Introduction to Operating Systems

Jinkyu Jeong (jinkyu@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Today’s Topics

- What is OS?
- History of OS
What is OS? (1)

- **Application view**
  - Provides an execution environment for running programs
  - Provides an abstract view of the underlying computer system
    - Processors → Processes, Threads
    - Memory → Address spaces (virtual memory)
    - Storage → Volumes, Directories, Files
    - I/O Devices → Files (ioctl)
    - Networks → Files (sockets, pipes, ...)
    - ...

What is OS? (2)

- **System view**
  - Manages various resources of a computer system
  - Sharing
  - Protection
  - Fairness
  - Efficiency
  - ...

**Resources**

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

[Image of green background with text]
What is OS? (3)

- Implementation view
  - Highly-concurrent, event-driven software

```
trap
```
```
System call
```
```
Interrupts
```
```
Hardware
```
Computer Systems

- Services
  - Applications
    - Middleware
    - Software Development Environment (compilers, loaders, editors, utilities, command interpreter, libraries)
  - Operating System (Kernel)
- Computer System Architecture (Hardware Platform)
History of Operating Systems
In The Beginning

In ENIAC, human labor is the operating systems
Evolution of Operating Systems

- **Phase 1: Hardware is expensive, humans are cheap**
  - User at console
  - Single-user systems, batching systems, multi-programming systems
  - Single user systems
    - OS = loader + libraries of common subroutines
    - Problem: low utilization of expensive components

- **Phase 2: Hardware is cheap, humans are expensive**
  - Time sharing: Users use cheap terminals and share servers

- **Phase 3: Hardware is very cheap, humans are very expensive**
  - Personal computing: One system per user
  - Distributed computing: lots of systems per user

- **Phase 4: Ubiquitous computing**
  - Mobile computing
  - Embedded computing
  - Internet-of-Things
1st Generation (1945-55) -- Vacuum Tubes and Plugboards

- No OS
- No programming languages
- No assembly languages

ENIAC (Electronic Numerical Integrator And Computer), 1946
Open Shop Model

- **Key OSs**
  - IBM 701
  - TX-0

- **Each user was allocated a minimum 15-min slot**
  - 10 min for setting up equipment
  - 5 min for computation

- **Wasted computer time of IBM 701 was $146,000 per month in 1954 dollars!**
2nd Generation (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.
  - CPU is underutilized due to the bottleneck in I/O
Batch Systems

- Emergence of fast input device
  - Magnetic tapes

- Operators collect punch cards and transfer them to magnetic tapes using satellite computers

- How can you improve batch process based on this technical advancement?
Batch Systems

- Using multiple tape readers makes computers work even at tape preparation time
- OS = loader + libraries of common subroutines
3rd Generation (1965-80) -- Integrated Circuits (ICs)

- Architectural advances
  - Using ICs: better price/performance
  - Large core memories
  - Secondary storage (drums) with random access
  - Hardware interrupts
  - On-line terminals

- The notion of “Computer Architecture”:  
  - IBM System/360 family
3rd Generation (1965-80)
-- Integrated Circuits (ICs)

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

- IBM MFT - ‘Multiprogramming with Fixed Tasks’
  - fixed partition boundaries

- IBM MVT - ‘Multiprogramming with Variable Tasks’
  - dynamic partition sizes and positions
  - OS scheduled jobs to minimize memory wastage

- Demand paging
  - First used in ATLAS OS
  - Core memory of 16 K and a drum of 96 K words
  - Core store is divided into 512 word pages
  - Drums and magnetic tapes also use 512 word fixed blocks

- ATLAS first used supervisor calls known as “extracodes”
Complications by Multiprogramming

- Jobs had to run at different places in memory
- Inefficient use of expensive memory
- One job could interfere with another e.g. overwrite other jobs memory
- Jobs could clash over use of I/O devices
Solutions to Complications

- Relocating linkers and loaders
- Hardware relocation base and limit registers
  - Read/write/execute protected access
- OS controlled access to I/O devices
- OS finally got complex

Core memory, the origin of “core dump”
3rd Generation (1965-80)
-- Integrated Circuits (ICs)

- Time-sharing systems
  - Improve response time
  - OS features
    - Swapping
    - Virtual memory
    - File system
    - Sophisticated CPU scheduling
    - Synchronization
    - Interprocess communication
    - Interactive shell
    - More protection, ...
Time Sharing

- John McCarthy at MIT proposed original idea of time sharing in 1959
- CTSS (compatible time-sharing system) developed at MIT in 1962 was the first time sharing OS
  - Multilevel feedback queue scheduling
  - Notion of background and foreground processes
- Multics File System (released in 1969)
  - Concept of general purpose file system
  - Hierarchical file systems for private and shared files
4th Generation (1980-)
-- LSIs & VLSIs

- **Architectural advances**
  - Microprocessors: 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
  - Networked / Distributed, etc.
OS History

CTSS (1961, MIT)  
(Compatible Time Sharing System)

OS/360 (1964, IBM)

MULTICS (1965, MIT, Bell Labs, GE)  
(MULTiplexed Information and Computing Service)

Unix (1969, Bell Labs)
Multics (1)

- **Multics**
  - Multiplexed Information and Computing Service
  - A time-shared, multi-processor mainframe “computing utility”
  - Originally started by MIT, GE, and Bell Labs for GE-645, a 36-bit system, in 1965.
    - 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
Multics (2)

### Multics innovations

- Hierarchical file system.
  - File / directory / path name / working directory
  - Access Control Lists (ACLs).
  - Long names on entries.
  - Multiple names on entries.
  - Symbolic links.
  - Storage quotas.
  - Removable devices.
  - The backup procedures.

- Lots of developments in management of virtual memory including segmentation and paging.
Multics (3)

- **Multics innovations (cont’d)**
  - Separating the command shell from the OS kernel.
  - Dynamic linking.
  - Implementation of an OS in a high level language (PL/1)
  - Management of shared memory.
  - Mapping of logical disk volumes onto physical volumes.
  - Many developments in the area of secure computer systems.
    - Multics was rated B2 by the NCSC in 1985.
    - A subsequent system (based on the Multics experience) built by Honeywell was the first computer system ever rated A1.
Multics innovations (cont’d)

• Multics Relational Data Store (MRDS) in 1976.
  – The first commercial relational DBMS.
  – The MRDS query language was similar to early SQL.
  – Concurrent access to a database by multiple processes was supported.
  – The database could be backed up in its entirety.

• Spreadsheets were developed on the Multics platform.

• Multics supports BCPL, BASIC, APL, FORTRAN, LISP, C, COBOL, ALGOL 68, and Pascal.

• Many optimizations for the LISP language through work on the Multics MACLISP compiler.
“... 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 systems
    - 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
Unix (4)

- Sun Solaris
- HP HP-UX
- IBM AIX
- Compaq (Digital) Tru64
- SGI Irix
- SCO Unixware
- Linux
- FreeBSD, NetBSD, OpenBSD
- Apple Mac OS X & iOS
- Android, etc.
Multics vs. Unix (1)

- **Timing**
  - Multics: 1965 – 2000
    - The system was made available for general use in 1969.
  - Unix: 1969 – Present

- **Design approach**
  - Multics: Top-down
    - Clear goals drive design and primitives.
  - Unix: Bottom-up
    - Primitives define functionality (No predefined goals), while retaining simplicity and elegance.
Multics vs. Unix (2)

Development

• Multics:
  – Well-organized:
    MIT Project MAC: Design is key strength.
    Bell Labs: Online update capabilities, robustness, scalability
    GE/Honeywell: Defense contractor, hardware.
  – 3000-page program module specification BEFORE coding.
  – An unanticipated phase of design iterations was necessary
    due to gross discrepancies between actual and expected
    performance.
    » Design, build, evaluate, design, build, evaluate, ...
  – $6M / year for five years.
  – 150 Man-Years for design and system programming, another
    50 Man-Years for improvements.
Multics vs. Unix (3)

- **Development (cont’d)**
  - Unix:
    - Two Man-Years.
    - Build/design mixed at same time.

- **Target users**
  - Multics: military, government agencies, NSA, ...
  - Unix: universities, research institutions, ...

- **Protection**
  - Multics: Protection rings
  - Unix: All or nothing (superuser model)
Multics vs. Unix (4)

- **Resource utilization**
  - **Multics:**
    - Memory: 256KWords/box (1 word = 36 bits)
    - Paging device (Drum): 4MWords
    - Disk: 136MB
  - **Unix:**
    - Old, unused PDP-7
    - Occupies 42KBytes on PDP-11/45 with 144KBytes of memory.
    - PDP-11 has a 1MB disk used for file system storage and swapping.
Multics vs. Unix (5)

Resource management

- Multics:
  - Multiprocessor support from start.
  - Transparent multi-level memory model: memory – drum – disk
  - True memory-mapped world view.
- Unix:
  - Simple 3 segment view.
  - Partial memory mapped model later.
Multics vs. Unix (5)

- **Summary**
  
  - **Multics:**
    - Too complicated, too costly hardware.
    - “Second system effect” (by Fred Brooks, The Mythical Man-Month)
      - System one was successful (CTSS)
      - System two aims to correct all flaws from #1
      - Almost always fails.
  
  - **Unix:**
    - Low cost hardware
    - University adoption
    - Simplicity, elegance, and ease of use.
### Windows & Linux

| Year | 1981 | 82 | 84 | 86 | 88 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 00 | 01 | 03 |
|------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| MS-DOS Family | 1.0 | 2.0 | 3.0 | 3.2 | 4.0 | 5.0 | 6.0 | 7.0 | | | | | | | | | |
| Windows Family | 1.0 | 2.0 | 3.0 | | | | | | 3.1 | | | | | | | | |
| Windows for WorkGroup Family | 3.1 | 3.11 | | | | | | 95 | | | | | | | | |
| Windows NT Family | | | | | 3.1 | 3.5 | 3.51 | 4.0 | | | | | | | | |
| Linux | 0.01 | 0.99 | 1.0 | 1.2 | 2.0 | 2.1 | 2.2 | | | | | | | | | | 2.4 | 2.6 |

**Images:**
- Windows Logo
- Linux Logo