UNIT 1

INTRODUCTION

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware. I
- It manages the computer hardware and provides a basis for application program.
- The purpose of an operating systems is to provide an **environment** in which a user can execute programs in a ***convenient*** and ***efficient*** manner.
- The operating system must ensure the correct operation of the computer system.
- Operating system execute user programs and make solving user problems easier.
- Use the computer hardware in an efficient manner.

What is an operating system? Some definitions:

- A program that is executed by the processor that frequently relinquishes control and must depend on the processor to regain control.
- A program that mediates between application programs and the hardware
- A set of procedures that enable a group of people to use a computer system.
  A program that controls the execution of application programs
- An interface between applications and hardware

The common idea behind these definitions:

- Operating systems control and support the usage of computer systems.
- computer system = hardware + software
- OS is a part of the computer software, it is a program. It is a very special program, that is the first to be executed when the computer is switched on, and is supposed to control and support the execution of other programs and the overall usage of the computer system.

The operating system controls the usage of the computer resources - hardware devices and software utilities. We can think of an operating system as a *Resource Manager*. Here are some of the resources managed by the OS:

- Processors,
- Main memory,
- Secondary Memory,
- Peripheral devices,
- Information.

- The operating system provides a number of services to assist the users of the computer system:
- For the programmers:
- Utilities - debuggers, editors, file management, etc.
- For the end users - provides the interface to the application programs
- For programs - loads instructions and data into memory, prepares I/O devises for usage, handles interrupts and error conditions.
- Operating system relinquishes control of the processor.
- Functions same way as ordinary computer software
  - It is program that is executed

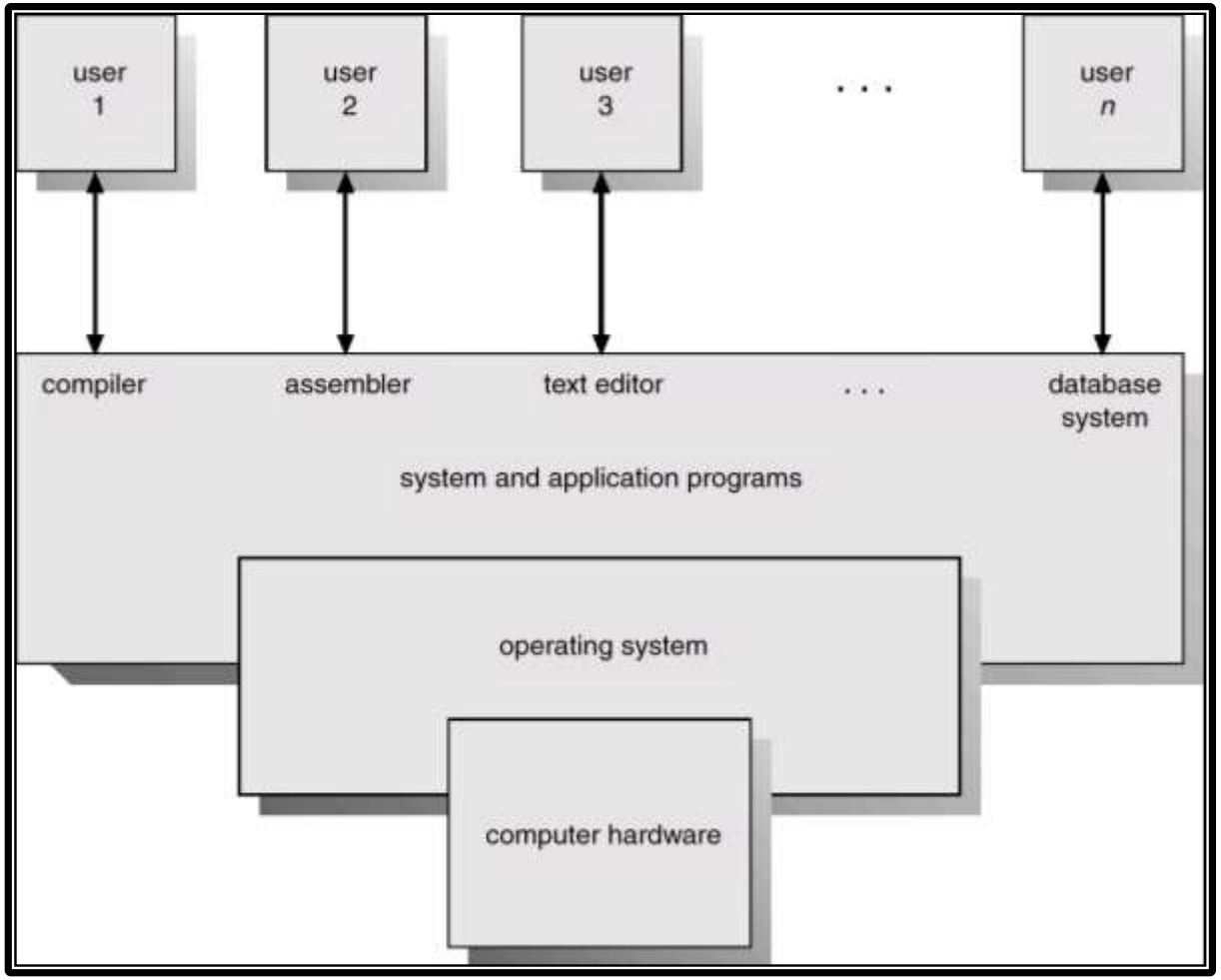
Main Objectives in OS design:

- Convenience
  - Makes the computer more convenient to use
- Efficiency
  - Allows computer system resources to be used in an efficient manner
- Ability to evolve
  - Permit effective development, testing, and introduction of new system functions without interfering with service

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

Abstract View of System Components



System View

- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operations of I/O devices .
- Kernel – the one program running at all times (all else being application programs).

  -Portion of operating system that is in main memory

  -Contains most frequently used functions

  -Also called the nucleus

Services Provided by the Operating System

- Program development
  - Editors and debuggers
- Program execution
- Access to I/O devices
- Controlled access to files
- System access

Services Provided by the Operating System

- Error detection and response
  - Internal and external hardware errors
    - Memory error
    - Device failure
  - Software errors
    - Arithmetic overflow
    - Access forbidden memory locations
  - Operating system cannot grant request of application

Services Provided by the Operating System

- Accounting
  - Collect usage statistics
  - Monitor performance
  - Used to anticipate future enhancements
  - Used for billing purposes

Early System

- Early computers were very large machines run from a console.The programmer would write a program and then operate the program directly from the operator’s console.
- First the program would be manually loaded into memory either from the front panel switches, paper tape or punched cards. Then the appropriate buttons would be pushed to load the starting address and to start the execution of the program.

- If errors are discovered the programmer could halt the program , examine the contents of the memory and register , and debug the program directly from the console.
- Output was printed.

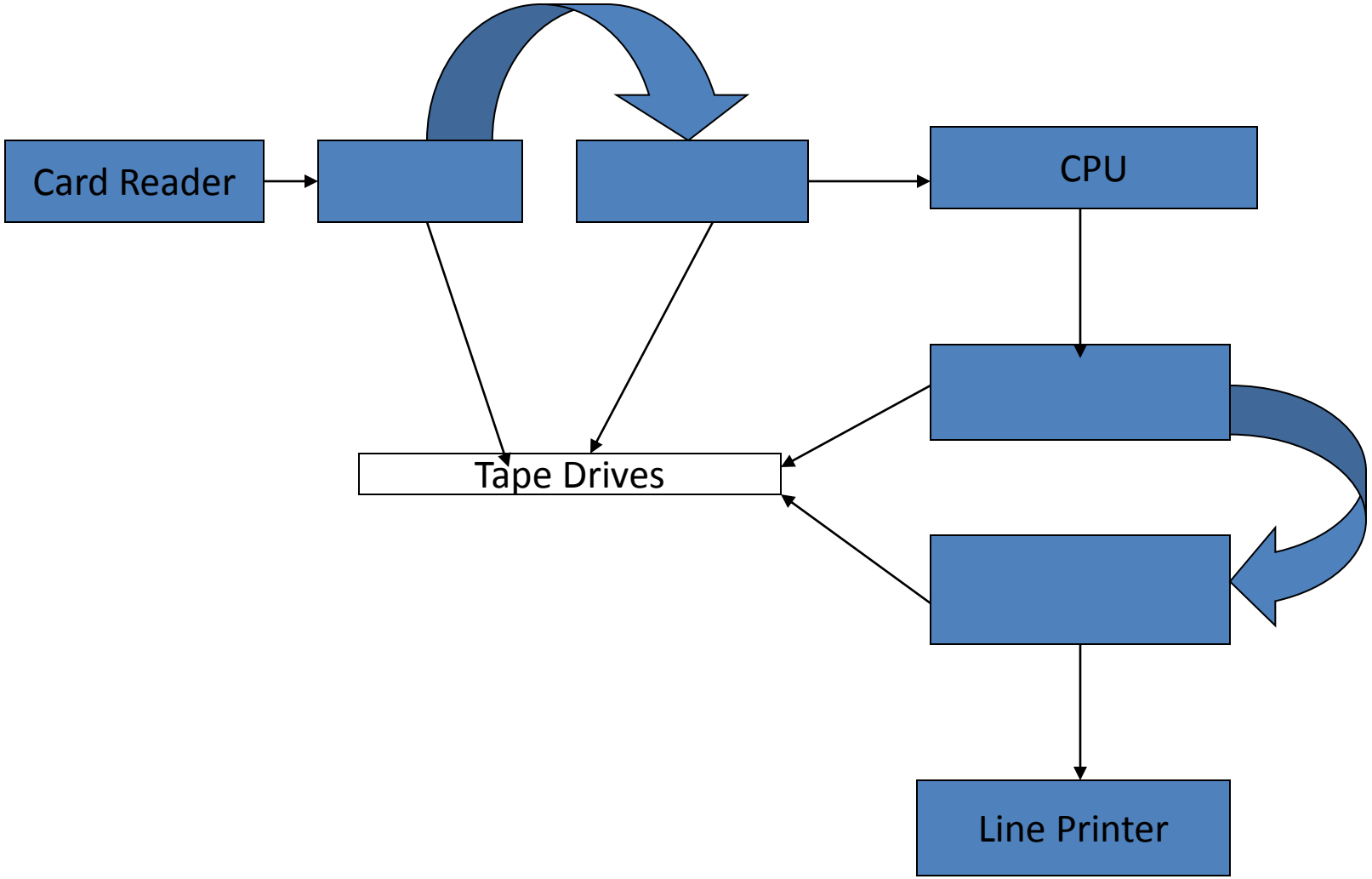
Performance

Off-Line Operation:

One common solution was to replace the very slow card readers and line printers with magnetic tape units. The majority of the computer system in the late 1950's and early 1960's were batch system reading from the card reader and writing to line printer. Rather than have the CPU read directly from cards, however, the cards were first copied on to a magnetic tape.

- When the tape was sufficiently full, it was taken down and carried over to the computer. When a card was needed for input to a program , it was read from the tape. Similarly , output was written to tape and the contents of the tape would be printed later. The card readers and line printers were operated off-line.
- Online- Operation

- OFF-Line Operation



Buffering

- Buffering attempts to keep both the CPU and I/O devices busy all the time. The idea is quite simple. After data has been read and the CPU is about to start operating on it, the input device is instructed to begin the next input immediately. The CPU and the input device are both busy. By time the CPU is ready for the next data item, the input device will have finished reading it. The CPU can then begin processing the newly read data, while the input device starts to read the following data.
- Similarly buffering can be done for output.

Spooling

In a disk system cards are read directly from the card reader on to the disk. The location of the card images is recorded in a table kept by the os. Each job is noted in the table as it is read in. When a job is executed, it request or card reader input and are satisfied reading from the disk. Similarly when the job rquests the printer to output a line, that line is copied in to the system buffer and written to the disk. When the job is completed, the output is actually printed. This form of processing is called spooling.

Types of Operating System

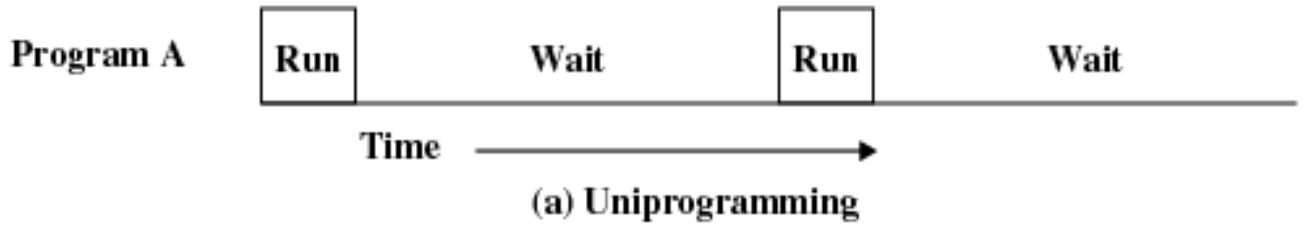
1. Batch Operating Systems

- Batch processing generally requires the program, data and appropriate system commands to be submitted together in the form of a job.
- Programs that do not require interaction and program with long execution times may be served well by a batch operating system. Ex :- payroll, forecasting, statistical analysis.
- Scheduling in batch system is very simple. Jobs are typically processed in the order of submission that is first-come first-served fashion.

- Memory Management in batch systems is also very simple. Memory is usually divided into two areas. One of them is permanently occupied by the resident portion of the operating system, and the other is used to load transient programs for execution. When the transient program terminates, a new program is loaded into the same area of memory.
- Batch systems often provide simple form of file management. Since access to files is also serial, little protection and no concurrency control of file access is required.

Uniprocessing

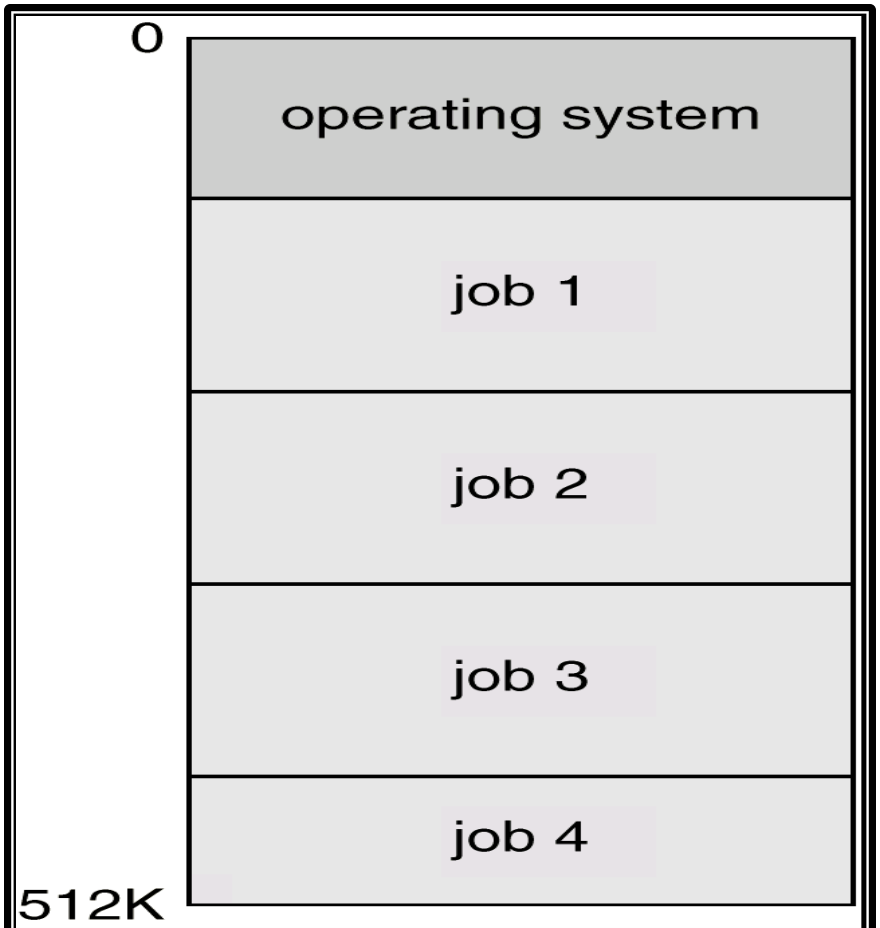
- Processor must wait for I/O instruction to complete before preceding



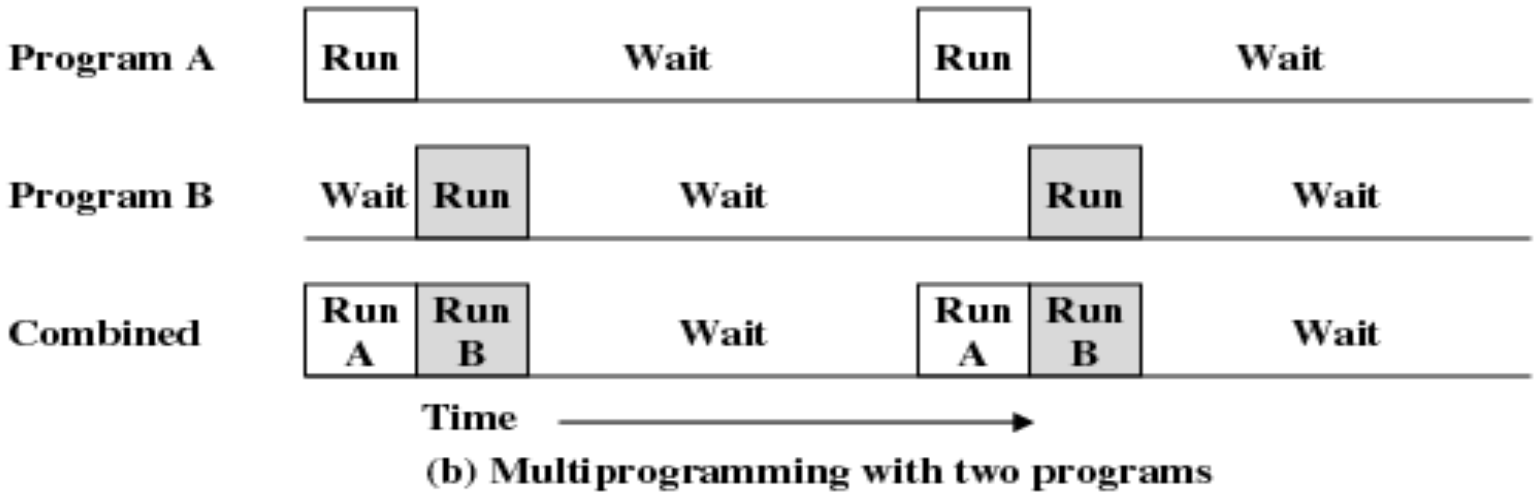
(a) Uniprogramming

Multiprogramming

- Several jobs are kept in main memory at the same time, and the
- CPU is multiplexed among them.

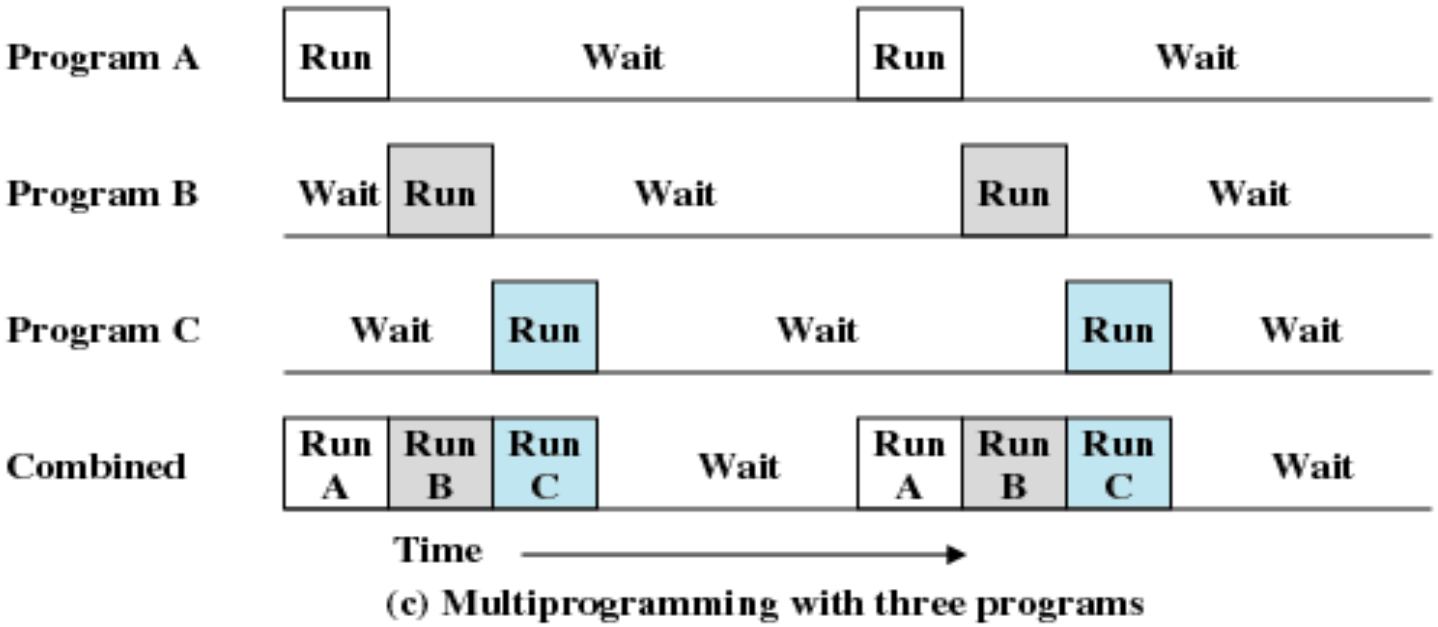


When one job needs to wait for I/O, the processor can switch to the other job.



(b) Multiprogramming with two programs

Multiprogramming



(c) Multiprogramming with three programs

OS Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of devices.
- Multiprogrammed system provide an environment where the various system resources can be utilized effectively.

Time-Sharing Systems–Interactive Computing

- Time sharing is a logical extension of multiprogramming.
- The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- A time-shared OS allows many users to share the computer simultaneously.
- Direct communication between the user and the system is provided.
- On-line system must be available for users to access data and code.

Real-Time Systems

- Real time operating systems are used in real time applications.
- A **Real-Time Operating System** (RTOS) is a computing environment that reacts to input within a specific time period.
- An operating system is considered real-time if it always allow its programs to perform tasks within specific time constraints, usually those expected by the user
- A real time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data.
- Processing must be done within the defined constraints or the system will fail.

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.

Real-Time Systems (Cont.)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

- To meet definition of RTOS, some or all of the following methods are employed:
- The RTOS performs few tasks, thus ensuring that the tasks will always be executed before the deadline
- The RTOS drops or reduces certain functions when they cannot be executed within the time constraints ("load shedding")
- The RTOS monitors input consistently and in a timely manner
- The RTOS monitors resources and can interrupt background processes as needed to ensure real-time execution
- The RTOS keeps track of how much of each resource (CPU time per timeslice, RAM, communications bandwidth, etc.) might possibly be used in the worst-case by the currently-running tasks, and refuses to accept a new task unless it "fits" in the remaining un-allocated resources.

Distributed Operating System

- A distributed computer system is a collection of autonomous computer systems capable of communication and cooperation via their hardware and software interconnections.
- Distributed computer systems evolved from computer networks in which a number of largely independent hosts are connected by communication links and protocols.
- Distributed OS usually provide the means for system-wide sharing of resources, such as computational capacity, files, and I/O devices.
- A distributed OS may facilitate access to remote resources, communication with remote processes and distribution of computations.

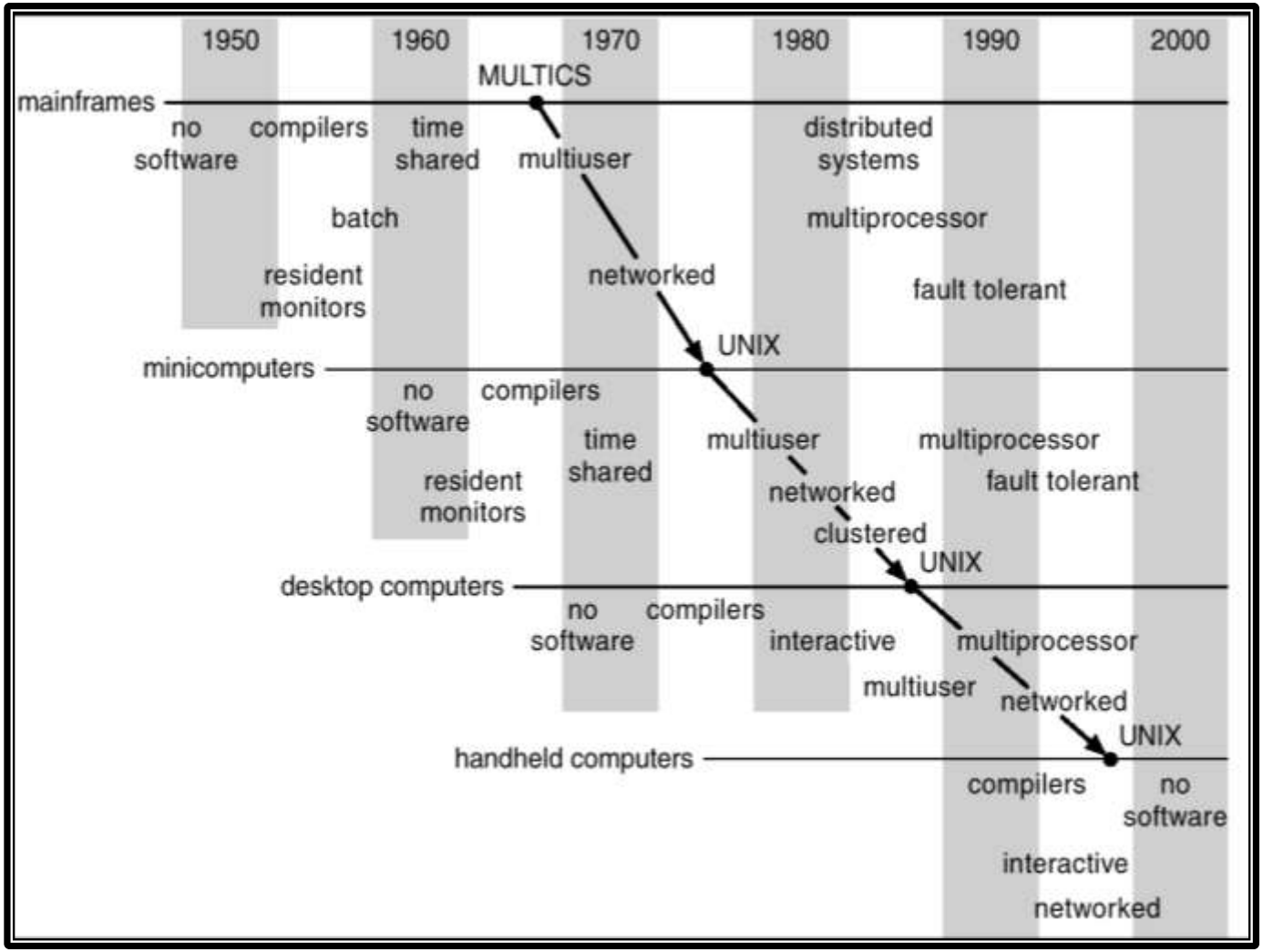
Distributed Operating System(cont..)

- Advantages:
  - resource sharing
  - computation speed-up
  - reliability
  - communication - e.g. email
- Applications - digital libraries, digital multimedia

I/O Protection

- All I/O instructions are privileged instructions.
- Must ensure that a user program could never gain control of the computer in monitor mode (I.e., a user program that, as part of its execution, stores a new address in the interrupt vector).

Migration of Operating-System Concepts and Features

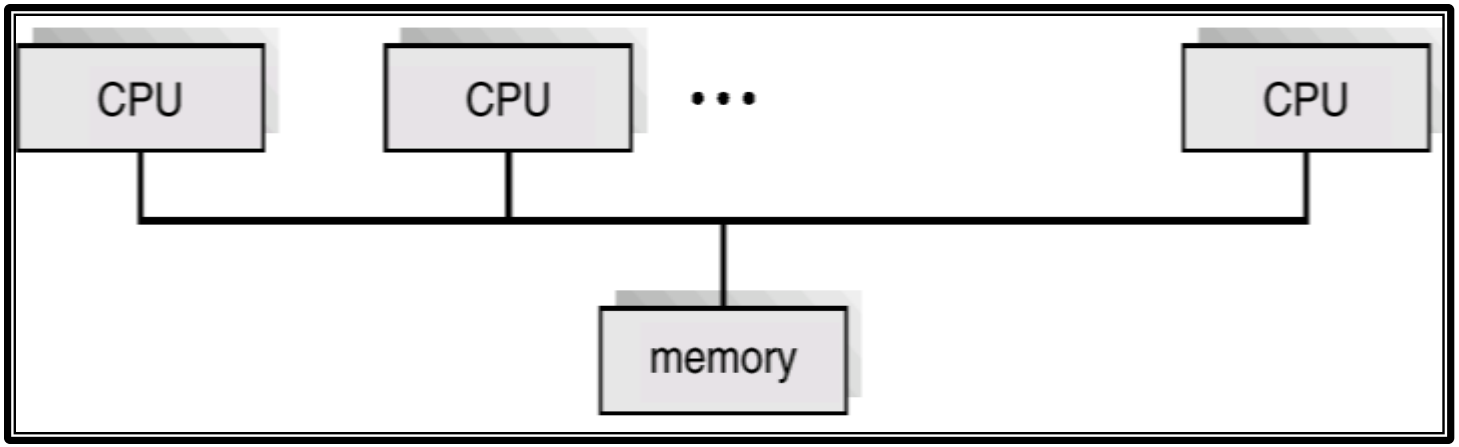


Multiprocessor System

- Multiprocessor systems with more than on CPU in close communication.
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems

- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
  - More common in extremely large systems

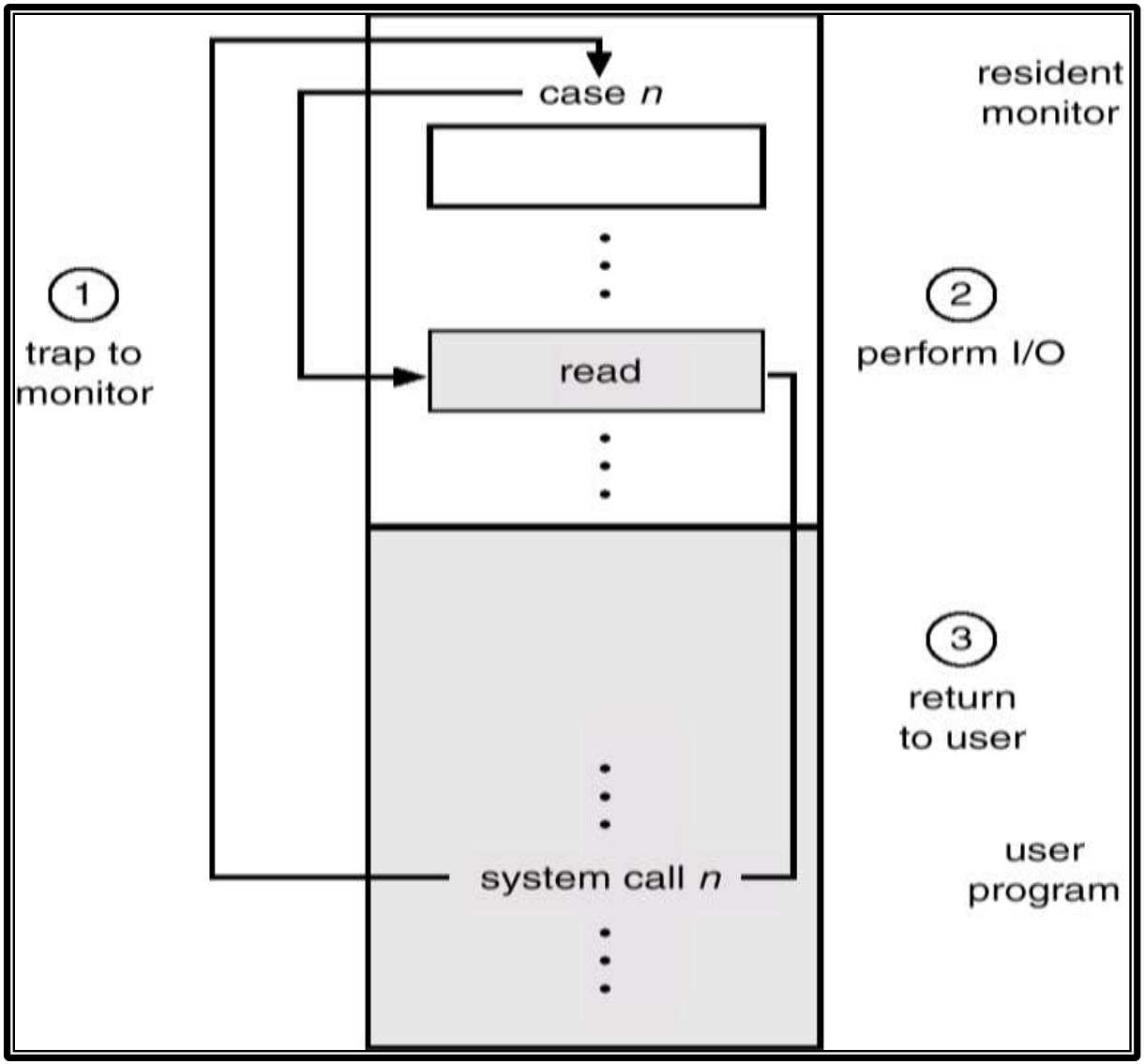
Symmetric Multiprocessing Architecture



I/O Protection

- All I/O instructions are privileged instructions.
- Must ensure that a user program could never gain control of the computer in monitor mode (I.e., a user program that, as part of its execution, stores a new address in the interrupt vector).

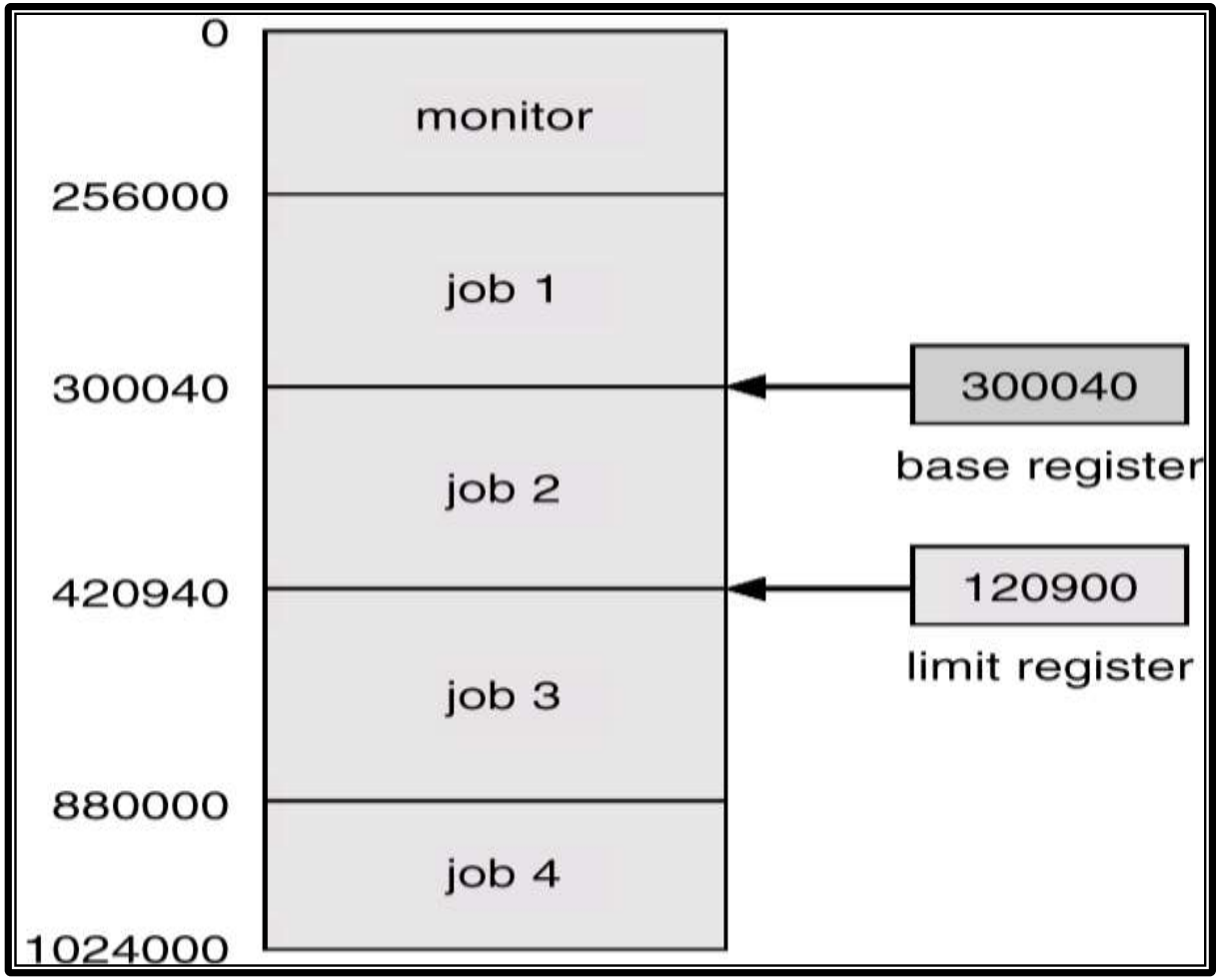
Use of A System Call to Perform I/O



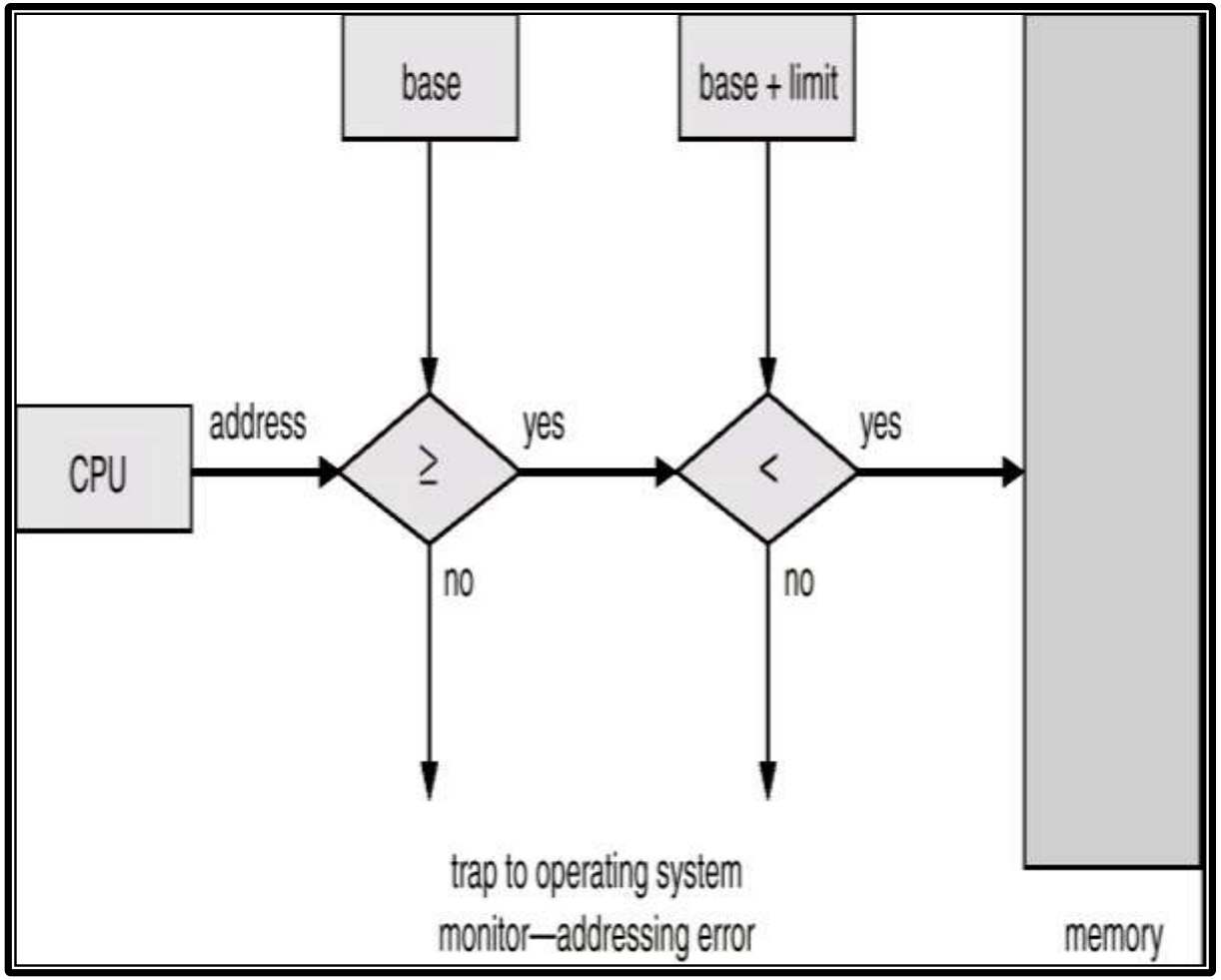
Memory Protection

- Must provide memory protection at least for the interrupt vector and the interrupt service routines.
- In order to have memory protection, add two registers that determine the range of legal addresses a program may access:
  - **Base register** – **holds the smallest legal physical memory address.**
  - **Limit register** – **contains the size of the range**
- Memory outside the defined range is protected.

Use of A Base and Limit Register



Hardware Address Protection



CPU Protection

- *Timer* – interrupts computer after specified period to ensure operating system maintains control.
  - **Timer is decremented every clock tick.**
  - **When timer reaches the value 0, an interrupt occurs.**
- Timer commonly used to implement time sharing.
- Time also used to compute the current time.
- Load-timer is a privileged instruction.

Operating System Services

- Program execution – system capability to load a program into memory and to run it.
- 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 capability to read, write, create, and delete files.
- Communications – exchange of information between processes executing either on the same computer or on different systems tied together by a network. Implemented via *shared memory* or *message passing*.
- 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.
- Protection – ensuring that all access to system resources is controlled.

User View

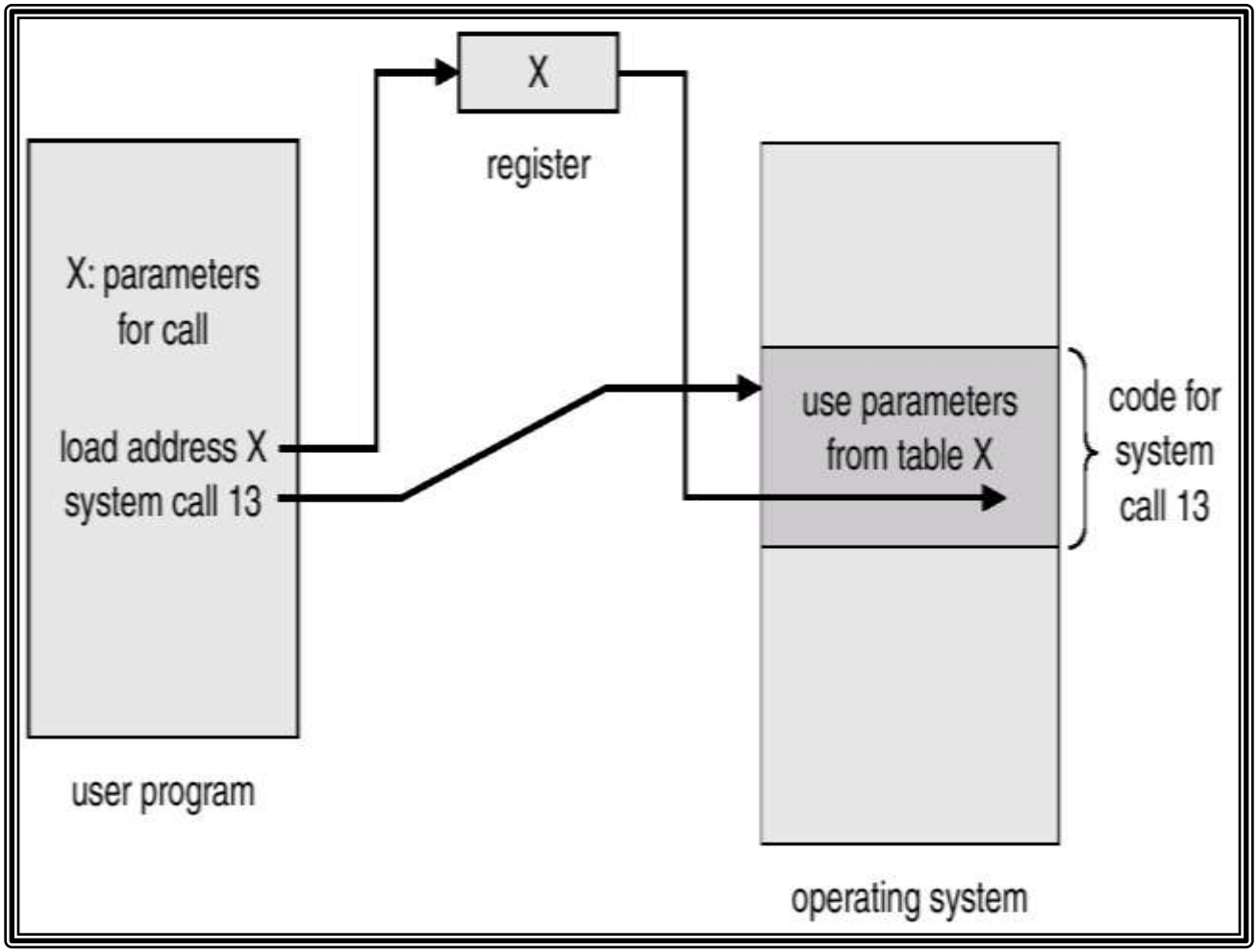
- Operating system services are provided in many different ways. Two method of providing services are system calls and system programs.
- System Calls:

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++)

- Three general methods are used to pass parameters between a running program and the operating system.
  - Pass parameters in *registers*.
  - Store the parameters in a table in memory, and the table address is passed as a parameter in a register. Ex. Linux.
  - *Push* (store) the parameters onto the *stack* by the program, and *pop* off the stack by operating system.

Passing of Parameters As A Table



Types of System Calls

- Process control: end, abort, load, execute etc.
- File management: create, delete, open, close etc.
- Device management: read, write, request device, release device etc.
- Information maintenance: get time or date, set time or date, set system date, get system data etc.
- Communications: create or release connection, send, receive message etc.

System Programs

- System programs provide a convenient environment for program development and execution. The can be divided into:
  - File manipulation
  - Status information
  - File modification
  - Programming language support
  - Program loading and execution
  - Communications
  - Application programs
- Most users' view of the operation system is defined by system programs, not the actual system calls.

Operating System View

- The view of an operating system seen by the user is defined mainly by the system programs particularly the command interpreter.
- The interrupt driven nature of an operating system defines the general structure. When an interrupt occurs, the hardware transfers control to the operating system.
- Several different types of interrupts may occur:

- A system call
- An I/O device interrupt
- A program error