ESE250: Digital Audio Basics

Week 9: March 19, 2013
Operating System (Processor Sharing)
Teaser: iPhone Multiprocessing

• What does your iPhone do?
Teaser: iPhone Multiprocessing

• What does your iPhone do?
  – stores data (calendar; contacts; pictures; … )
  – analog i/o (touchscreen; music; movies; … )
  – asynchronous communications (email; text; … )
  – synchronous communications (phone)
Teaser: iPhone Multiprocessing

• What does your iPhone do?
  – stores data (calendar; contacts; pictures; … )
  – analog i/o (touchscreen; music; movies; … )
  – asynchronous communications (email; text; … )
  – synchronous communications (phone)

• How can it do this all at once?
Why multiple “simultaneous” applications?

• Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS...)?
Why multiple “simultaneous” applications?

- Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS…)?
  - To make the best use of unspent capacity
Why multiple “simultaneous” applications?

• Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS…)?
  – To make the best use of unspent capacity
  – Need to service real-time tasks while doing other things
Why multiple “simultaneous” applications?

• Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS…)?
  – To make the best use of unspent capacity
  – Need to service real-time tasks while doing other things
  – Separate UI from computation task
Why multiple “simultaneous” applications?

- Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS...)?
  - To make the best use of unspent capacity
  - Need to service real-time tasks while doing other things
  - Separate UI from computation task
  - Allow device operation to continue during a long latency task
Why multiple “simultaneous” applications?

• Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS…)?
  – To make the best use of unspent **capacity**
  – Need to service **real-time tasks** while doing other things
  – Separate **UI** from computation task
  – Allow device operation to continue during a **long latency task**
  – Preserve existing **state** of application
Why multiple “simultaneous” applications?

- Why would we want to run many applications simultaneously on our computer (or phone, or TiVo, or GPS...)?
  - To make the best use of unspent **capacity**
  - Need to service **real-time tasks** while doing other things
  - Separate **UI** from computation task
  - Allow device operation to continue during a **long latency task**
  - Preserve existing **state** of application
  - Programs waiting for **asynchronous events**
Outline

• What is an Operating System (OS)?
  – What does it do?

• Interlude

• Virtualization (sharing)
  – Processor
  – Memory (process isolation)
  – Devices
“Stored Program” Processor

• By filling in memory, can program to perform any computation
Programming the Processor

• How can we change the computation?
Programming the Processor

• How can we change the computation?
  – How do we get the bits into memory?
Programming the Processor

- How can we change the computation?
  - How do we get the bits into memory?

- What if we had to reboot machine for every application?
More than one Program?

• Could we have multiple applications?
More than one Program?

• Could we have multiple applications?
  – How would they be organised?
More than one Program?

• Could we have multiple applications?
  – How would they be organised?
  – Just run one at a time for now
Multiple Running Programs?

• How can multiple applications run simultaneously on the processor?
Coordination?

• Each program knows about every other program in the system.

• Is this necessary?
  – How much memory would it use?

• Implications?
Coordination?

- Where would it be acceptable to do this?
- Where would it be unworkable?
Coordination?

• Where would it be acceptable to do this?
  – Proprietary system with small set of applications all developed in-house.

• Where would it be unworkable?
  –
Coordination?

- Where would it be acceptable to do this?
  - Proprietary system with small set of applications all developed in-house.

- Where would it be unworkable?
  - Any upgradeable platform (e.g. laptop, iPhone)
  - Any platform integrating non-source applications from variety of sources
Role of Operating System

- Higher-level, **shared** support for all programs
  - Could put this support in each program, but most programs need similar support
  - Needs to be abstracted from program

- Resource **sharing**
  - Processor, memory, “devices” (net, printer, audio)

- Polite **sharing**
  - Isolation and protection

- **Idea:** Expensive/limited resources can be shared in time – OS manages this sharing
  - A generalization/expansion of idea from Week 8
Shared Support

• What software support do most programs need?
• Examples:
  – Memory allocation/deallocation
  – Handle I/O: keyboard/screen
  – Draw pretty boxes/menus/selections
Multiple Applications

- Simplest case: loader/dispatcher
- A program that runs other programs

Would you like to
- Edit a file?
- Play a song?
- Load a song?
- Play a game?
- Load a new application?
Application Load / App Menu
Application Selector Sketch
Application Selector Sketch

1. Copy selected program into memory
2. Clear/reserve space for it
3. Branch to start of application
   - Like procedure call
4. Upon exit (return)
   - Go back and display menu
Software to Use Device

• Consider a display
  – Low level:
    • Set pixel @ address 0x019509 to 0xFF0000
  – Might mean
    • Set pixel x=101, y=275 to red
  – Be a part of
    • Draw a red line on the screen
    • Print the character A
Decoding

- Give each character pattern a short name (1B long)
- Use to lookup the 14B bit pattern
Devices

- Displays
- Input (keyboard, mouse)
- Storage (hard drive, USB drive, CDROM)
- Network (ethernet, wifi, bluetooth)
- Microphone, speakers
- GPS
- Printer
Device Coordination

• Coordinate among multiple users
  – Don’t want programs accessing hardware directly (ignorant of other users)
  – OS handles access to devices

• Exclusively allocate to one application at a time – or allow interleaved use?
  – Speaker?
  – Printer?
  – Network?
  – Screen? (portion of screen?)
  – Hard disk?
  – Radio?
Interlude

iPhone App Store
Jailbreak
App Store

- Launched July 2008
  - Click-to-buy-and-install applications
  - Direct sales from developers (Apple only middleman)
- Lightweight path from developer (innovation) to consumer
- Over 40 Billion Apps sold
- 160,000+ developers
- Top 20 Apps
  - $350K—800K to developers

- Easy User Installation
- Apple Controls Content
  + Ensure positive user experience
  + Ensure reliable operation
  - Limit competition
  - Limit what you can do
Jailbreak iPhone

• To bypass Apple software control
• Gives access to applications and uses Apple not approve
• Drawn-out legal dispute
  – Apple claims violates DMCA
  – July 2010: declared legal
  – January 2013: declared illegal
• Will certainly void your Apple Warranty
• **One impact:** exposes more multiprocessing capabilities
Virtualization
Virtualization

• We have seen how OS works as an agent between software and hardware resources
• Provides an abstract view separate from the physical view
• Hides physical view
• Provides abstract view to software
  – Abstract from physical resource limits
Big Idea

• Virtualize the processor!
  – Make it look like we have multiple processors
  – With each program running on its own processor

• Abstract
  – the machine seen by the program from the physical machine
Terminology: Process

• Process
  – A *virtualization* of the physical processor
    • an instance of a program in execution
  – Virtual processor
What does our program see?

• Physically
  – One processor
    • One PC
    • One data memory
    • One instruction memory
  – These are its state
    • Terminology: context
Executing the Program

• To execute program
  – Keep track of state of machine
    1. Value of counter (Program counter)
    2. Contents of instruction memory
    3. Contents of data memory
Execution Exercise

• Simulate one of the 3 cases (as indicated on your worksheet) for the 12 cycles shown.
Execution: Getting Started
Execution Exercise

• Simulate one of the 3 cases (as indicated on your worksheet) for the 12 cycles shown.
One Processor, One Program

• On the physical machine, can only run one program
  – Why?
    • One PC
    • One memory
Virtualization

• We have seen how OS works as an agent between software and hardware resources
• Provides an abstract view separate from the physical view
• Hides physical view
• Provides abstract view to software
  – Abstract from physical resource limits
Virtualization

- We have seen how OS works as an agent between software and hardware resources
- Provides an abstract view separate from the physical view
- Hides physical view
- Provides abstract view to software
  - Abstract from physical resource limits
Key Idea

- Can capture **state** of a processor
  - All the information that defines the current point in the computation
Executing the Program

- To execute program
  - Keep track of state of machine
    1. Value of counter (Program counter)
    2. Contents of instruction memory
    3. Contents of data memory
Key Idea

• Can capture **state** of a processor
  − All the information that defines the current point in the computation
State in Memory

- word
- firefox
- media play
- java
Sharing Processor

• Now that we can save/restore the state
• Can share processor among processes
  (Restore state; run for time; save state)

• **Isolation:** none of the processes need to know about each other
  – Each thinks it has the a whole machine
Memory?

• “Save all of memory”?  
  – Must have more memory  
  – Enough to hold all the memory of all the running programs == all the processes  
• Each program perceives itself to own the entire machine  
  – Each may put program in same place?  
  – Shouldn’t have to know about other programs, where they use memory
Saving Memory?

• Each program has view it owns machine
  – Each may put program in same place?
  – Shouldn’t have to know about other programs, where their stacks are…

• Could:
  – Have programs operate 0…max_process_mem
  – Copy data in and out of this range
  – Keep elsewhere
    • more memory not visible to program
    • On disk
Memory Save/Restore

normal process sees

ESE250 S'13: DeHon, Kadric, Kod, Wilson-Shah
Memory Save/Restore

- word
- firefox
- media play
- java

normal process sees
Memory Save/Restore

normal process sees

word  firefox  media play  java

ESE250 S'13: DeHon, Kadric, Kod, Wilson-Shah  Week 9 – Operating System
Memory Save/Restore

word

firefox

media play

java

normal process sees
Memory Save/Restore

- Normal process sees
- word
- firefox
- media play
- java
Saving Memory?

• Each program has view it owns machine
  – Each may put program in same place?
  – Shouldn’t have to know about other programs, where their stacks are…

• Could:
  – Have programs operate 0…max_process_mem
  – Copy data in and out of this range
  – Keep elsewhere
    • more memory not visible to program
    • On disk
Management Program

• Need another program / process:
  – Manage swap of running processes
  – Decide what to run next
  – Decide when to stop a process
  – “Meta-program.”

• ...process manager/scheduler
Time-Sliced Sharing

• Simplest version:
  - Run each process for 10,000 cycles
  - Then swap to next process
  - Looks like each process runs on a processor 1/n-th the speed of the real processor

• More sophisticated:
  - Assign uneven time to processes
  - Also change when process waits for input
  - What are cases where this is appropriate?
Time Switch Exercise

• Write down your +6 cycle state on the swap sheet
  – Make sure to also write down your IMEM designator
• Want to exchange A->B, B->C, C->A
• Pass to right.
• Keep when you get the right target.
  – Pass through if you’re already satisfied.
Time Switch Exercise

• Write down your +6 cycle state on the swap sheet
  – Make sure to also write down your IMEM designator
• Want to exchange A->B, B->C, C->A
• Pass to right.
• Keep when you get the right target.
  – Pass through if you’re already satisfied.
Time Switch Exercise Demo

Simulating a case:

• Processor runs A for 6 cycles
  – Then stores off to memory.
• Processor runs B for 6 cycles
  – Then stores off to memory
• Processor reads A state from memory and runs for another 6 cycles
• Processor reads B state from memory and runs for another 6 cycles
Device Virtualization

• Similar concept
  – Identify state of device
  – Save/restore state as use “virtual” device
• Window as virtualization for screen
  – May not even be visible (e.g. minimized)
Review: Key Idea

• Can capture **state** of a processor
  – All the information that defines the current point in the computation
  – *i.e.* program counter, data and instruction memory…
• Can save that in memory
  – A different memory from what the process sees
  – (could be different range of addresses)
• Fully represents the running program
• Can restore that from memory to the processor
• Can save/restore without affecting the **functional** behavior of the program
Big Ideas

• Virtualize hardware
  – Identify state; save/restore from memory

• **Program view:** owns complete machine

• Allows programs to share limited physical hardware (e.g. processor)
  – Provide illusion of unlimited hardware

• **Operating System** is the program that manages this sharing
Learn More

• Courses
  – CIS380 – operating systems
ESE250: Digital Audio Basics

Week 9: March 19, 2013
Operating System
(Processor Sharing)