CSE 30341
Operating System Principles

Overview/Introduction

Syllabus

• Instructor:
  – Christian Poellabauer (cpoellab@nd.edu)
• Course Meetings
  – TR 9:30 – 10:45
  – DeBartolo 101
• TAs:
  – Jian Yang, Josh Siva, Qiyu Zhi, Louis Daudet (office hours & locations will be announced)
• Office Hours
  – Tuesday 12-1 & Thursday 11-12 or send email
  – Course web site, announcements
Textbook

• Course Textbook (not required)

Kernigan / Ritchie for C Programming is optional

Grading

• Projects 50%
• Exam 1 15%
• Exam 2 15%
• Final Exam 20%

Fixed grading scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90-100</td>
</tr>
<tr>
<td>A-</td>
<td>88-90</td>
</tr>
<tr>
<td>B+</td>
<td>86-88</td>
</tr>
<tr>
<td>B</td>
<td>76-86</td>
</tr>
<tr>
<td>B-</td>
<td>74-76</td>
</tr>
<tr>
<td>C</td>
<td>72-74</td>
</tr>
<tr>
<td>C-</td>
<td>60-62</td>
</tr>
<tr>
<td>D</td>
<td>50-60</td>
</tr>
<tr>
<td>F</td>
<td>&lt; 50 OR &lt; 40 Projects OR &lt; 40 Exams</td>
</tr>
</tbody>
</table>
Projects

- Done individually
- Four projects
- Multiple weeks each
- May have evaluation component
- Linux computers
  - Fitzpatrick
  - Stinson/Remick
- Collaboration/Honor Policy

What is an Operating System?

- A program that acts as an **intermediary between a user of a computer and the computer hardware**

- Operating system goals
  - Execute user programs
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner
  - Hide complexities from user (“layer of abstraction”)
Computer System Structure

- Four components
  - **Hardware** – provides basic computing resources
    - CPU, memory, I/O devices
  - **Operating system**
    - Controls and coordinates use of hardware among various applications and users
  - **Application programs**
    - Word processors, compilers, web browsers, database systems, video games
  - **Users**
    - People, machines, other computers
What is an OS? (System View)

• OS is a **resource allocator**
  – Manages all resources
  – Decides between conflicting requests for efficient and fair resource use

• OS is a **control program**
  – Controls execution of programs to prevent errors and improper use of the computer

What is an OS? (User View)

• Users want **convenience, ease of use, performance, ...**
  – Don’t really care about details of how applications are executed

• Users needs may depend on type of computer:
  – Shared computer
    • Mainframes, any kind of centralized computing systems
    • “Am I getting my fair share?”
  – Dedicated computer (workstation, PC, laptop)
    • Do I get good performance? Is it easy to use?
  – Mobile devices (smartphones, etc.)
    • Do I get good performance? Will my battery last?
  – Embedded systems
    • Does it work correctly? Is it predictable? *(Do we even need an OS?)*
What Is “Inside” An OS?

• No universally accepted definition

• One extreme:
  – “Everything a vendor ships when you order an operating system” (everything on the CD/DVD)

• Another extreme:
  – Minimum necessary functionality (e.g., the “kernel”)
  – Kernel: the one program running at all times on the computer (without it, you wouldn’t have an OS)
  – Everything else is either a system program (ships with the operating system) or an application program (bought, downloaded, programmed, …)

What Is “Inside” An OS?

• Inside kernel: “essential” parts
  – CPU scheduler, memory manager, interrupt controller, etc.

• Outside kernel: “non-essential” parts
  – File management commands, user management & access right commands, performance tools, debugging, compilers, …
Summary

• “Glue” between hardware and “user space”
  – User space: application-level, using an OS
  – Kernel space: within the operating system
• User perspective: provide a layer of abstraction; convenience, ease-of-use, etc.
• System perspective: manage/control limited resources, efficiency, support multiple users/application

• Kernel (essential parts) + systems programs

Computer System Organization

• One or more CPUs, device controllers connect through common bus providing access to shared memory
• Concurrent execution of CPUs and devices competing for memory, bus, etc.
Computer Startup

- **Bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution (Unix: 1st process = swapper/sched, 2nd process = init)

---

BIOS

Basic Input / Output System
CPU & I/O

• I/O devices and the CPU can execute concurrently
• Multiple things going on at once
• How?
  – Each device has its own controller (“mini CPU”)
  – Each controller responsible for particular device type
  – Each controller has some local memory (buffer)
• Need way to move data
  – CPU moves data from/to main memory to/from local buffers
  – I/O is from the device to local buffer of controller
  – Device controller needs to informs CPU that it has finished its operation

Polling vs. Interrupts

• Think of a set of devices
• Polling
  – Are you done?
  – Are you done?
  – Are you done?
  – Are you done?
• Interrupt
  – I need HELP!

Ferris Bueller’s “Day Off”
Common Functions of Interrupts

- **Interrupt occurs**
  - Transfer to *interrupt service routine (ISR)*
  - **Interrupt vector**
    - Addresses of the service routines
  - Save & come back
    - Stop our current assembly instruction
    - Come back to resume after we are done (remember where!)
- **Multiple types of interrupts**
  - **Hardware interrupts**
  - **Trap or exception** -> software-generated interrupt
  - Errors or user requests
- An operating system is **interrupt driven!**

Interrupt Handling

- The operating system preserves the state of the CPU by storing **registers** and the **program counter**
- Incoming interrupts are disabled (prevent lost interrupts)
- Which device?
  - Polling (no identity known)
  - Vectored (sends identity along)
- Separate segments of code determine what action should be taken for each type of interrupt
Lecture Wrap Up

• Key Points
  – What is an operating system?
  – What is a kernel?
  – What is an interrupt and how does it work?