A little historical perspective on OS.

What are operating systems interesting and important?

What is an operating system (OS)?

Introduction and History of Operating Systems

Prerequisite & Course Sign Up (handout)

Course Organization & Outline (handout)

Today: Introduction to Operating Systems
Strict late policies and polices on cheating

Programming assignments will use Java

Course Requirements:

(Select at least one of the following)

Textbook: *Applied Operating System Concepts Iled* and *Operating Systems* by Silberschatz, Galvin, Gagne

Course Organization:

Enrollment policy

Course is at capacity
Introduction to Operating Systems

Office Hours:
- Instructor: Tu-Thu: 2:30-3:30, CS 336 or by appnt

Miscellaneous:
- Office Hours
- Discussion section to help you with lab assignments
- Accounts in the Ed-Lab: 30+ Linux-based PCs
**User Applications**

Virtual Machine Interface

Physical Machine Interface

- **Goal:** Design an OS so that the machine is *convenient* to use for software engineering problems and *efficient*.

  - Examples: concurrency, memory protection, networking, and security.

  - Achieve fairness and efficiency (throughput).

**Coordination:** The OS coordinates multiple applications and users to

- **Time-sharing:**

  - Example: the system, virtual memory, networking, CPU scheduling, and

**Services:** The OS provides standard services (the interface) which the

---

**Operating System: Salient Features**

---

**What is an Operating System?**
Intersect without understanding the components. Obviously, you cannot understand the implications of how components
classes as in encapsulation mechanism, etc.) and I/O systems, etc.) and solid programming skills (complex data structures,
background: To understand this course you must have a solid basic
apply elsewhere.

Not many operating systems are under development so you are unlikely to get a job building an OS. However, understanding operating systems will

Software meet.
• System Interaction Point: The OS is the point where hardware and
  application programs to work at all.
• Basic Understanding: The OS provides the services that allow
  • As systems change the OS must adapt (e.g., new hardware, software).
  • Putting functionality in hardware or software.
  • Performance and simplicity of OS design.
  • Performance and the convenience of OS abstractions.
• System Design: How to make tradeoffs between
  memory, CPU’s, resources, world wide computing, etc.
• Abstraction: How to get the OS to give users an illusion of infinite

Why Study Operating Systems?

Why Study Operating Systems?
OS design and development is a science

- OS/360 released with 1000 known bugs
- Multics announced in 1963, released in 1969
- First OS failures
- Decides which processes to resume when one gives up the CPU
- Protects one program's memory from other programs
- Decides which processes to start
- OS manages interactions between concurrent programs
- OS runs until it performs I/O, then another job gets the CPU

4. Multiprogramming: Several programs run at the same time, sharing the machine, i.e., I/O and CPU processing overlap.

Phase I: Hardware is very expensive, humans are cheap

| History of Operating Systems |

| Hardware is very expensive, humans are cheap |

Phase I: Hardware is very expensive, humans are cheap
now we want to share across machines

Why? Distributed computing & networking - we still want to share resources, but

Did not really work... LINUX is splitting all this functionality back into OS

multi-processing, concurrency, and protection

Idea was to make the OS simple (again) by getting rid of support for

6. Personal computing: computers are cheap, so put one in each terminal

Phase 3: Hardware is very cheap, humans are expensive

History of Operating Systems

Department of Computer Science, VJays Ambast

Phase 2: Hardware is cheap, humans are expensive

History of Operating Systems

Department of Computer Science, VJays Ambast
Magnitude change in almost every computer system component.

- From 1953 to now (the 40 year history of computing), 9 orders of magnitude change.

<table>
<thead>
<tr>
<th></th>
<th>1999</th>
<th>1983</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 MByte</td>
<td>32 MByte</td>
<td>Store</td>
</tr>
<tr>
<td>10 MB/ps</td>
<td>1 GB/ps</td>
<td>Network</td>
</tr>
<tr>
<td>$500</td>
<td>$100'000</td>
<td>Memory</td>
</tr>
<tr>
<td>0.5 MIP</td>
<td>0.5 MIP</td>
<td>Price/MIP</td>
</tr>
</tbody>
</table>

Example

- Change is one of the defining forces in computer science.

Batch processing was right for its time, but not anymore.

History Lesson

8. Real-time systems allow computers to control physical machines or resources.

- Soft real-time OSs allow deadlines to be missed.
- Hard real-time OSs must meet stringent requirements.
- Timing requirements are de-coupled when tasks must be accomplished.

- Provide high-quality interaction so as in virtual reality.

7. Parallel and distributed computing allow multiple processors to share resources.

- Hardware is very cheap, processing demands are increasing.

History of Operating Systems
transportation to nearly the speed of light - 7 orders of magnitude.

and fax (text & pictures), communication went from the speed of
communication – at the invention of the telephone (voice), TV (video)

magnitude.

(10 miles/hour) to the Concorde (1500 miles/hour) - 2 orders of
transportation – over the last 200 years, we have gone from horseback

Examples:

This degree of change has no counterpart in any other area of business.

History Lesson