Lecture 26:
How does a computer... run many applications simultaneously?

Users Run Many Applications Simultaneously

Firefox
Grab
iTunes
Microsoft PowerPoint
Microsoft Word
Preview
Scratch
Terminal
Finder

Expect all to be running, doing work for you...
How does computer do this with only 1 (or 2) CPUs?
What is an Operating System?

Operating System (OS):
- Software that converts hardware into a useful form for applications
- Very complex: millions of lines of code, 1000 person-years

What does OS do?

Manage hardware resources for applications

What do you think this entails?

Role #1: Provide *standard library* for accessing resources
- Allow applications to reuse common facilities
- Make different devices look the same
- Provide higher-level abstractions

Role #2: Coordinate usage of resources (i.e., manager)
- Virtualize resources so multiple users or applications can share
- Protect applications from one another
- Provide efficient and fair access to resources
What are different HW Resources?

CPU: Process scheduler
- Determines when and for long each process executes

Memory: Memory manager
- Determines when and how memory is allocated to processes
- Decides what to do when main memory is full

Disk: File system
- Organizes named collections of data in persistent storage

Network: Networking
- Enables processes to communicate with one another

How to run multiple applications?

OS executes “processes” not applications
- Related to application, but not identical

What is a process?
Execution stream in context of process state

Informally:
Everything you need to run an application

More processes running than you might expect!

Find by running variants of “ps” (Unix-based)
How to run multiple processes?

Multi-programming: Multiple processes resident at a time

- Same as multi-tasking
- Opposite: Uni-programming
  - Only one process resident at a time
  - Examples: First systems and DOS for PCs
- Different than multiprocessing
  - Multiprocessing: Systems with multiple processors
- Advantages: Better user convenience and system performance
  - Why does it improve system performance???

How to Support Multiprogramming?

OS provides illusion that each has HW to itself
Must be able to handle misbehaving applications (that don’t want to share!)
How to Share CPU?

Illusion?
- Switch quickly between active processes: Time sharing
- Processes must not be able to hold onto CPU forever!

OS and HW work together to perform context switch
- Change contents of registers and Program Counter (PC)
- Change active address space in memory

Context switch is mechanism
OS must also implement policy
- Many processes want to run, but which should run when??

Example:
Service at a Deli

Many customers waiting for service at deli...
In what order should customers be handled?
Easiest Policy?

First-come-first-served (FCFS)
- Customers take ticket when arrive, serve next number
- Customers add to end of line, serve next customer in line

Why do you think this a good policy?
- Easy to implement
- Intuitively Fair: Earlier you arrive, sooner you get service

Why could it be bad?
- Treats all customers identically but may have different requirements

Different Requirements?

1) Some costumers have a deadline

2) Some customers are more important than others

3) Some costumers have short orders, others very long orders
1) Scheduler for Handling... Deadlines?

Earliest Deadline First
- Ask everyone when need to be done by
- Serve next by deadline that must be met

Examples in Real World?
- Sometimes in long lines for airline check-in

What is good?
- Everyone finishes by when they need to

What is bad about this approach?
- Not fair: Works best when everyone works together
- Needs knowledge and trust: When is your real deadline?
- Might not be able to meet all deadlines

Use in computer systems...
- Used for “real time” applications (control systems and video)

2) Scheduler for Handling... Important Customers?

Priority-based scheduling
- Allow important customers to move to front of line

Examples in real life?
- First-class in airlines, Fast Pass at Amusement Parks

Advantages?
- Give fastest service to most important customers (make them happiest)

Disadvantages?
- Less important customers can starve
  - Might never receive service if many important customers keep arriving
  - Extreme of “unfair”
- Determining who is “important” can be difficult
  - Spend the most money? Influence the most other people? Angriest?

Computer systems
- Give higher priority to system processes
- Higher priority to “interactive” processes
3) Scheduler for Handling... Short Jobs?

"Shortest Job First" (SJF)
- Figure out which customer has shortest order
- Let shortest orders go to front of line

Examples in Real World?
- Decide to interrupt counter person with question...
- Separate lines for "10 items or less"

Advantages
- Creates optimal schedule for average waiting time
  - Minimizes average waiting time over all customers
  - Moving short job before long job:
    - Improvement in wait time of short job > Penalty to long job

Disadvantages
- Customers with many items can starve; unfair
- How can you tell how long a job will take? Incentive to lie!
  - Must solve for this to work in computer systems!
  - How???

Big Idea:
Use Past to Predict Future

Processes behave in future similarly to behavior in past (just like people?)
- Did this process use CPU for small time in the past?
- Use info to schedule short bursts
- Past doesn’t mean yesterday – refers to this execution of process

Job: Different than “application” or “process”
- Process alternates between CPU and I/O (e.g., wait for user input)
- Job is the CPU burst
- Measure past CPU bursts of one process
Details: Implementation in OS

Multi-level feedback queue
  • Multi-level queue:
    N lists of different priorities
  • Feedback:
    Move to queue based on length of last CPU burst

OS runs process at highest (priority) queue
  • Processes in same queue scheduled FCFS
  • Can also fix starvation problem by moving up processes not scheduled for awhile...

Motivation for Sharing Memory:
Cutting-Edge Libraries

Why is this relevant to sharing memory?
How to Share Memory?

Illusion provided by Operating System?
- Each process has all of physical memory to itself

Reality: Reside in physical memory at same time

Technique: Space sharing

Challenges with Sharing Memory

1) Ensure one process cannot r/w another process’s memory
   - OS and HW cooperate to implement protection mechanism
     - Translate memory references from logical to physical addresses
   - Not focus of today’s lecture
Challenges with Sharing Memory

2) Not enough physical memory for all address spaces
   • What can OS do if processes need more memory?
   • What policy should OS use to determine what is kept in memory and what is not?

Logical View of Address Space
Everything process can address thru memory including data and code

Memory Hierarchy
Leverage memory hierarchy of machine architecture
   • If doesn’t fit in one level, move to next level down

size | registers | cache | main memory | disk storage

speed | cost
Sizes of Memory Hierarchy

Your search is over.

- Brand New 20" iMac
- 4 GB RAM / 250 GB HDD
- Intel Core 2 Duo 2.0 GHz
- Mac OSX Leopard 10.5
- iLife '08

Challenge

What parts of each address space should OS keep in main physical memory vs. disk storage?
Today’s Summary

Operating System: Software that manages hardware

- Provides illusion to each process that it’s only one running
  - Context switches CPU across processes (Time share)
  - Protects memory across processes (Space share)

- Scheduling policies for CPU:
  - First-come-first-served (FCFS), Earliest-deadline-first,
    Priority-based, Shortest-Job-First (SJF)
  - Use past behavior to approx SJF: Multi-level Feedback Queue

- Caching policies for Memory:
  - Speed of fastest memory; Capacity largest memory
  - Optimal Replacement Algorithm requires knowledge of future
  - Use past to predict future (Least-Recently-Used)

Announcements

- HW 7 due before class Friday: Design of Project 2
- Working on Project 2