Introduction to Operating Systems

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Why OS?
What is an OS?

- Software that converts hardware into a useful form for applications

![Diagram](image-url)
OS: Application View

- Provides an execution environment for running programs

- Provides an abstract view of the underlying computer system
  - Processors → Processes, Threads
  - Memory → Address space (virtual memory)
  - Storage → Volumes, Directories, Files
  - I/O Devices → Files (+ ioctl's)
  - Networks → Files (sockets, pipes, …)
OS: System View

- Manages various resources of a computer system

- Sharing
- Protection
- Fairness
- Efficiency
- ...

Resources

- CPU
- Memory
- I/O devices
- Queues
- Energy
- ...

...
OS: Implementation View

- Highly-concurrent, event-driven software

- Two kinds of events
  - System calls
  - Interrupts
OS: Three Pieces

- **Virtualization**
  - How to make each application believe it has each resource to itself?

- **Concurrency**
  - How to handle concurrent events correctly and efficiently?

- **Persistence**
  - How to make information survive power loss?
IG (1945-55)

- Vacuum tubes and plugboards
  - No OS
  - No programming languages
  - No assembly languages

**ENIAC** (Electronic Numerical Integrator And Computer), 1946
2G (1955-65)

- Transistors and mainframes
- Batch systems
  - One job at a time
  - Card readers, tape drives, line printers

- OS is always resident in memory and merely transfers a control (just a library)
- CPU is underutilized due to the bottleneck in I/O
3G (1965-80)

- Architectural advances
  - Integrated Circuits (ICs): better price/performance
  - Disk drives
  - On-line terminals

- Established the notion of “Computer Architecture”
  - IBM System/360 family
3G (1965-80)

- **Multiprogrammed systems**
  - Increase CPU utilization
  - **OS features**
    - Job scheduling
    - Memory management
    - CPU scheduling
    - Concurrency
    - Protection
    - Spooling (Simultaneous Peripheral Operation On-Line)

- IBM OS/360 (1964)
3G (1965-80)

- **Time-sharing systems**
  - Improve response time
  - **OS features**
    - Sophisticated CPU scheduling
    - Virtual memory and swapping
    - File system
    - Synchronization
    - Interprocess communication
    - Interactive shell
    - More protection, …

- MIT CTSS (1961), Multics (1965), Unix (1969)
4G (1980-)

- Architectural advances
  - Microprocessors (LSIs & VLSIs): smaller and faster
  - Storage: larger and faster
  - Personal computers
  - CPU work is offloaded to I/O devices

- Modern OS features
  - GUI (Graphical User Interface)
  - Multimedia
  - Internet & Web
  - Mobile / Networked / Distributed
  - Virtualization, etc.
Multics (1)

- Multiplexed Information and Computing Service
- A time-shared, multi-processor mainframe “computing utility”
- Originally started by MIT, GE, and Bell Labs in 1965 for GE-645, a 36-bit system
  - Bell Labs quit in 1969 and built Unix
  - GE’s computer business, including Multics, was taken over by Honeywell in 1970
  - Last system shutdown on 10/31/2000
- [http://www.multicians.org](http://www.multicians.org)
Multics (2)

- **Multics innovations**
  - Hierarchical file system
    - ACLs, long names, hard & symbolic links, quota, …
  - Virtual memory (segmentation and paging)
  - User-level command shell
  - Dynamic linking, shared memory
  - Implementation in a high level language (PL/1)
  - Mapping of logical disk volumes onto physical volumes
  - Support for BCPL, APL, Fortran, Lisp, C, Cobol, Algol, Pascal, …
  - Multics Relational Data Store (MRDS), Spreadsheets
  - Rated B2 by NCSC (National Computer Security Center)
Unix (I)

“… When BTL (Bell Telephone Laboratories) withdrew from the Multics project, they needed to rewrite an operating system in order to play space war on another smaller machine (a DEC PDP-7 with 4K memory for user programs). The result was a system which a punning colleague called UNICS (UNiplexed Information and Computing Services) – an ‘emasculated Multics’; no one recalls whose idea the change to UNIX was.”


“… It was the summer of '69. In fact, my wife went on vacation to my family's place in California.... I allocated a week each to the operating system, the shell, the editor, and the assembler, to reproduce itself, and during the month she was gone, it was totally rewritten in a form that looked like an operating system, with tools that were sort of known, you know, assembler, editor, and shell .... Yeh, essentially one person for a month.”

-- Ken Thompson
Unix (2)

- **Unix features**
  - **Hierarchical file system**
    - Special files: uniform I/O, naming, and protection
    - Removable file systems via mount/umount
    - i-node
  - **Process control**
    - fork(), exec(), wait(), exit()
    - Pipes for inter-process communication
  - **Shells**
    - Standard I/O and I/O redirection
    - Filters, command separators
    - Shell scripts
  - **Signals**
Multics vs. Unix

- **Multics**
  - Top-down approach
  - 150 MY for design and system programming, another 50 MY for improvements
  - Too complicated, too costly hardware
  - Many novel ideas had a great impact

- **Unix**
  - Bottom-up approach
  - 2 MY: Simplicity, elegance, and ease of use
  - Low cost hardware, university adoption
  - The root of the modern operating systems