What is an Operating System?

August 23rd, 2017

Prof. Ion Stoica

http://cs162.eecs.Berkeley.edu
Who am I?

• Ion Stoica
  – 465 Soda Hall (RISE Lab)
  – Web: http://www.cs.berkeley.edu/~istoica/
  – Office hours: Mondays 11-12pm, and Wednesdays 12-1pm in 465F Soda

• Research areas:
  – ML systems (Clipper, Ray, Pywren, …)
  – Big Data systems (Apache Spark, Succinct, …)
  – Previous: Cloud computing (Apache Mesos, Alluxio), Peer-to-Peer networking (Chord), Networking QoS
“The Magnificent Seven” - CS162 TAs

Michael Do  Anurag Khandelwal  Qifan Pu  Stephanie Wang

Neeraja Yadwadkar  Patrick Yang  Justin Yum
This and Next Week

• Sections Tuesdays/Wednesdays – attend any section you want
  – We’ll assign permanent sections after forming project groups
  – This week will help us determine the section balance

• This is an Early Drop Deadline course (September 1)
  – If you are not serious about taking, please drop early
  – Dept will continue to admit students as other students drop

• On the waitlist ???
  – Unfortunately, we maxed out sections and this room capacity
Goals for Today

• What is an Operating System?
  – And – what is it not?
• What makes Operating Systems so exciting?
• Oh, and “How does this class operate?”

Interactive is important!
  Ask Questions!

What is an operating system?

• Special layer of software that provides application software access to hardware resources
  – Convenient abstraction of complex hardware devices
  – Protected access to shared resources
  – Security and authentication
  – Communication amongst logical entities
What Does an OS do?

• Provide abstractions to apps
  – File systems
  – Processes, threads
  – VM, containers
  – Naming system
  – …

• Manage resources:
  – Memory, CPU, storage, …

• Achieves the above by implementing specific algos and techniques:
  – Scheduling
  – Concurrency
  – Transactions
  – Security
  – …
OS Basics: “Virtual Machine” Boundary

- Hardware:
  - Processor
  - Storage
  - Networks
  - Displays
  - Inputs

- Software:
  - OS Hardware Virtualization
  - Instruction Set Architecture (ISA)
  - Memory
  - Processes
  - Address Spaces
  - Windows
  - Sockets
  - Files

- Additional:
  - Threads
  - Address Spaces
OS Basics: Program => Process

- Processor
- Memory
- OS
- ISA
- Storage
- Networks
- Displays
- Inputs

Hardware Virtualization

- Threads
- Address Spaces
- Processes
- Files
- Windows
- Sockets
OS Basics: Context Switch

OS Hardware Virtualization

Hardware
- Processor
- ISA
- Storage

Software
- Threads
- Address Spaces
- Processes
- Windows
- Files
- Sockets

Memory
- OS

Networks
- Displays

Inputs
OS Basics: Scheduling, Protection

Software

OS Hardware Virtualization

Hardware

Memory

Protection Boundary

Processor

Storage

Networks

Displays

Inputs

Threads

Address Spaces

Processes

Windows

Sockets

OS

Files

Windows

Sockets
OS Basics: Loading

Hardware

- Processor
- Controller
- Storage

Software

- Threads
- Address Spaces
- Processes
- Windows
- Sockets
- Memory
- Files

Protection Boundary

Networks

Displays

Inputs
What makes Operating Systems Exciting and Challenging?
Technology Trends: Moore’s Law

Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months. Called “Moore’s Law”

Microprocessors have become smaller, denser, and more powerful.

2X transistors/Chip Every 1.5 years
New Challenge: Slowdown in Joy’s law of Performance


⇒ Sea change in chip design: multiple “cores” or processors per chip

- VAX: 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2002
- RISC + x86: ??%/year 2002 to present
Another Challenge: Power Density

- Moore’s law extrapolation
  - Potential power density reaching amazing levels!
- Flip side: battery life very important
  - Moore’s law can yield more functionality at equivalent (or less) total energy consumption

Source: S. Borkar (Intel)
• Today: multiple CPUs/person!
  – Approaching 100s?
ManyCore Chips: The future is here

- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
  - 24 “tiles” with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers
  - Hardware support for message-passing

- Amazon X1 instances
  - 128 virtual cores, 2 TB RAM
- How to program these?
  - Use 2 CPUs for video/audio
  - Use 1 for word processor, 1 for browser
  - 76 for virus checking???
- Parallelism must be exploited at all levels
The End of Moore’s Law…

- Moore’s Law has (officially) ended -- Feb 2016
  - No longer getting 2 x transistors/chip every 18 months…
  - or even every 24 months
- May have only 2-3 smallest geometry fabrication plants left:
  - Intel and Samsung and/or TSMC
- Vendors moving to 3D stacked chips
  - More layers in old geometries

- Retail hard disk capacity in GB
Network Capacity

Challenge: Complexity

• Applications consisting of...
  – … a variety of software modules that …
  – … run on a variety of devices (machines) that
    » … implement different hardware architectures
    » … run competing applications
    » … fail in unexpected ways
    » … can be under a variety of attacks

• Not feasible to test software for all possible environments and combinations of components and devices
  – The question is not whether there are bugs but how serious are the bugs!
A Modern Processor: Intel Sandy Bridge

- **Package:** LGA 1155
  - 1155 pins
  - 95W design envelope
- **Cache:**
  - L1: 32K Inst, 32K Data (3 clock access)
  - L2: 256K (8 clock access)
  - Shared L3: 3MB – 20MB (not out yet)
- **Transistor count:**
  - 504 Million (2 cores, 3MB L3)
  - 2.27 Billion (8 cores, 20MB L3)
- Note that ring bus is on high metal layers – above the Shared L3 Cache
HW Functionality comes with great complexity!

Intel Sandy Bridge I/O Configuration

Proc
Caches
Memory
Busses
adapters
Controllers
I/O Devices:
Disks
Displays
Keyboards

networks
Increasing Software Complexity

From MIT's 6.033 course
Example: Some Mars Rover ("Pathfinder") Requirements

- Pathfinder hardware limitations/complexity:
  - 20Mhz processor, 128MB of DRAM, VxWorks OS
  - cameras, scientific instruments, batteries, solar panels, and locomotion equipment
  - Many independent processes work together
- Can’t hit reset button very easily!
  - Must reboot itself if necessary
  - Must always be able to receive commands from Earth
- Individual Programs must not interfere
  - Suppose the MUT (Martian Universal Translator Module) buggy
  - Better not crash antenna positioning software!
- Further, all software may crash occasionally
  - Automatic restart with diagnostics sent to Earth
  - Periodic checkpoint of results saved?
- Certain functions time critical:
  - Need to stop before hitting something
  - Must track orbit of Earth for communication
- A lot of similarity with the Internet of Things?
  - Complexity, QoS, Inaccessibility, Power limitations …?
How do we tame complexity?

• Every piece of computer hardware different
  – Different CPU
    » Pentium, PowerPC, ColdFire, ARM, MIPS
  – Different amounts of memory, disk, …
  – Different types of devices
    » Mice, Keyboards, Sensors, Cameras, Fingerprint readers
  – Different networking environment
    » Cable, DSL, Wireless, Firewalls,…

• Questions:
  – Does the programmer need to write a single program that performs many independent activities?
  – Does every program have to be altered for every piece of hardware?
  – Does a faulty program crash everything?
  – Does every program have access to all hardware?
## OS Tool: Virtual Machine Abstraction

### Application

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Virtual Machine Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Physical Machine Interface</td>
</tr>
</tbody>
</table>

- **Software Engineering Problem:**
  - Turn hardware/software quirks ⇒ what programmers want/need
  - Optimize for convenience, utilization, security, reliability, etc…
- **For any OS area (e.g. file systems, virtual memory, networking, scheduling):**
  - What’s the hardware interface? (physical reality)
  - What’s the application interface? (nicer abstraction)
Virtual Machines

• Software emulation of an abstract machine
  – Give programs illusion they own the machine
  – Make it look like hardware has features you want

• Two types of “Virtual Machine”s
  – Process VM: supports the execution of a single program; this functionality typically provided by OS
  – System VM: supports the execution of an entire OS and its applications (e.g., VMWare Fusion, Virtual box, Parallels Desktop, Xen)
Process VMs

• Programming simplicity
  – Each process thinks it has all memory/CPU time
  – Each process thinks it owns all devices
  – Different devices appear to have same high level interface
  – Device interfaces more powerful than raw hardware
    » Bitmapped display ⇒ windowing system
    » Ethernet card ⇒ reliable, ordered, networking (TCP/IP)

• Fault Isolation
  – Processes unable to directly impact other processes
  – Bugs cannot crash whole machine

• Protection and Portability
  – Java interface safe and stable across many platforms
System Virtual Machines: Layers of OSs

- Useful for OS development
  - When OS crashes, restricted to one VM
  - Can aid testing programs on other OSs
5 min break
Greatest Artifact of Human Civilization...
Internet Scale: Over 3.8 Billion Users!

- ARPANet: 1969
- RFC 675 TCP/IP: 1974
- Internet: 1990
- WWW: 2.0 B 1/26/11
- % of world's population: 38%
- 3.8 B
- 2017

8/23/17 Stoica CS162 © UCB Fall 2017 Lec 1.35
## Internet Scale: Over 3.8 Billion Users!

### World Internet Usage and Population Statistics

**June 30, 2017 - Update**

<table>
<thead>
<tr>
<th>World Regions</th>
<th>Population (2017 Est.)</th>
<th>Population % of World</th>
<th>Internet Users 30 June 2017</th>
<th>Penetration Rate (% Pop.)</th>
<th>Growth 2000-2017</th>
<th>Internet Users %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>1,246,504,865</td>
<td>16.6 %</td>
<td>388,104,452</td>
<td>31.1 %</td>
<td>8,497.0 %</td>
<td>10.1 %</td>
</tr>
<tr>
<td>Asia</td>
<td>4,148,177,672</td>
<td>55.2 %</td>
<td>1,909,408,707</td>
<td>46.0 %</td>
<td>1,570.5 %</td>
<td>49.8 %</td>
</tr>
<tr>
<td>Europe</td>
<td>822,710,362</td>
<td>10.9 %</td>
<td>650,558,113</td>
<td>79.1 %</td>
<td>519.0 %</td>
<td>17.0 %</td>
</tr>
<tr>
<td>Latin America / Caribbean</td>
<td>647,604,645</td>
<td>8.6 %</td>
<td>392,215,155</td>
<td>60.6 %</td>
<td>2,070.7 %</td>
<td>10.2 %</td>
</tr>
<tr>
<td>Middle East</td>
<td>250,327,574</td>
<td>3.3 %</td>
<td>146,972,123</td>
<td>58.7 %</td>
<td>4,374.3 %</td>
<td>3.8 %</td>
</tr>
<tr>
<td>North America</td>
<td>363,224,006</td>
<td>4.8 %</td>
<td>320,059,368</td>
<td>88.1 %</td>
<td>196.1 %</td>
<td>8.3 %</td>
</tr>
<tr>
<td>Oceania / Australia</td>
<td>40,479,846</td>
<td>0.5 %</td>
<td>28,180,356</td>
<td>69.6 %</td>
<td>269.8 %</td>
<td>0.7 %</td>
</tr>
<tr>
<td><strong>WORLD TOTAL</strong></td>
<td><strong>7,519,028,970</strong></td>
<td><strong>100.0 %</strong></td>
<td><strong>3,835,498,274</strong></td>
<td><strong>51.0 %</strong></td>
<td><strong>962.5 %</strong></td>
<td><strong>100.0 %</strong></td>
</tr>
</tbody>
</table>

**NOTES:**
2. CLICK on each world region name for detailed regional usage information.
3. Demographic (Population) numbers are based on data from the United Nations Population Division.
4. Internet usage information comes from data published by Nielsen Online, by ITU, the International Telecommunications Union, by GfK, by local ICT Regulators and other reliable sources.
5. For definitions, navigation help and disclaimers, please refer to the Website Surfing Guide.
6. Information from this site may be cited, giving the due credit and placing a link back to www.internetworldstats.com. Copyright © 2017, Miniwatts Marketing Group. All rights reserved worldwide.

Not Only PCs connected to the Internet

• Smartphone shipments exceed PC shipments!

• 2011 shipments:
  – 487M smartphones
  – 414M PC clients
    » 210M notebooks
    » 112M desktops
    » 63M tablets
  – 25M smart TVs

• 4 billion phones in the world ➔ smartphone over next decade

- 341.6 million in Q2 2017
- 61.1 million in Q3 2015
- 36.2 million in Q1 2017
- 222 million in 2016
Societal Scale Information Systems
(Or the “Internet of Things”?)

- The world is a large distributed system
  - Microprocessors in everything
  - Vast infrastructure behind them
Example: What’s in a Search Query?

- Complex interaction of multiple components in multiple administrative domains
  - Systems, services, protocols, …
Infrastructure, Textbook & Readings

• Infrastructure
  – Website: http://cs162.eecs.berkeley.edu
  – Piazza: https://piazza.com/berkeley/fall2017/cs162
  – Webcast: Cal Central -
    https://calcentral.berkeley.edu/academics/teaching-semester/fall-2017/class/compsci-162


  – Copies in Bechtel

• Online supplements
  – See course website
  – Includes Appendices, sample problems, etc.
  – Networking, Databases, Software Eng, Security
  – Some Research Papers!
Syllabus

• OS Concepts: How to Navigate as a Systems Programmer!
  – Process, I/O, Networks and Virtual Machines
• Concurrency
  – Threads, scheduling, locks, deadlock, scalability, fairness
• Address Space
  – Virtual memory, address translation, protection, sharing
• File Systems
  – I/O devices, file objects, storage, naming, caching, performance, paging, transactions, databases
• Distributed Systems
  – Protocols, N-Tiers, RPC, NFS, DHTs, Consistency, Scalability, multicast
• Reliability & Security
  – Fault tolerance, protection, security
• Cloud Infrastructure
Learning by Doing

• Individual Homeworks: Learn Systems Programming
  – 0. Tools, Autograding, recall C, executable
  – 1. Simple Shell
  – 2. Web server
  – 3. Memory allocation

• Three Group Projects (Pintos in C)
  – 1. Threads & Scheduling
  – 2. User-programs
  – 3. File Systems
Group Project Simulates Industrial Environment

- Project teams have 4 members (try really hard to find 4 members – 3 members requires serious justification)
  - Must work in groups in “the real world”
  - Same section much preferred

- Communicate with colleagues (team members)
  - Communication problems are natural
  - What have you done?
  - What answers you need from others?
  - You must document your work!!

- Communicate with supervisor (TAs)
  - What is the team’s plan?
  - What is each member’s responsibility?
  - Short progress reports are required
  - Design Documents: High-level description for a manager!
Getting started

• Start homework 0, this Friday, 8/25
  – Github account
  – Registration survey
  – Vagrant virtualbox – VM environment for the course
    » Consistent, managed environment on your machine
  – Get familiar with all the cs162 tools
  – Submit to autograder via git

• Start forming a project group
Grading

- 42% three midterms (14% each)
  - Thursday, 9/28, 6:30-8p
  - Monday, 10/23, 6:30-8p (no class)
  - Wednesday, 11/29, 6:30-8p (no class)
- 35% projects
- 15% homework
- 8% participation
- No final exam

Projects
- Initial design document, Design review, Code, Final design
- Submission via git push triggers autograder
Personal Integrity

- UCB Academic Honor Code: "As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others."

CS 162 Collaboration Policy

Explaining a concept to someone in another group
Discussing algorithms/testing strategies with other groups
Helping debug someone else’s code (in another group)
Searching online for generic algorithms (e.g., hash table)

Sharing code or test cases with another group
Copying OR reading another group’s code or test cases
Copying OR reading online code or test cases from prior years

We compare all project submissions against prior year submissions and online solutions and will take actions (described on the course overview page) against offenders
Lecture Goal

Interactive!!!
What is an Operating System?

- **Referee**
  - Manage sharing of resources, Protection, Isolation
    - Resource allocation, isolation, communication

- **Illusionist**
  - Provide clean, easy to use abstractions of physical resources
    - Infinite memory, dedicated machine
    - Higher level objects: files, users, messages
    - Masking limitations, virtualization

- **Glue**
  - Common services
    - Storage, Window system, Networking
    - Sharing, Authorization
    - Look and feel
What is an Operating System, ... Really?

• Most Likely:
  – Memory Management
  – I/O Management
  – CPU Scheduling
  – Communications? (Does Email belong in OS?)
  – Multitasking/multiprogramming?

• What about?
  – File System?
  – Multimedia Support?
  – User Interface?
  – Internet Browser? 😊

• Is this only interesting to Academics??
Operating System Definition (Cont.)

• No universally accepted definition
• “Everything a vendor ships when you order an operating system” is good approximation
  – But varies wildly
• “The one program running at all times on the computer” is the kernel
  – Everything else is either a system program (ships with the operating system) or an application program
“In conclusion…”

• Operating systems provide a virtual machine abstraction to handle diverse hardware
  – Operating systems simplify application development by providing standard services

• Operating systems coordinate resources and protect users from each other
  – Operating systems can provide an array of fault containment, fault tolerance, and fault recovery

• CS162 combines things from many other areas of computer science:
  – Languages, data structures, hardware, and algorithms