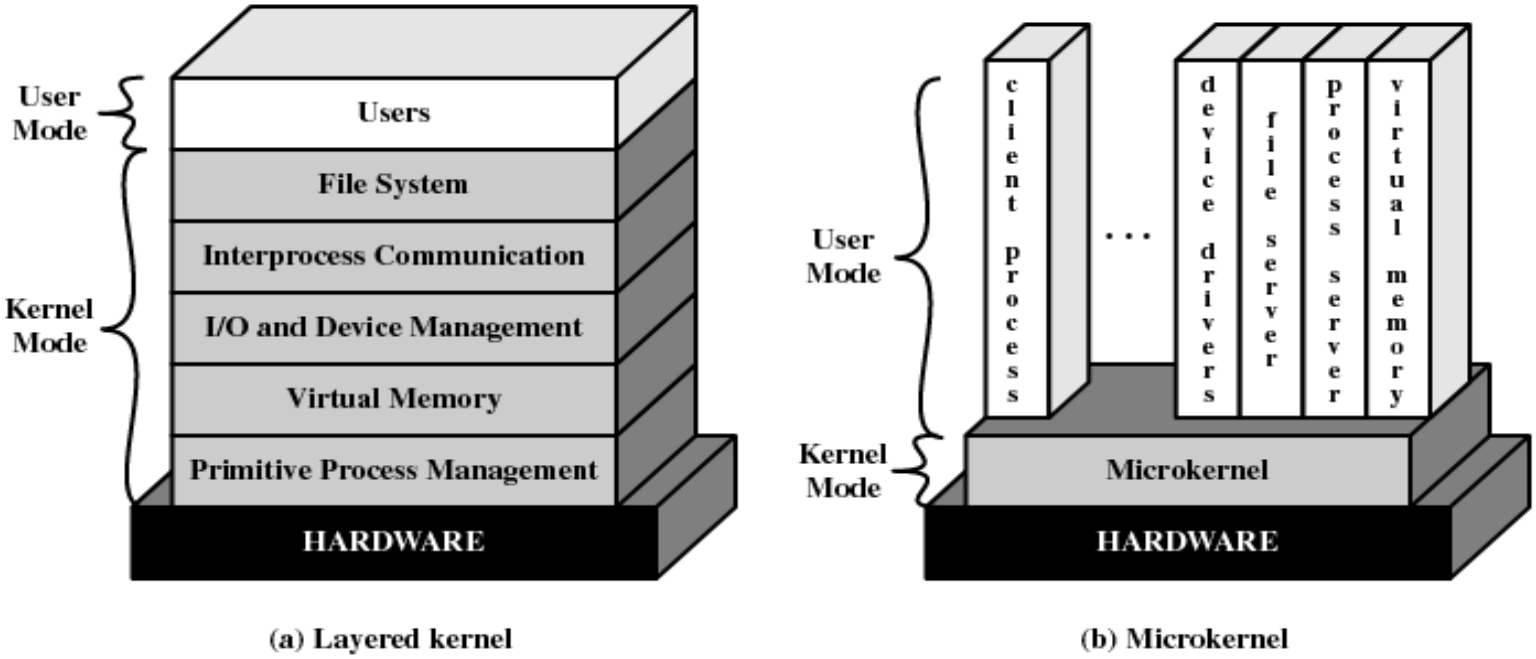


Figure 4.10 Kernel Architecture

Benefits of a Microkernel Organization

- Extensibility
  - Allows the addition of new services
- Flexibility
  - New features added
  - Existing features can be subtracted

Benefits of a Microkernel Organization

- Reliability
  - Modular design
  - Small microkernel can be rigorously tested
- Portability
  - Changes needed to port the system to a new processor is changed in the microkernel - not in the other services

Operating System Design

- Design and Implementation of OS not "solvable", but some approaches have proven successful.
- Start by defining goals and specifications
- The design of the system will be affected by the choice of hardware and the type of system: batch, time shared, single user, multiuser, distributed, real time or general purpose.

Operating System Design Goals

- User goals – operating system should be convenient to use, easy to learn, reliable, safe, and fast.

- System goals – operating system should be easy to design, implement, and maintain, as well as flexible, reliable, error-free, and efficient.

Summary

- Operating system services
- Additional operating system functions
- System calls
  - Types of system calls
- Layered approach
- Microkernel
  - Benefits of microkernel organization
- Operating system design