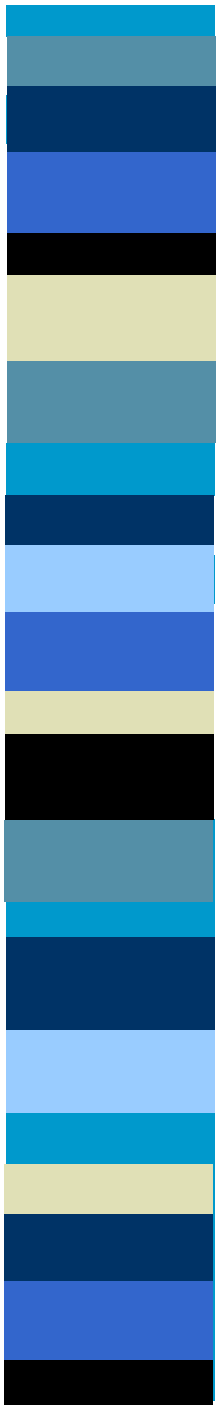


System Calls

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 forsystems programming allow system calls to be made directly (e.g., C. Bliss, PL/360, PERL ,C,C++)

System calls can be grouped into 5 categories:

Process Control: end, abort, load, execute, create process, terminate process, allocate and free memory.

File Manipulation: create file, delete file, open file, close file, read file, and write file.

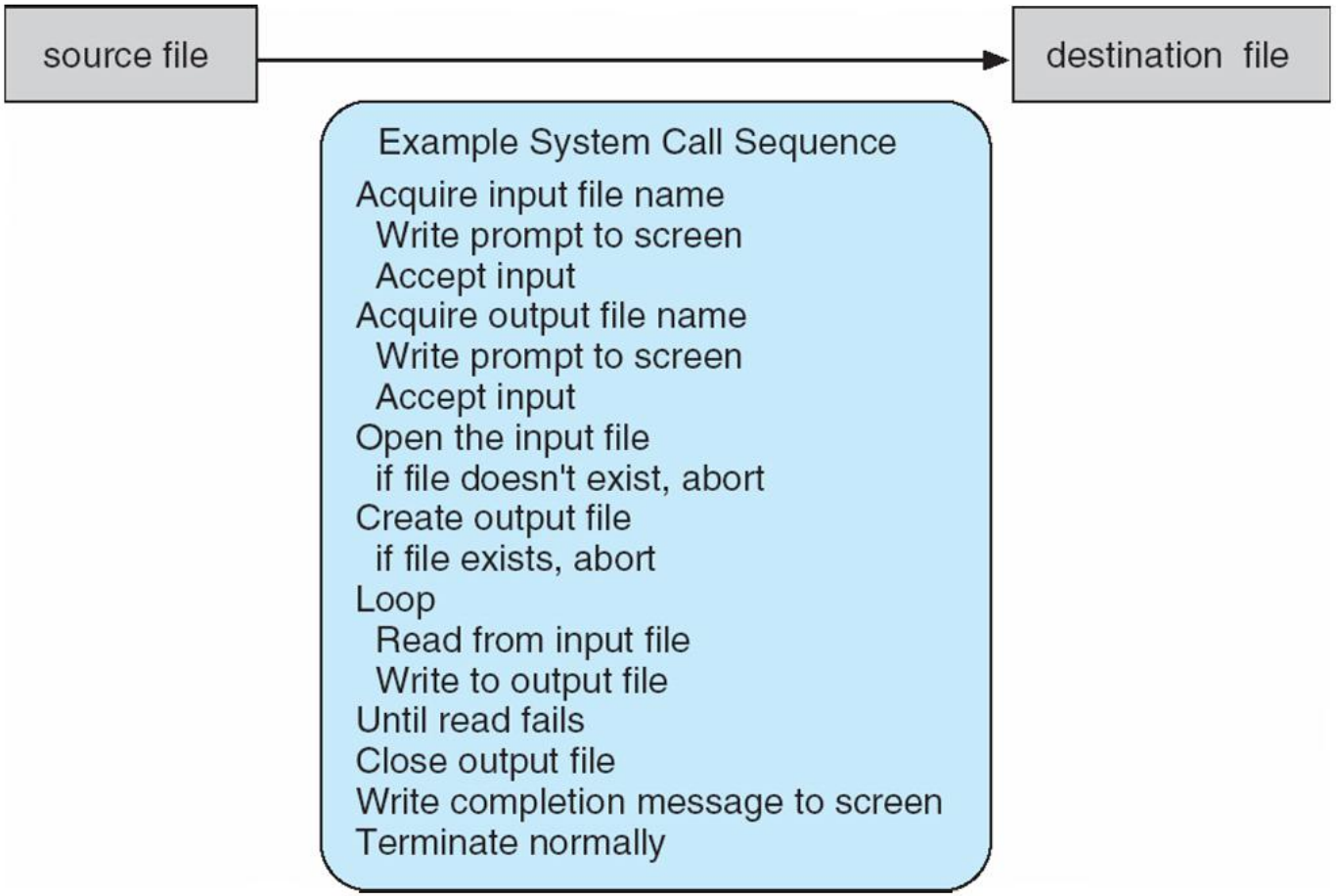
Device Manipulation: request device, release device, read, write.

Information Maintenance: get time or date, set time or date, get process or file or device.

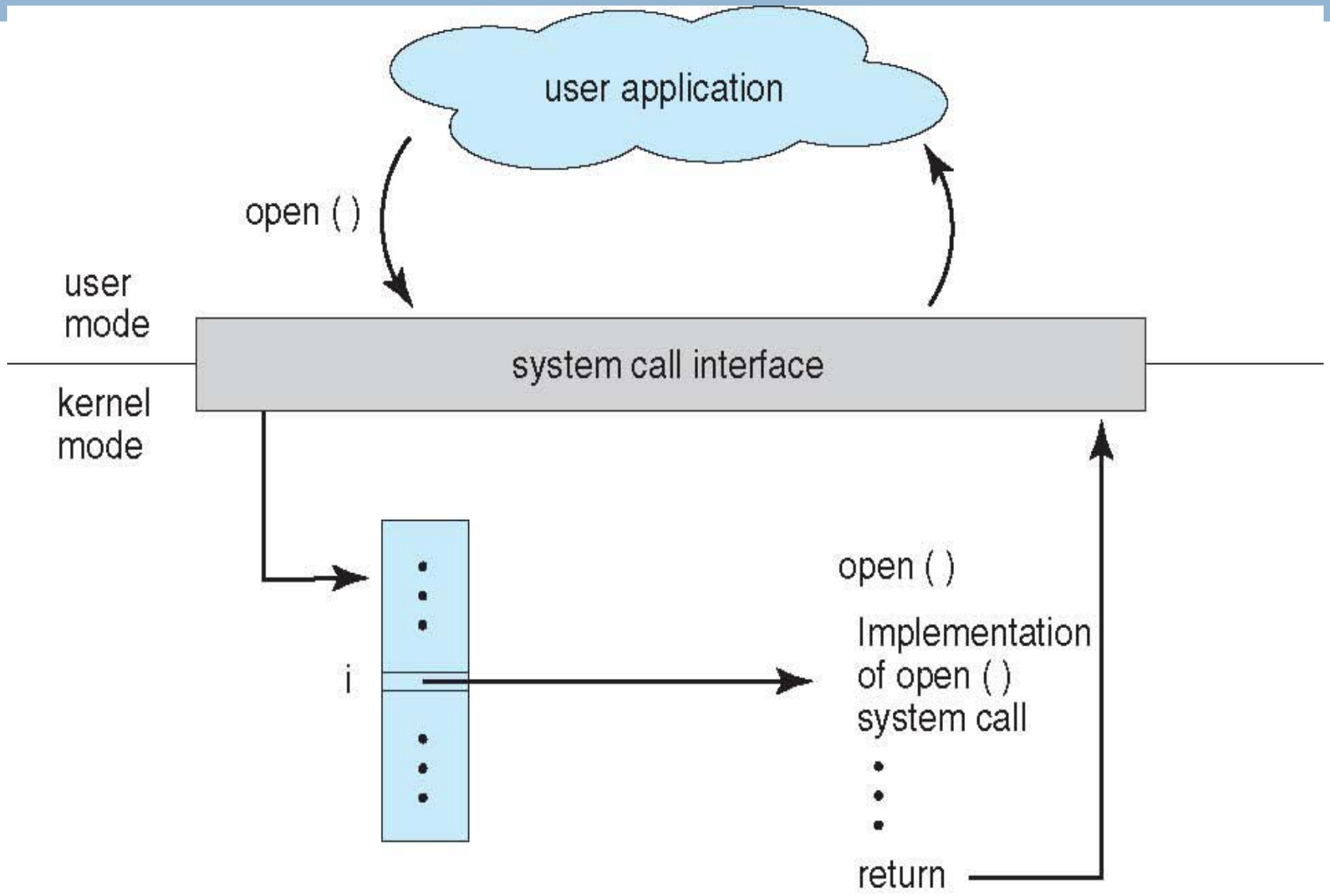
Communications: create or delete communication connection, send and receive messages.

Example of System Calls

- System call sequence to copy the contents of one file to another file



API – System Call – OS Relationship



Examples of Windows and Unix System Calls

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

Operating-System Operations

Interrupt driven by hardware

- Software error or request creates **exception** or **trap**
- Division by zero, request for operating system service
  - Other process problems include infinite loop, processes modifying each other or the operating system

- **Dual-mode** operation allows OS to protect itself and other system components
- **User mode** and **kernel mode**

  - **Mode bit** provided by hardware

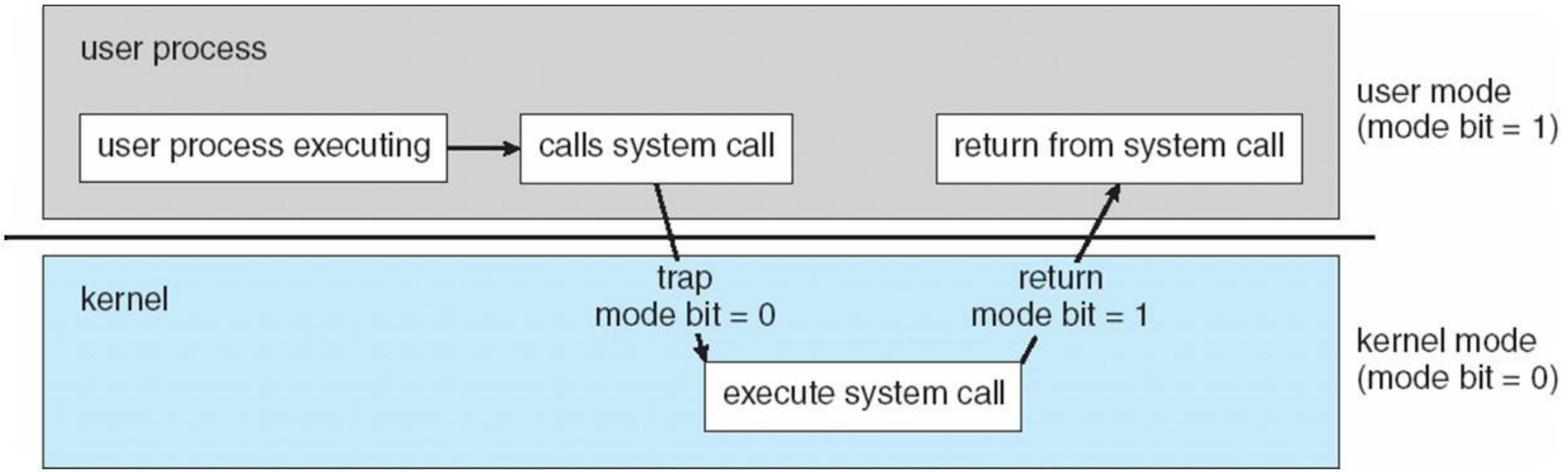
  - Provides ability to distinguish when system is running user code or kernel code

    - Some instructions designated as **privileged**, only executable in kernel mode

    - System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
- Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



EVOLUTION OF OS:

Evolution of an OS from simple Batch processing to today's OS-

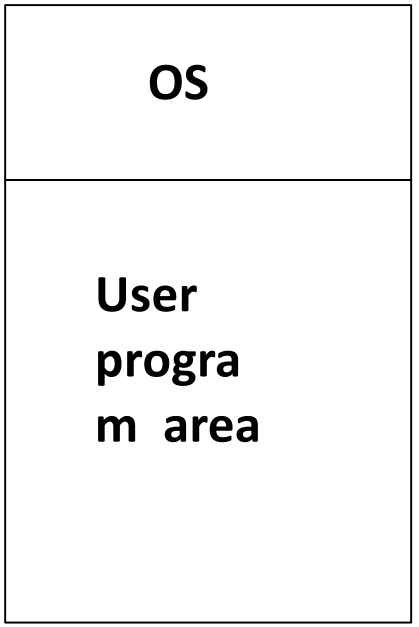
- Batch Operating System
- Multiprogramming Operating System
- Time Sharing or Multitasking Operating System
- Real-Time Systems Distributed
- Operating System Multiprocessor
- Systems

BATCH PROCESSING:

- In Batch processing same type of jobs batch *(BATCH- a set of jobs with similar needs)* together and execute at a time.
- The OS was simple, its major task was to transfer control from one job to the next.
- The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
- The OS was always resident in memory.
- Common Input devices were card readers and tape drives.

Common output devices were line *printers*, *tape drives*, and *card punches*.

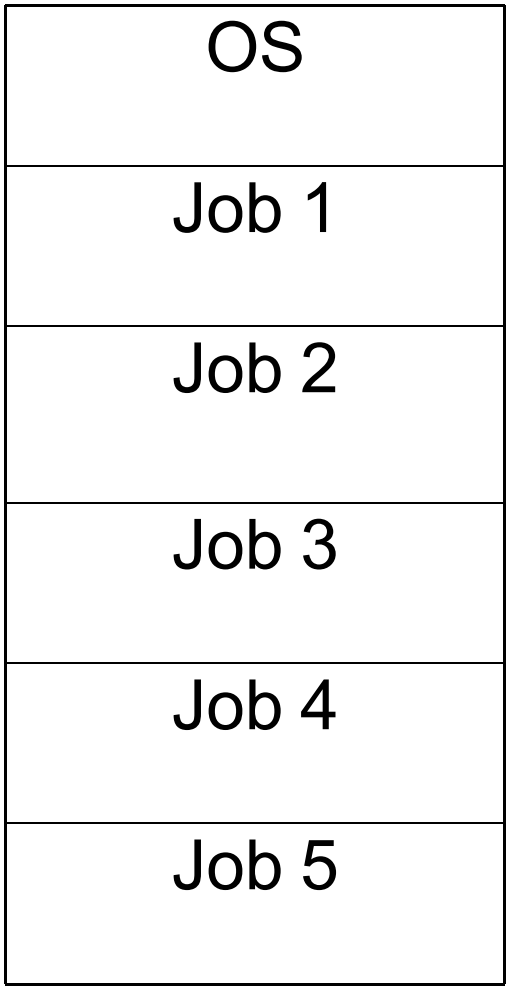
Users did not interact directly with the computer systems, but he prepared a job (comprising of the program, the data, & some control information).



MULTIPROGRAMMING:

- Multiprogramming is a technique to execute number of programs simultaneously by a single processor.
- In Multiprogramming, number of processes reside in main memory at a time.
- The OS picks and begins to executes one of the jobs in the main memory.
- If any I/O wait happened in a process, then CPU switches from that job to another job.
- Hence CPU in not idle at any time.

MULTIPROGRAMMING



- Figure dipicts the layout of multiprogramming system.
- The main memory consists of 5 jobs at a time, the CPU executes one by one.

Advantages:

- Efficient memory utilization
- Throughput increases
- CPU is never idle, so performance increases.

TIME SHARING SYSTEMS:

- Time sharing, or multitasking, is a logical extension of multiprogramming.
- Multiple jobs are executed by switching the CPU between them.
- In this, the CPU time is shared by different processes, so it is called as “Time sharing Systems”.
- Time slice is defined by the OS, for sharing CPU time between processes.
- Examples: Unix.

REAL-TIME SYSTEMS:

- A real-time operating system is a multitasking operating system intended for applications with fixed deadlines.
- Such applications include some small embedded systems, automobile engine controllers, industrial robots, spacecraft, industrial control, and some large-scale computing systems.
- An early example of a large-scale real-time operating system was Transaction Processing Facility developed by American Airlines and IBM for the Sabre Airline Reservations System.

These are of two types:

a) Hard Real Time

These OS guarantee that critical tasks be completed within a certain range of time.

Ex: A complete car welding by robot hardly on the time,controlling scientific experiments, medical imaging systems, industrial control systems.

b) Soft Real Time.

A soft real time system is the system that can not fully guarantees the completion of a particular job within the certain time limit.

These OS provides some relaxation in time limit. These system are

based on priorities.

Examples : Mobile phone, digital cameras,Multimedia systems, digital audio system.

DISTRIBUTED SYSTEMS:

- A distributed computer system is a collection of autonomous computer systems capable of communication and cooperation via their H/w and S/w interconnection.
- The distributed operating system provides a illusion to its users that it has a single uni-processor system, although it is actually consisted of multiprocessors.
- Distributed OS provide the means for system-wide sharing of resources such as computational capacity, files and I/O devices.
- Ex: UNIX, LINUX.

Multiprocessor or Parallel Systems

- Multiprocessor systems with more than on CPU in close communication.

Tightly coupled system or Symmetric multiprocessing (SMP)

– Processors share memory and a clock; communication usually takes place through the shared memory.

- Each processor runs and identical copy of the operating system.

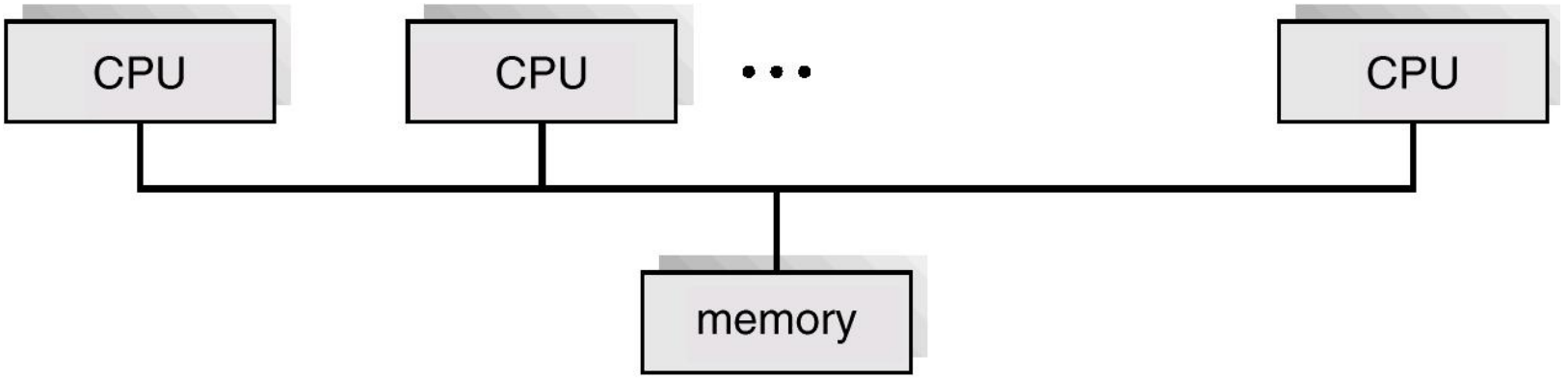
- Most modern operating systems support SMP

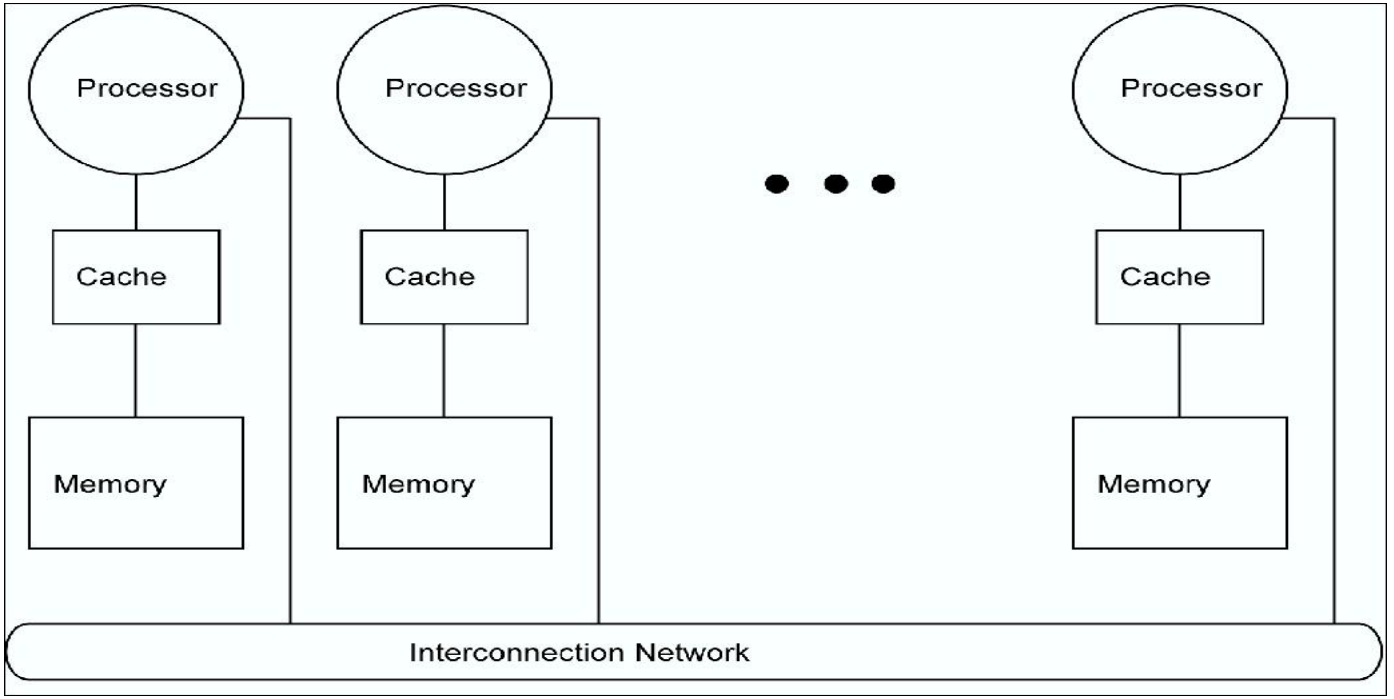
2) *Loosely coupled system or Asymmetric multiprocessing (ASMP) :*

– Each processor has its own local memory; processors communicate with one another through various communications lines,such as high speed buses or telephone lines.

- Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.

Multiprocessor or Parallel Systems

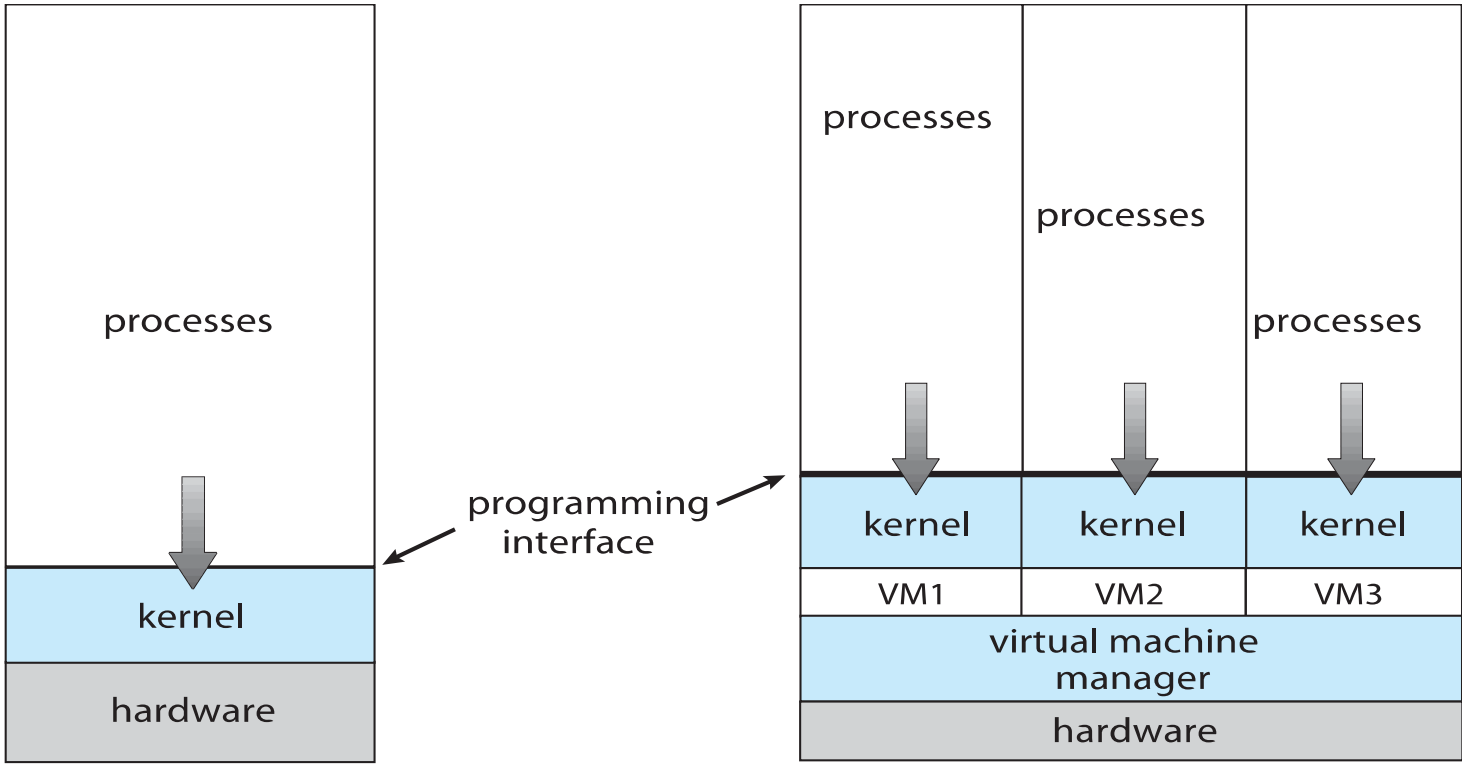




Virtual Machines

- Fundamental idea – abstract hardware of a single computer into several different execution environments
  - Similar to layered approach
  - But layer creates virtual system (**virtual machine**, or **VM**) on which operation systems or applications can run
- Several components
  - **Host** – underlying hardware system
  - **Virtual machine manager** (**VMM**) or **hypervisor** – creates and runs virtual machines by providing interface that is ***identical*** to the host
    - (Except in the case of paravirtualization)
  - **Guest** – process provided with virtual copy of the host
    - Usually an operating system
- Single physical machine can run multiple operating systems concurrently, each in its own virtual machine

System Models



(a) Nonvirtual machine

(b) Virtual machine

Implementation of VMMs

- Vary greatly, with options including:
  - **Type 0 hypervisors** - Hardware-based solutions that provide support for virtual machine creation and management via firmware
    - IBM LPARs and Oracle LDOMs are examples
  - **Type 1 hypervisors** - Operating-system-like software built to provide virtualization
    - Including VMware ESX, Joyent SmartOS, and Citrix XenServer
  - **Type 1 hypervisors** – Also includes general-purpose operating systems that provide standard functions as well as VMM functions
    - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM
  - **Type 2 hypervisors** - Applications that run on standard operating systems but provide VMM features to guest operating systems
    - Includeing VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox

History

- First appeared in IBM mainframes in 1972
- Allowed multiple users to share a batch-oriented system
- Formal definition of virtualization helped move it beyond IBM
  - A VMM provides an environment for programs that is essentially identical to the original machine
  - Programs running within that environment show only minor performance decreases
  - The VMM is in complete control of system resources

In late 1990s Intel CPUs fast enough for researchers to try virtualizing on general purpose PCs

- **Xen** and **VMware** created technologies, still used today
- Virtualization has expanded to many OSes, CPUs, VMMs

Benefits and Features

- Host system protected from VMs, VMs protected from each other
  - I.e. A virus less likely to spread
  - Sharing is provided though via shared file system volume, network communication
- Freeze, **suspend**, running VM
  - Then can move or copy somewhere else and **resume**
  - Snapshot of a given state, able to restore back to that state
    - Some VMMs allow multiple snapshots per VM
  - **Clone** by creating copy and running both original and copy
- Great for OS research, better system development efficiency
- Run multiple, different OSes on a single machine
  - **Consolidation**, app dev, …
- **Templating** – create an OS + application VM, provide it to customers, use it to create multiple instances of that combination
- **Live migration** – move a running VM from one host to another!
  - No interruption of user access
- All those features taken together -> **cloud computing**
  - Using APIs, programs tell cloud infrastructure (servers, networking, storage) to create new guests, VMs, virtual desktops

Implementation of VMMs (cont.)

- Other variations include:
  - **Paravirtualization** - Technique in which the guest operating system is modified to work in cooperation with the VMM to optimize performance
  - **Programming-environment virtualization** - VMMs do not virtualize real hardware but instead create an optimized virtual system
    - Used by Oracle Java and Microsoft.Net
  - **Emulators** – Allow applications written for one hardware environment to run on a very different hardware environment, such as a different type of CPU
  - **Application containment** - Not virtualization at all but rather provides virtualization-like features by segregating applications from the operating system, making them more secure, manageable
    - Including Oracle Solaris Zones, BSD Jails, and IBM AIX WPARs
- Much variation due to breadth, depth and importance of virtualization in modern computing

Buffering

- A temporary storage area (buffers) to read data from input device or send data to the output device
- keep CPU busy–because I/O operation is slow
- Reading and writing data from hard disk takes long time. so to improve the speed for data processing the data next required by processor is stored is cache memory or CPU register.
- for e.g. to cut certain line from text file to copy into another file. cut data get stored in to buffer (CPU register) to get back stored into another file.

Spooling

- Spooling is an acronym for **simultaneous peripheral operations on line**. Spooling refers to putting data of various I/O jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices.

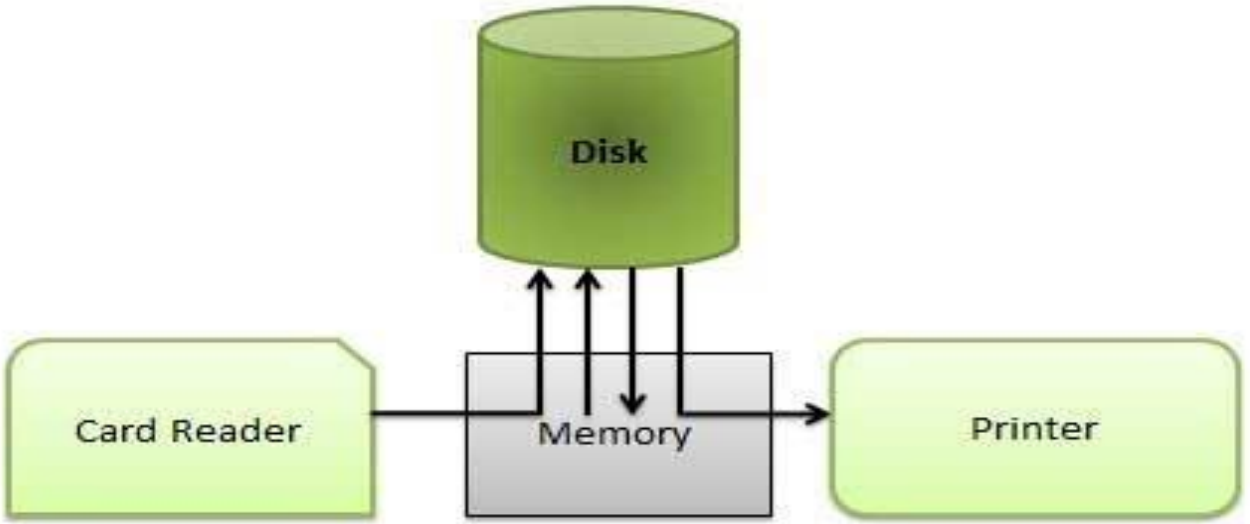
  Operating system does the following activities related to distributed environment.

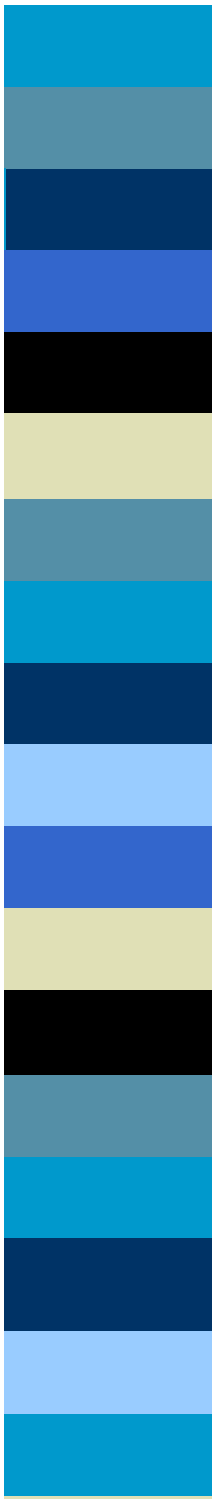
- OS handles I/O device data spooling as devices have different data access rates.
- OS maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.
- OS maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion.
- It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task.

Spooling

Advantage

- The spooling operation uses a disk as a very large buffer.
- Spooling is capable of overlapping I/O operation for one job with processor operations for another job.





Thank You