Operating System

Introduction
(Ch 1)

Let’s Get Started!
• What are some OSes you know?
  – Guess if you are not sure
• Pick an OS you know:
  – What are some things you like about it?
  – What are some things you don’t like about it?

What is an Operating System?
• An Extended Machine (Top-down)
  – Transforming - new resource
    • ex: Win98 device manager
• A Resource Manager (Bottom-up)
  – Multiplexing - illusion of several resources
    • ex: browse the web AND read email
  – Scheduling - deciding who gets what when
    • ex: compile fast OR edit fast
• Why have an OS?
  – Convenient and Efficient
    • Programming hardware difficult
    • Idle hardware “wasteful”
Where in the Book are we?

- Chapter 1
  - 1.1 overview (done)
  - 1.2 history (next)
  - 1.3 overview (read on your own)
  - 1.4 hardware (review on your own, as needed)
  - 1.5 concepts
  - 1.6 structure

OS History

- Helps understand key requirements
  - Not one brilliant design
    - (despite what Gates or Torvalds might say)
  - Fixed previous problems, added new ones
  - Tradeoffs
- Closely tied to:
  - Hardware history
  - User history

Hardware History

<table>
<thead>
<tr>
<th></th>
<th>1981</th>
<th>1999</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>1</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>$/Power</td>
<td>$100K</td>
<td>$45</td>
<td>2200</td>
</tr>
<tr>
<td>Memory</td>
<td>128K</td>
<td>128M</td>
<td>1000</td>
</tr>
<tr>
<td>Disk Capacity</td>
<td>10M</td>
<td>10G</td>
<td>1000</td>
</tr>
<tr>
<td>Net Bandwidth</td>
<td>9600b/s</td>
<td>155Mb/s</td>
<td>15K</td>
</tr>
<tr>
<td>Users / Mach.</td>
<td>10s</td>
<td>&lt;=1</td>
<td>10</td>
</tr>
</tbody>
</table>

- Comments? Change!

Hardware Very Expensive
Humans Cheap

- Single program execution (no OS)
- Hardwire “programming”
- Programming slow, not “offline”!
  - Punch cards

Hardware Very Expensive
Humans Cheap

- Punch cards
- Fortran or assembler
- Waste computer time walking!
  - Batch programs on tape
Hardware Very Expensive
Humans Cheap
- Programs read in from tape
- Two applications:
  - Scientific
  - Data processing
- CPU idle during I/O!
  - Multiprogramming with partitions
  - Spooling as jobs finished

Hardware is Cheap
Humans Expensive
- Turn around time 1/2 day
- Programmer time wasted!
  - “Sigh. In the good old days….”
  - Time-sharing
  - Multics (sorta)
  - New problems
    - response time
    - thrashing
    - file-systems

Hardware Very Cheap
Humans Very Expensive
- Personal computers
  - Network operating systems
  - Distributed operating systems
- OSes today
  - small == 1000K
  - large == 10,000K
  - need to evolve quickly
    - hardware upgrades, new user services, bug fixes
  - efficient and/or modular kernels

Windows NT/2000 History
- 1988, v1
  - split from joint work with IBM OS/2
  - Win32 API
- 1990, v3.1
  - Server and Workstation versions
- 1997(?), v4
  - Win95 interface
  - Graphics to kernel
  - More NT licenses sold than all Unix combined

Windows NT/2000 Today
- Microsoft has 80% to 90% of OS market
  - mostly PC’s
- 800 MHz Intel Pentium
- NT aiming at robust, server market
  - network, web and database
- Platforms
  - Intel 386+ only
- NT is 12,000,000 lines of code
- 2000 is 18,000,000 lines of code
Linux History

- Open Source
  - Release Early, Release Often, Delegate
  - “The Cathedral or the Bazaar”
- Bday 1991, Linus Torvalds, 80386 processor
  - v.01, limited devices, no networking,
  - with proper Unix process support!
- 1994, v1.0
  - networking (Internet)
  - enhanced file system (over Minix)
  - many devices, dynamic kernel modules

Linux History

- Development convention
  - Odd numbered minor versions “development”
  - Even numbered minor versions “stable”
- 1995, v1.2
  - more hardware
  - 8086 mode (DOS emulation) included
  - Sparc, Alpha, MIPS support started
- 1996, v2.0
  - multiple architectures, multiple processors
  - threads, memory management ....

Linux Today

- v2.4
- 3,000,000 lines of code
- 7-10 million users
- Estimated growth 25%/year through 2003
  - all others, 10% combined

Outline

- Operating System Concepts
  - Processes
  - Memory management
  - Input/Output
  - Files
  - System Calls
  - Shells
- Operating System Structure
  - Simple Systems
  - Virtual Machines
  - Micro Kernels

The Process

- Program in execution
- Running -> Suspended -> Running
- Example: the Shell
- Process “Tree”
- Signals
- UID (GID)
- (Two weeks)

Memory Management

- One chunk of physical memory
- Needs to be shared with all processes
  - multiprocessing
- 32 bit architecture, 2^32 bytes 4GB!
  - virtual memory
- (Two weeks)
Input/Output
- OS manage resources, including other devices
- Significant fraction of code
  - Up to 90%
- Want to be simple to use
- (2 days)

Files
- Store data on disk
- Directory “Tree”
- Working directory
- Protection bits
  - 9 in Unix: rwx bits, ex: rwxr-xr--x
- Abstraction of I/O device
  - terminal, printer, network, modem
- Pipe
- (1 day)

System Calls
- Way processes communicate with OS
- example: `write(file, string, size)`
- OS specific!
- POSIX (1980s)
  - Portable Operating System (UNIX-ish)
- (Most of the projects use them)
- (One of the projects will add system calls)

Shells
- User’s interface to OS
- Simple commands
  - “cd”, “cat”, “top”
- Modifiers
  - ‘&’, ‘|’, ‘>’
- (Hey, do some process and shell examples!)

Outline
- Operating System Structure
  - Simple Systems
  - Virtual Machines
  - Micro Kernels
Simple Systems

- Unix (see `/vmunix`)
  - Applications
  - Signals, File Sys, Swapping, Scheduling ...
  - Terminal, Device, Memory

- “The Big Mess”
- Some move towards a more modular kernel

Virtual Machines

- IBM VM/370 → VMWare
  - Complete protection
  - OS development, emulation
  - Performance!
  - (Exokernel says can have subset of kernel, 1.7.4)

Virtual Machines

- Java Virtual Machine
  - Java program
  - Java OS
  - Java VM
  - Operating System
  - Hardware

- Platform independence!

Micro Kernel

- Mach
  - User Process
  - File Server
  - Mem Server
  - Kernel

- Client-Server
- Good performance
- Adaptable to distributed OS
- Robust
- Careful about mechanism!

WinNT/2000 Structure

- User Level Space
  - Netscape
  - Win32 Subsystem
  - File System
- Executive / Privileged Space
  - Security
  - I/O
- Kernel Space
  - Scheduler
  - Memory Manager
  - IPC

“Micro Kernel?”

Linux Structure

- “Simple” system
  - Applications, User Space
  - System Libraries
  - Kernel
  - ppp
  - systemd
  - Terminal, Device, Memory

- Loadable Modules
  - done after “boot”
  - allow 3rd party vendors
  - easier for development