OPERATING SYSTEMS

OVERVIEW

Jerry Breecher

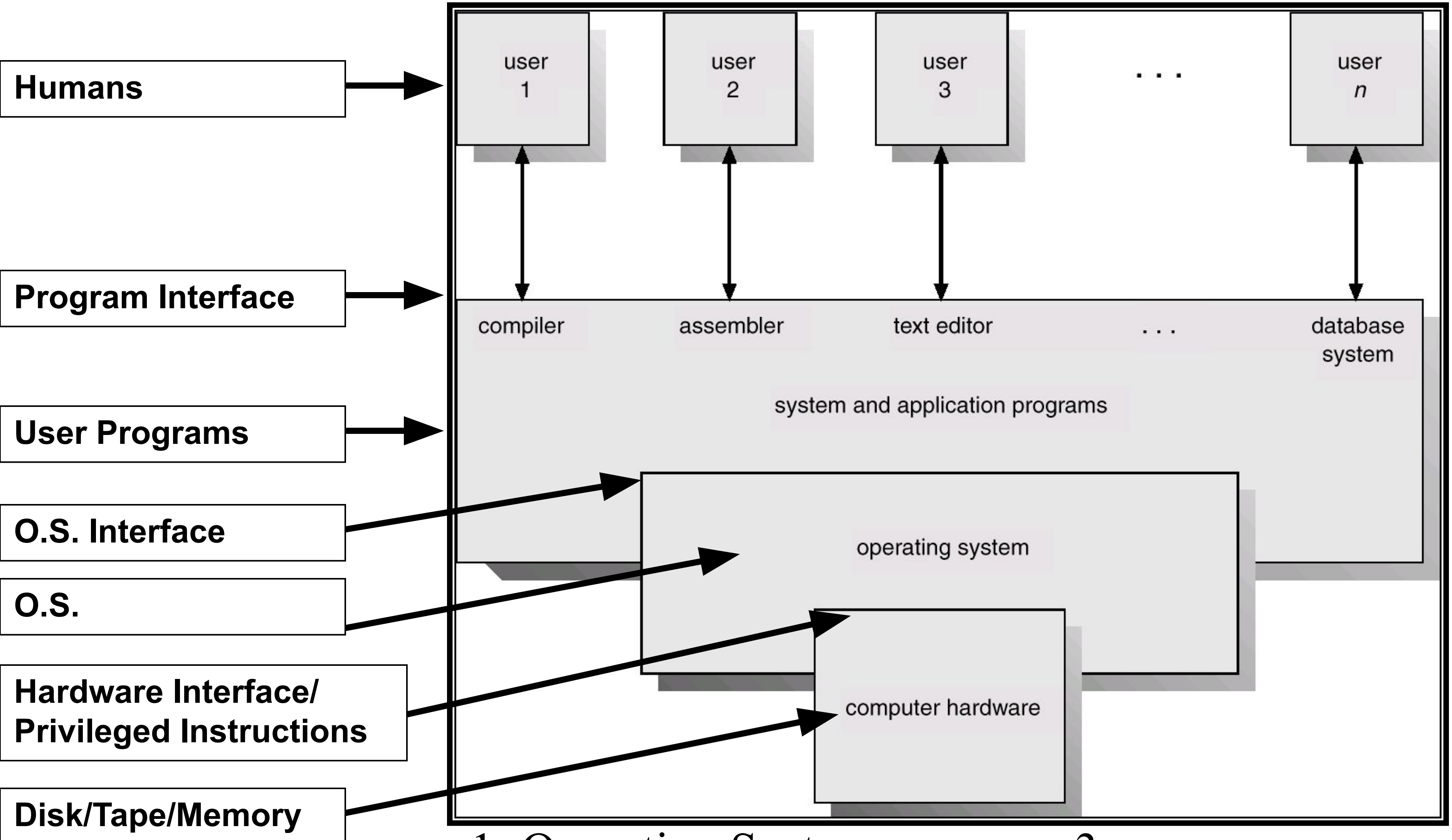
OPERATING SYSTEM OVERVIEW

WHAT IS AN OPERATING SYSTEM?

- An interface between users and hardware - an environment "architecture”
- Allows convenient usage; hides the tedious stuff
- Allows efficient usage; parallel activity, avoids wasted cycles
- Provides information protection
- Gives each user a slice of the resources
- Acts as a control program.

OPERATING SYSTEM OVERVIEW

The Layers Of A System



OPERATING SYSTEM OVERVIEW

Components

A mechanism for scheduling jobs or processes. Scheduling can be as simple as running the next process, or it can use relatively complex rules to pick a running process.

A method for simultaneous CPU execution and IO handling. Processing is going on even as IO is occurring in preparation for future CPU work.

Off Line Processing; not only are IO and CPU happening concurrently, but some off-board processing is occurring with the IO.

OPERATING SYSTEM OVERVIEW

Components

The CPU is wasted if a job waits for I/O. This leads to:

- **Multiprogramming** (dynamic switching). While one job waits for a resource, the CPU can find another job to run. It means that several jobs are ready to run and only need the CPU in order to continue.

CPU scheduling is the subject of Chapter 6.

All of this leads to:

- **memory management**
- **resource scheduling**
- **deadlock protection**

which are the subject of the rest of this course.

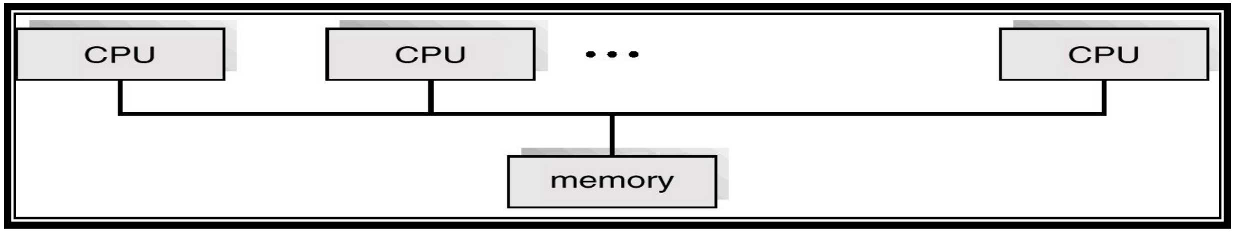
OPERATING SYSTEM OVERVIEW

Characteristics

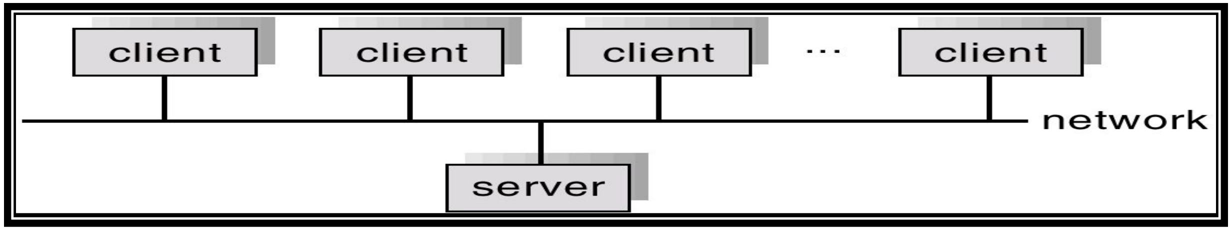
Other Characteristics include:

- **Time Sharing** - multiprogramming environment that's also interactive.

- **Multiprocessing** - Tightly coupled systems that communicate via shared memory. Used for scientific applications. Used for speed improvement by putting together a number of off-the-shelf processors.



- **Distributed Systems** - Loosely coupled systems that communicate via message passing. Advantages include resource sharing, speed up, reliability, communication.



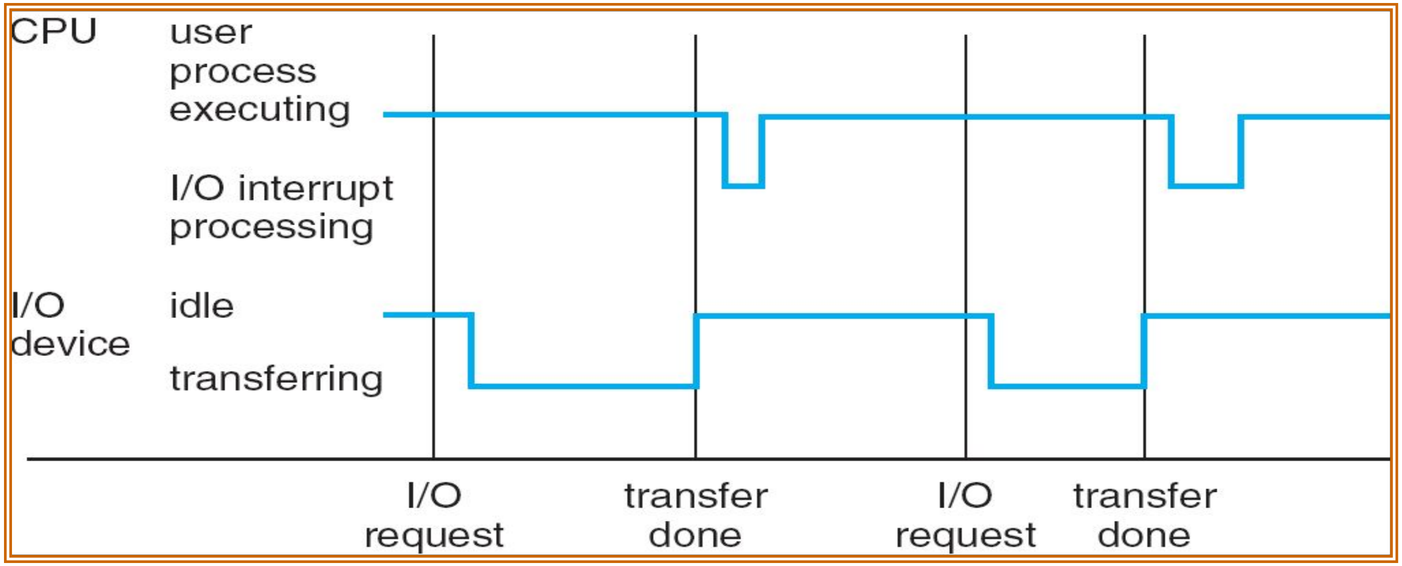
- **Real Time Systems** - Rapid response time is main characteristic. Used in control of applications where rapid response to a stimulus is essential.

OPERATING SYSTEM OVERVIEW

Characteristics

Interrupts:

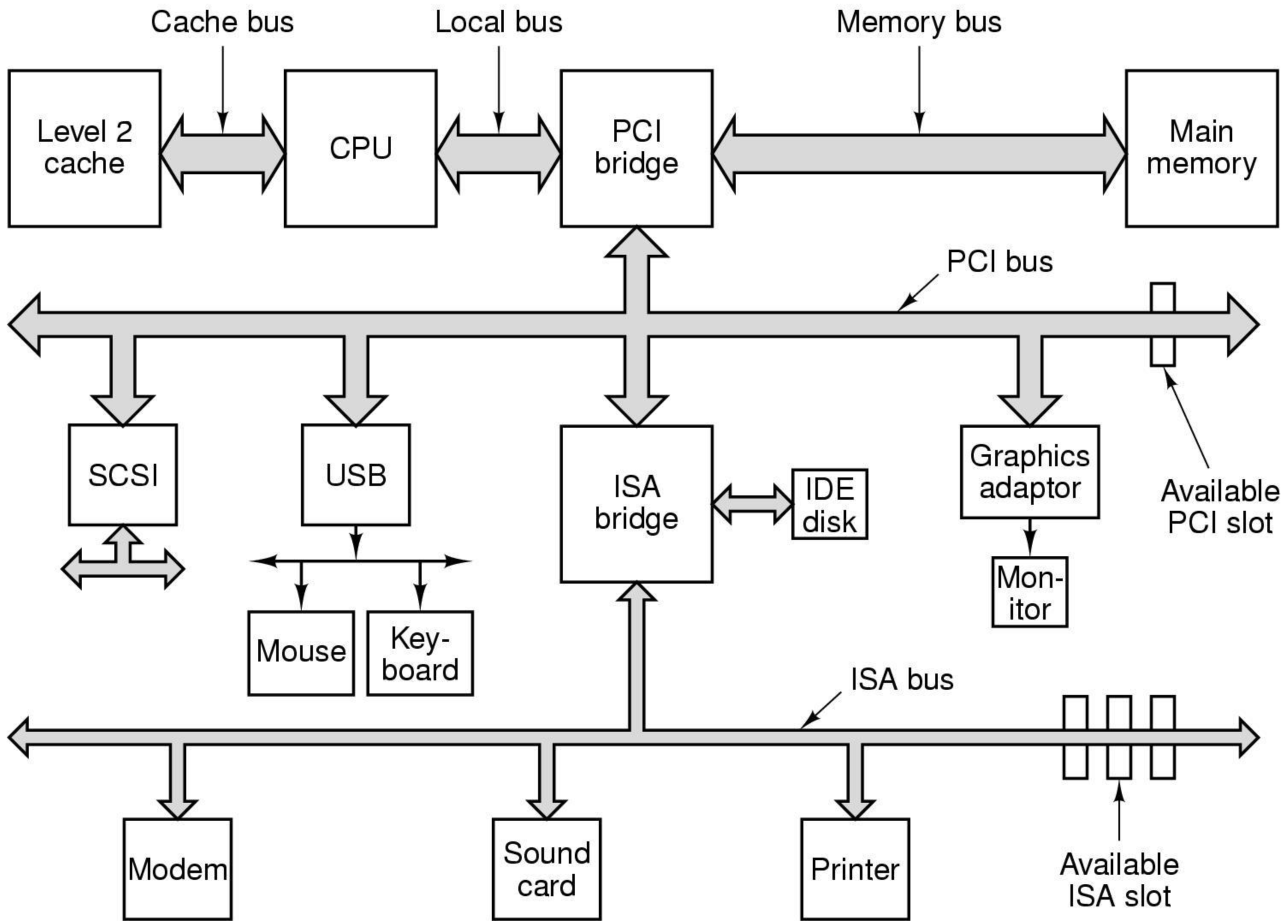
- Interrupt transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*.
- A *trap* is a software-generated interrupt caused either by an error or a user request.
- An operating system is *interrupt* driven.



OPERATING SYSTEM OVERVIEW

Hardware Support

These are the devices that make up a typical system.

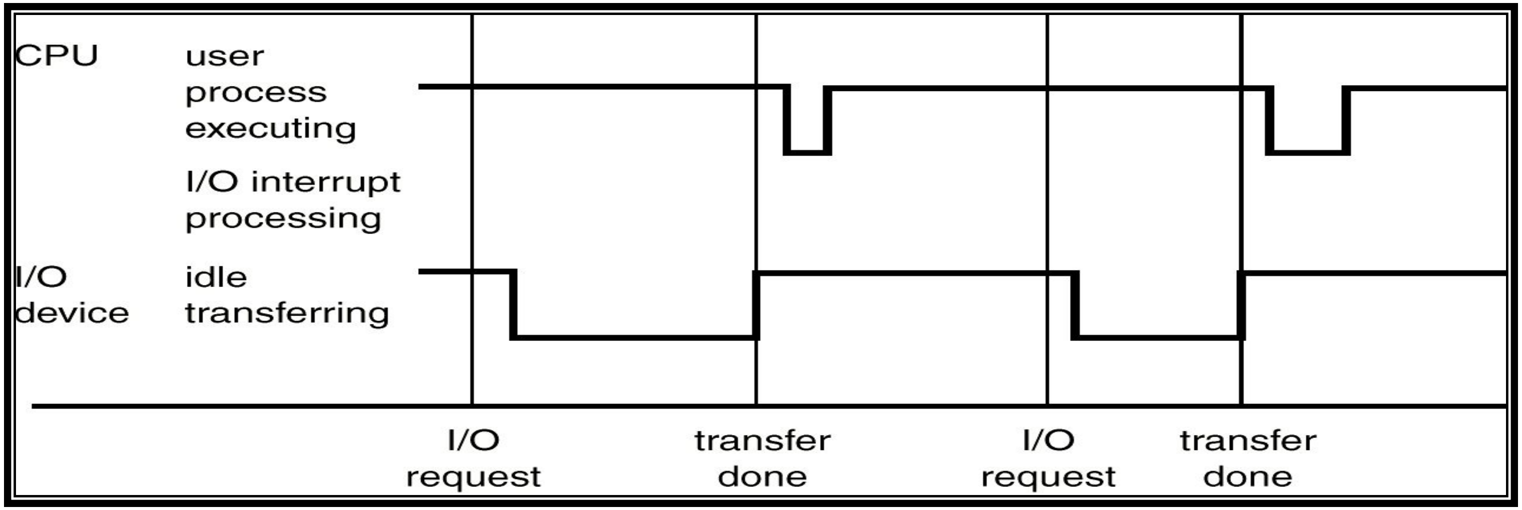


Any of these devices can cause an electrical interrupt that grabs the attention of the CPU.

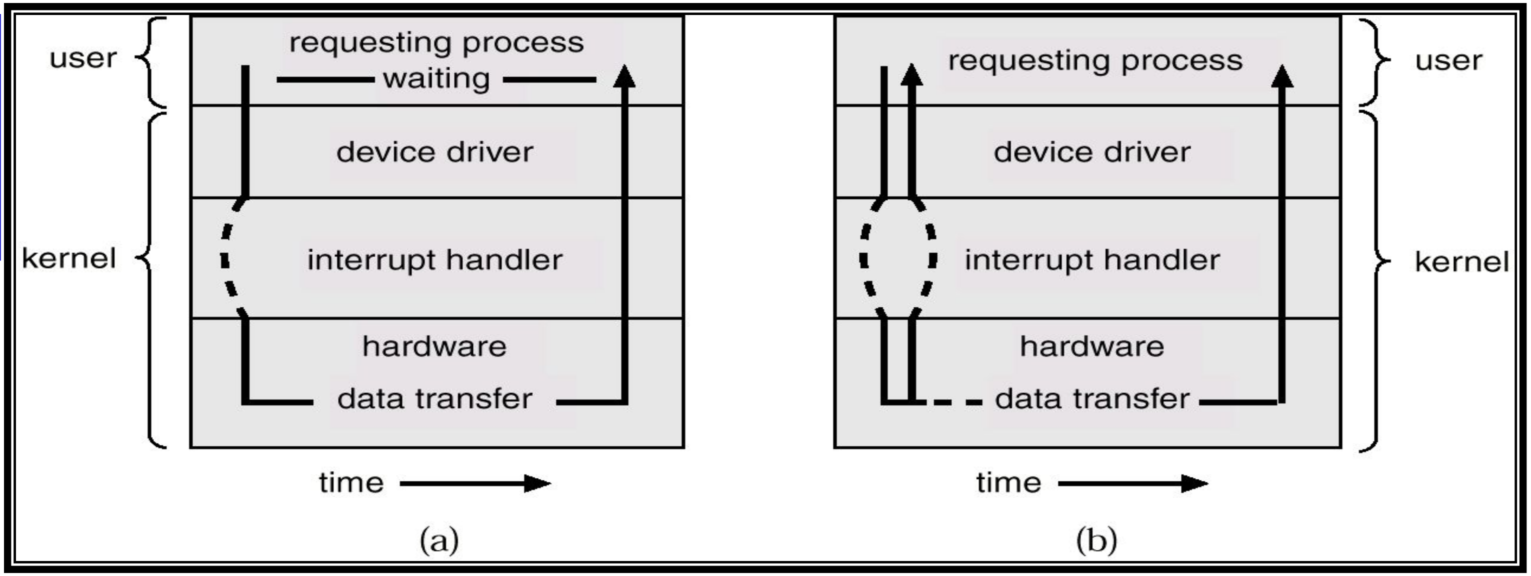
OPERATING SYSTEM OVERVIEW

Hardware Support

Sequence of events for processing an IO request.

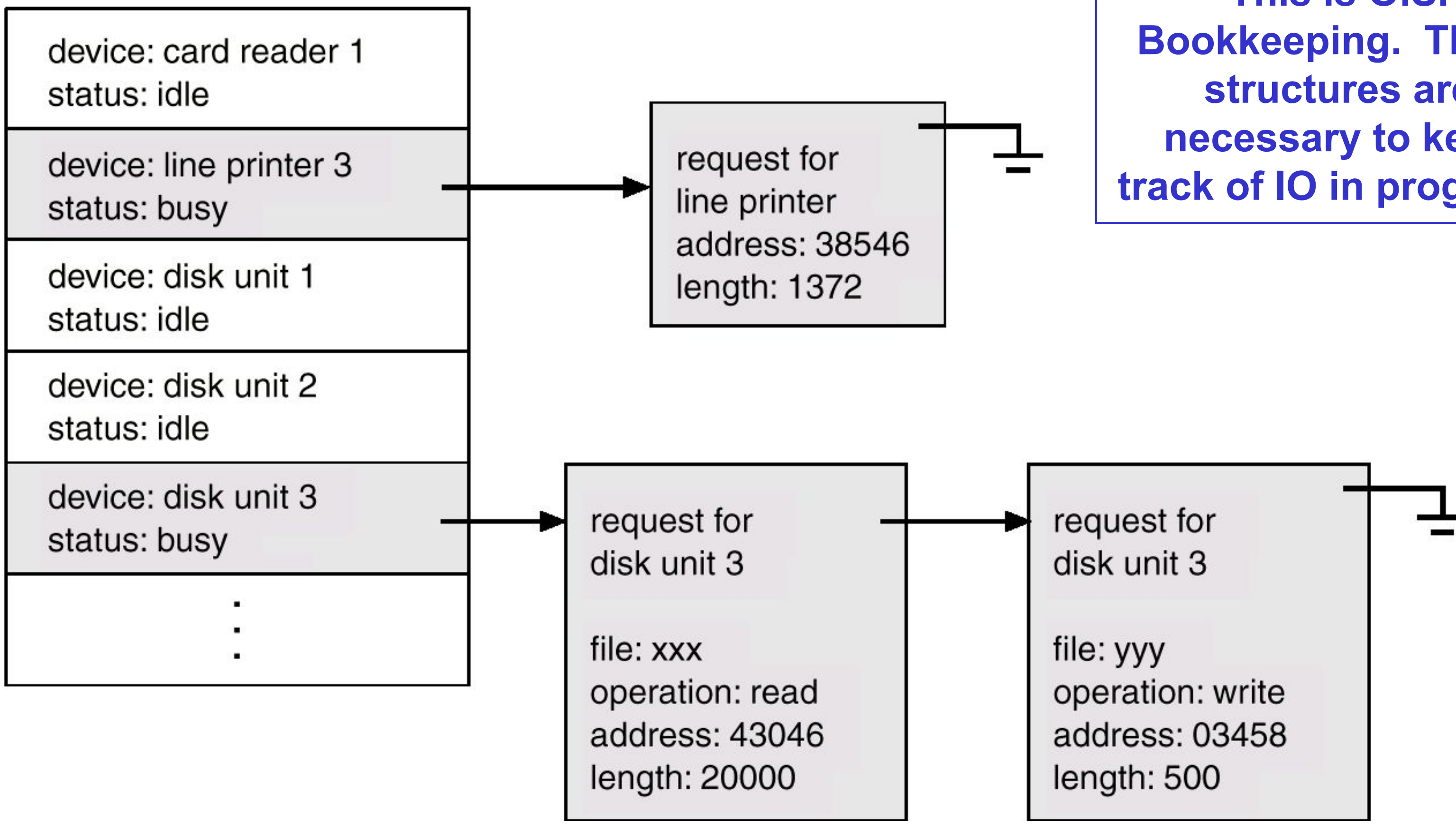


Comparing Synchronous and Asynchronous IO Operations



OPERATING SYSTEM OVERVIEW

Hardware Support



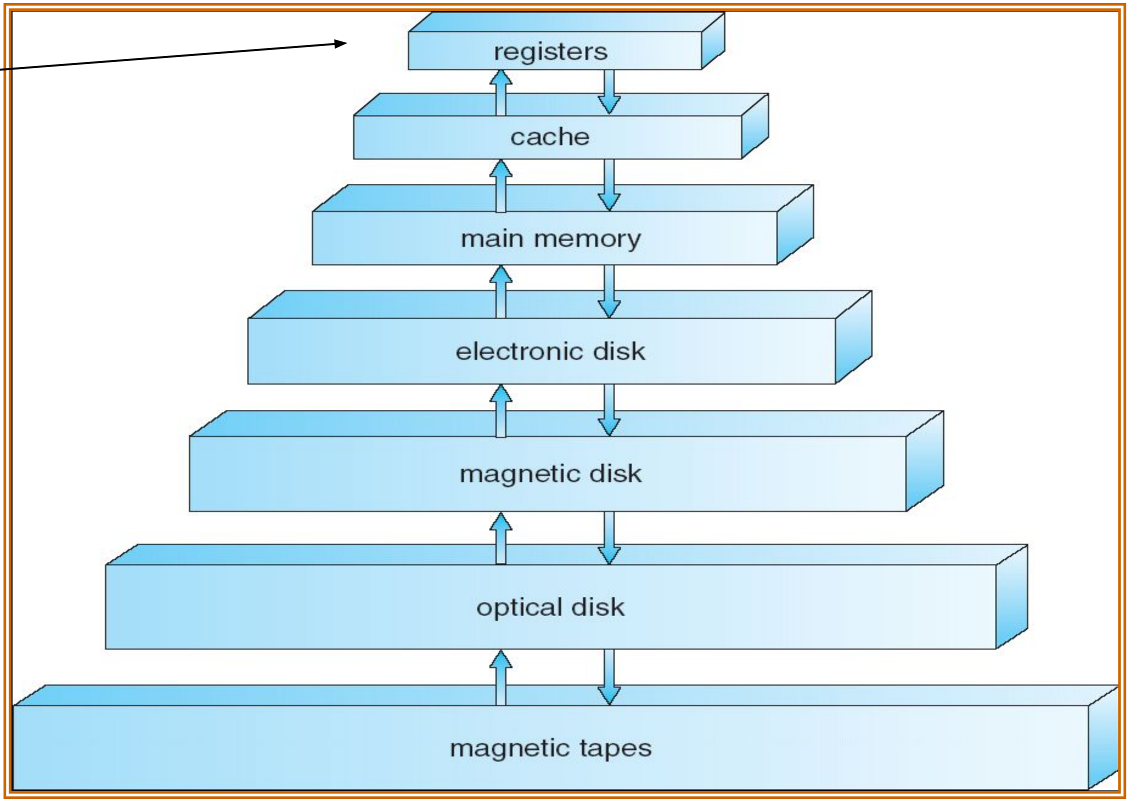
This is O.S. Bookkeeping. These structures are necessary to keep track of IO in progress.

OPERATING SYSTEM OVERVIEW

Storage Hierarchy

Very fast storage is very expensive. So the Operating System manages a hierarchy of storage devices in order to make the best use of resources. In fact, **considerable** effort goes into this support.

Fast and Expensive



Slow an Cheap

OPERATING SYSTEM OVERVIEW

Storage Hierarchy

Performance:

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Storage Hierarchy

Caching:

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

OPERATING SYSTEM OVERVIEW

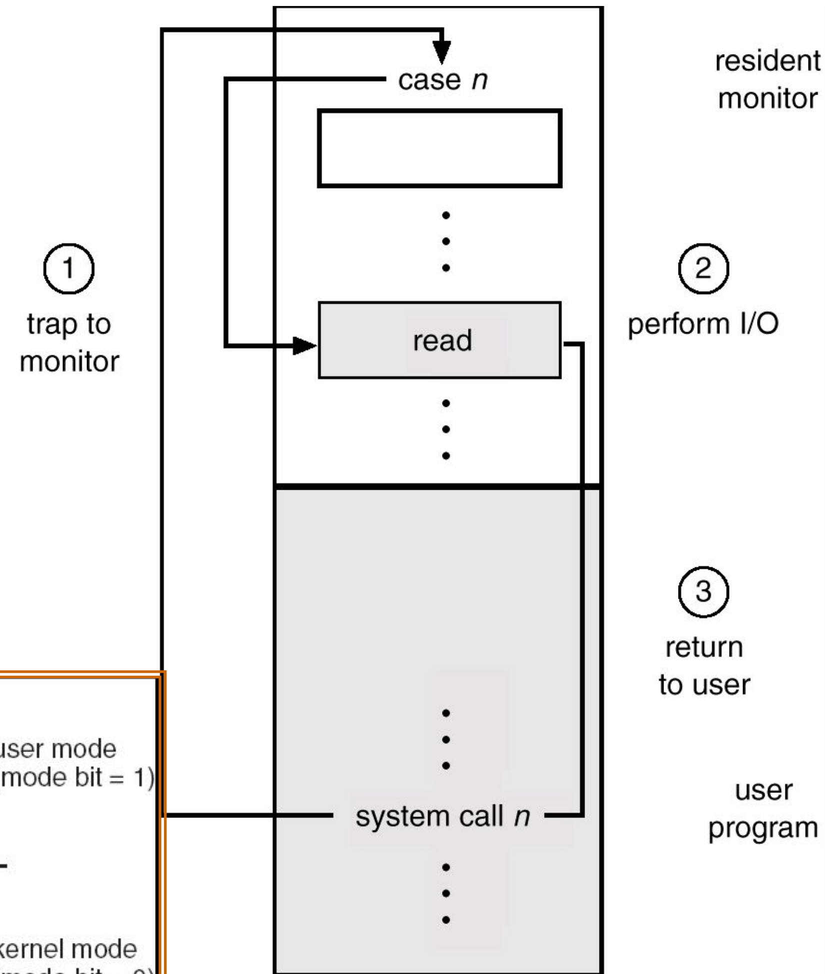
Protection

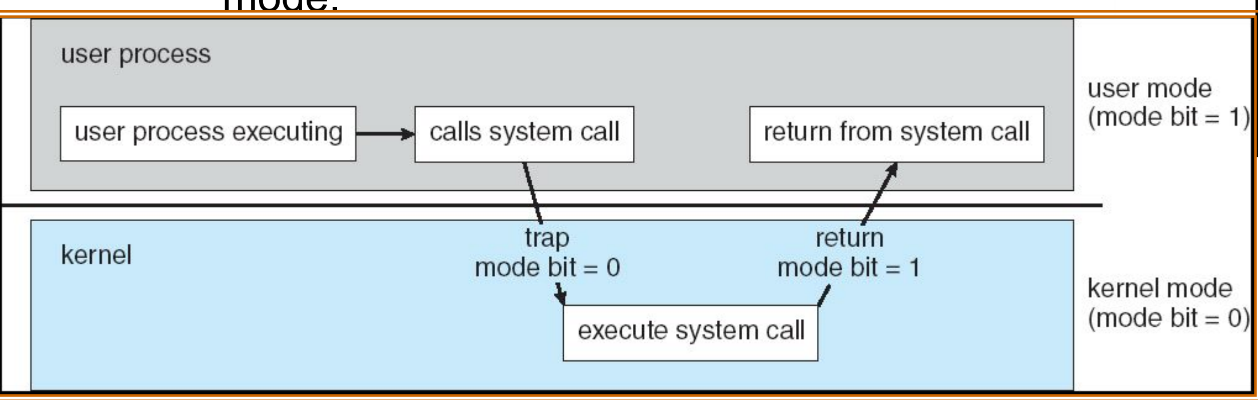
The goal is protecting the Operating System and others from malicious or ignorant users.

The **User/Supervisor Mode** and **privileged** instructions.

Concurrent threads might interfere with others. This leads to protection of resources by user/supervisor mode. These resources include:

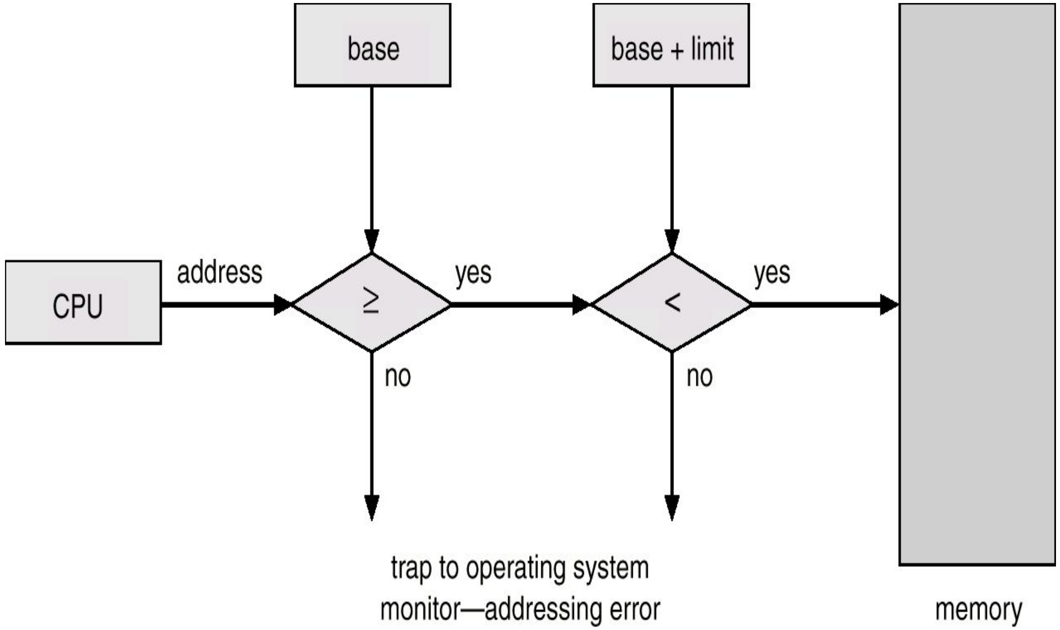
I/O Define I/O instructions as privileged; they can be executed only in Supervisor mode. System calls get us from user to supervisor mode.





Protection

Memory A user program can only access its own logical memory. For instance, it can't modify supervisor code. Depends on an address translation scheme such as that shown here.



OPERATING SYSTEM OVERVIEW

Protection

CPU A clock prevents programs from using all the CPU time. This clock causes an interrupt that causes the operating system to gain control from a user program.

For machines connected together, this protection must extend across:

Shared resources,
Multiprocessor Architectures,
Clustered Systems

The practice of this is called “distributed operating systems”.

OPERATING SYSTEM OVERVIEW

WRAPUP

We've completed our first overview of an Operating System – this was the equivalent of a Satellite picture.

The next view will be at the level of a high flying plane.

After that, we'll be at ground level, looking at pieces in detail.