OPERATING SYSTEMS:
OVERVIEW AND STRUCTURES

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.

The Layers Of A System:

USER (Human)

V <---- Interface (Dependent on program)
V

APPLICATION PROGRAM (Compiler, editor, games ..)

V <---- Interface (System calls) --- OS Interface -- API
V

OPERATING SYSTEM (Serves program requests)

V <---- Interface (Hardware/privileged instructions)
V

HARDWARE (Disk, tape, memory, CPU)

V
V

CHARACTERISTICS OF OPERATING SYSTEMS:

- A mechanism for scheduling jobs or processes. This was at one time called a monitor. Scheduling can be as simple as running the next process, or it can become relatively complicated.

- 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.

- 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 4.

- All of this leads to:

  memory management

  resource scheduling

  deadlock protection

which are the subject of the rest of this course.

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.

HARDWARE SUPPORT FOR OPERATING SYSTEMS:

Interrupts - a device kicks the CPU in order to get service. The CPU no longer needs to poll.

- Depend on interrupts to determine what is to be done next.
- Hardware and Software interrupts.
- Can sit and wait for an interrupt or run another user.
- The interrupt handler chooses the code to be run for a particular device.
- Device table gives the status for each device.
  <<< SEE FIGURE 2.3 >>>

DMA (Direct Memory Access) I/O controllers have access to host memory, without bothering the CPU.

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 & Expensive >>

Registers
|
Cache
|
Main Memory
|
Electronic Disk
|
Magnetic Disk
|
Optical Disk
|
Magnetic Tapes

<< Slow & Inexpensive >>

PROTECTION:

Protecting the Operating System and other users from errant users.

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

Concurrent jobs 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.
<<< SEE FIGURE 2.9>>>

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.
<<< SEE FIGURE 2.8 >>> for simplistic example.

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".