EE 469 Operating Systems Engineering

Introduction to Operating Systems

Spring 2001

David S. Ebert

EE 469: Operating Systems Engineering

What's this course about?

- A) Design of operating rooms
- B) Some boring computer software you used to have to know in the dark ages
- C) Important component of every computer system that every computer engineer and programmer should understand

Why Do We Care About Operating Systems?

Why do I care about Operating Systems since I do research in photorealistic graphics, modeling natural phenomena, real-time graphics and visualization?

Examples of Graphics Problems

Pixar animation (e.g., Toy Story, Monsters Inc.)

- avg, 20 min /frame, 24 frame / s, 80 minutes = 38400 minutes to computer the final animation = 26.67 days on 1 machine
- They use a huge rendering farm of thousands of processors

Medical / Scientific Visualization - 100 Gb to terabytes of data

- want to generate images accurately and FAST
- interface with special purpose hardware

3D Displays - image inside a 3D cube - need 120 Gb/sec to drive display

VR systems

- special purpose hardware, sometime home-made
- fixed interaction time constraint 10-30 frames per second or motion sickness

Computer Games - see next slide

Why Do We Care About Operating Systems?

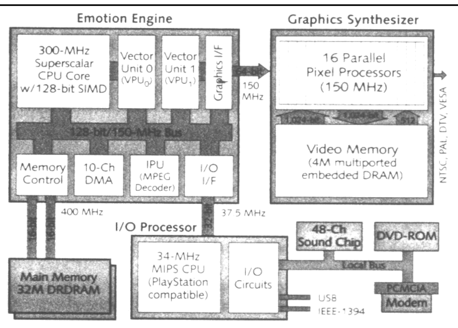
Enables the development of efficient programs

- Example <u>Program and output</u>

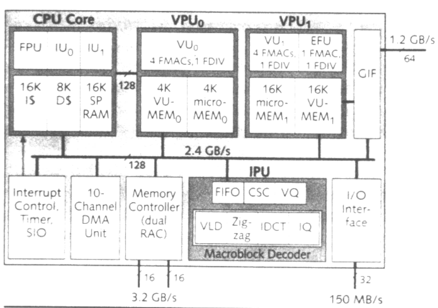
You may have to develop for a system without a high-level interface

- Example: Sony PlayStation 2

Sony PlayStation 2 Details



More PS2 Details



Some Performance Numbers for PS2

295 Mhz Emotion Engine, 6.2 GFlops, 128-bit processor

147 Mhz Rasterizer

25 million polys / second

2 Vector processing units, 1 scalar processing unit

All data transfer is DMA controlled

Most games programmed controlling all the components directly from the game with system calls

What is an Operating System?

Definition 1: ***A program that acts as an intermediary between a user of a computer and the computer hardware.***

Definition 2: ***A control program - controls the execution of user programs to prevent errors and improper use of resources***

Operating system goals:

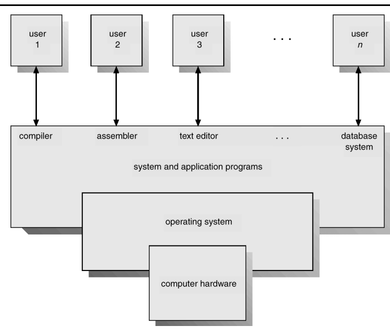
- Convenience
  - *Execute user programs and make solving user problems easier.*
  - *Make the computer system convenient to use.*
- Use the computer hardware in an efficient manner.

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



Operating System Definitions

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)

Basic Function of an O.S.

Resource Allocation

- CPU (scheduling)
- Memory (management)
- File Storage (management)
- Devices (management)

History of Operating Systems

I. Sequential Execution

- Bare Computer
  - *hands on, user did everything themselves*
- Libraries
  - *device drivers and common routings*
- **Batch O.S. - first rudimentary O.S.**
- Offline Processing

II. Overlap CPU Processing and I/O

III. Give up Some Efficiency for Convenience

Batch Operating Systems

First rudimentary operating system

Hire an operator, User ≠ operator

Add a card reader

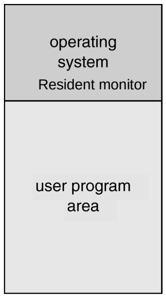
Reduce setup time by batching similar jobs

Automatic job sequencing – automatically transfers control from one job to another.

Resident monitor - always in memory

- initial control in monitor
- control transfers to job
- when job completes control transfers back to monitor
- 3 Parts - loader, job sequencer, control card interpreter

Memory Layout for a Simple Batch System



Batch O.S.

Faster than no OS - more efficient operation

Less convenience or more? - not hands on, but don't have to wait for your turn to work the computer

Problem: Slow Performance – I/O and CPU could not overlap ; card reader very slow.

Solution: Off-line Processing

Process input to faster device

More efficient use of CPU

Device Independence - program uses logical devices, not physical devices

Overlap of I/O and CPU Processing

- specialized card readers and printers
- satellite processing - small computer does transfer from/to mag tape or disk

Turnaround time worse

Overlap CPU Processing and I/O

Architecture support needed:

- Interrupt -initiated I/O
- DMA

Several Types

- Buffering
- Spooling
- Multiprogramming

Buffering

Overlap I/O of job with its execution

- After CPU reads info from buffer, input device begins next input immediately
- Doesn't wait for CPU to initiate it

I/O bound jobs

- Input buffer always empty and/or output buffer always full

CPU bound jobs

- Input buffer always full and/or output buffer always empty

Still used heavily today

Spooling (Simultaneous Peripheral Operation On-Line)

Overlap I/O of one job with computation of another (many) job(s).

While executing one job, the O.S.

- Reads next job from card reader into a storage area on the disk (job queue).
- Outputs printout of previous job from disk to printer.

CPU & I/O Utilization much higher

Job pool – ***data structure that allows the OS to select which job to run next in order to increase CPU utilization. => Job Scheduling***

Multiprogramming

Job scheduling allows multiprogramming

Greatly increases CPU Utilization

- Can have cpu always executing a job

Several jobs in memory

- execute 1, when it has to wait for I/O, start another one

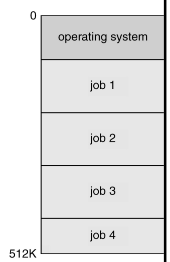
As long as there is a job to execute, CPU will never be idle

CPU Scheduling Needed:

- Fair
- Avoid Starvation

Memory Management also needed:

- Fair
- Secure



Time-Sharing Systems–Interactive Computing

User Interaction is the key

Give each user the illusion that she has full use of the entire machine.

Job Switching

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
- A job is swapped in and out of memory to the disk.
- Put limit on how long each job can use CPU before a job switch occurs

OS does more work without an increase in efficiency

Extension to multiprogramming

Personal-Computer Systems

Personal computers – ***computer system dedicated to a single user.***

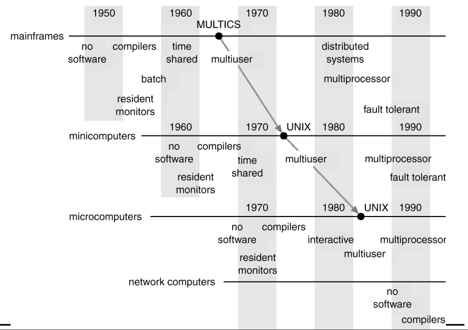
I/O devices – keyboards, mice, display screens, small printers.

User convenience and responsiveness.

Can adopt technology developed for larger operating system

Often individuals have sole use of computer and do not need advanced CPU utilization of protection features.

Migration of Operating-System Concepts and Features



Parallel Systems

Multiprocessor systems with more than one 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*

Parallel Systems (Cont.)

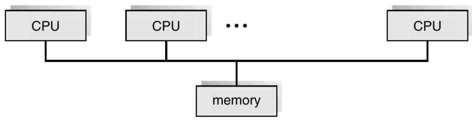
Symmetric multiprocessing (SMP)

- Each processor runs an 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 allocates work to slave processors.
- More common in extremely large systems

Symmetric Multiprocessing Architecture



Real-Time Systems

Often used as a control device

- in a dedicated application such as controlling scientific experiments, medical imaging, industrial control, etc.

Well-defined fixed-time constraints.

Hard real-time system.

- 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 system

- Limited utility in industrial control or robotics
- Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

Distributed Systems

Distribute the computation among several physical processors.

Loosely coupled system –

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

Advantages of distributed systems.

- Resources Sharing
- Computation speed up – load sharing
- Reliability
- Communications

Distributed Systems (Cont.)

Network Operating System

- provides file sharing
- provides communication scheme
- runs independently from other computers on the network

Distributed Operating System

- less autonomy between computers
- gives the impression there is a single operating system controlling the network.