Operating Systems
Overview

Instructor: Kartik Gopalan

Class website:
http://oscourse.github.io
What is an Operating System?
What is an Operating System?

- A bunch of software and data residing somewhere in memory.
  - But it's not just *any* software.
- OS is the *most privileged* software in a computer.
  - *Privileged* means that OS can do special things, like write to disk, talk over the network, control memory and CPU usage, etc.
- OS manages all system resources
  - CPU, Memory, and I/O devices
Why do we need an OS?

- Program 1
  - (Software)
  - Instruction Set Architecture (ISA)
- Hardware
Why do we need an OS?

But the program doesn’t know how to access hardware devices for input/output (I/O)
Why do we need an OS?

But what if two programs need to share the hardware?

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Program 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>HW Multiplexing (Sharing)</td>
<td>Instruction Set Architecture (ISA)</td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
</tr>
</tbody>
</table>
Why do we need an OS?

But what if the two programs don’t trust each other?

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Program 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Sharing</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Protection</td>
<td></td>
</tr>
</tbody>
</table>

Instruction Set Architecture (ISA)

Hardware
But when does the OS “run”? 
But when does the OS “run”? 

Four ways to invoke OS code

- (a) System Calls
- (b) Exceptions
- (c) Interrupts
- (d) Kernel Threads
Interfaces in a Computer System

ISA = Instruction Set Architecture
ABI = Application Binary Interface
API = Application Programming Interface
History of OS

- 1950s and 1960s: Early operating systems were simple batch processing systems
  - Users provided their own “OS” as libraries.
- 1960s and 1970s: Multi-programming on mainframes
  - Concurrency, memory protection, Kernel mode, system calls, hardware privilege levels, trap handling
  - Earliest Multics hardware and OS on IBM mainframes
  - Which led to the first UNIX OS which pioneered file systems, shell, pipes, and the C language.
- 1980s: Personal computing era
  - MacOS, IBM PC and its DOS, Windows, and so forth.
  - Unfortunately, many lessons from earlier multi-programming era were forgotten and had to be re-learned (painfully).
- 1980s also saw the fragmentation of UNIX
  - Each big company had its own version (IBM, Apple, HP, SUN, SGI, NCR, AT&T….)
  - LOT of legal wrangling over IP and copyrights
- 1990s: Then came BSD and Linux
  - Open source.
  - Led the way to modern OSes and cloud platforms
- 1990s also saw wider adoption of threads and parallelism
- 2000 and beyond: Mobile device OS and hypervisors
  - Android, iOS
  - VMWare ESX, Xen, Linux/KVM etc.