Operating System (OS)

National Tsing-Hua University
2016, Fall Semester
Historical Prospective

Mainframe Systems

Computer-system architecture

Special-purpose Systems
System Category

- Mainframe Systems
  - Batch
  - Multi-programming
  - Time-sharing

- Computer-system architecture

- Special-purpose Systems
Mainframe Systems

- One of the earliest computers
  - Slow I/O devices: card reader/printer, tape drivers

- Evolution:
  - Batch ➔ Multi-programming ➔ Time-shared

- Still exists in today’s world...
  - For critical application with better **reliability & security**
  - Bulk data processing
  - Widely used in hospitals, banks

IBM 704 mainframe in 1954
Mainframe: Batch Systems

- Processing steps:
  - Users submit jobs (program, data, control card)
  - Operator sort jobs with similar requirements
  - OS simply transfer control from one job to the next

Memory layout: 
- Operating system
- User program area

Diagram:
- $JOB
- $FNT
- $LOAD
- $RUN
- $END

Data for the program
Program to be compiled
Mainframe: Batch Systems

- **Drawbacks:**
  - One job at a time
  - No interaction between users and jobs
  - CPU is often idle
  - I/O speed $\ll$ CPU speed (at least 1:1000)

- **OS doesn’t need to make any decision**
Mainframe: Multi-programming System

- Overlaps the I/O and computation of jobs
  - Keeps both CPU and I/O devices working at higher rates

- **Spooling** *(Simultaneous Peripheral Operation On-Line)*
  - I/O is done with no CPU intervention
  - CPU just needs to be notified when I/O is done
Mainframe: Multi-programming System

- Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.

<table>
<thead>
<tr>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job1</td>
</tr>
<tr>
<td>Job2</td>
</tr>
<tr>
<td>Job3</td>
</tr>
<tr>
<td>Job4</td>
</tr>
</tbody>
</table>

CPU Scheduling

Memory

Job Scheduling

Disk

Job pool
Mainframe: Multi-programming System

OS tasks

- Memory management (chap 9) — the system must allocate the memory to several jobs.
- CPU scheduling (chap 6) — the system must choose among several jobs ready to run.
- I/O system (chap 13) — I/O routine supplied by the system, allocation of devices.
Mainframe: Time-sharing System (Multi-tasking System)

- An interactive system provides direct communication between the users and the system
  - CPU switches among jobs so frequently that users may interact with programs
  - Users can see results immediately (response time < 1s)
  - Usually, keyboard/screen are used

- Multiple users can share the computer simultaneously

- Switch job when
  - finish
  - waiting I/O
  - a short period of time
Mainframe: Time-sharing System (Multi-tasking System)

- **OS tasks**

  - **Virtual memory (chap 10)** — jobs swap in and out of memory to obtain reasonable response time
  
  - **File system and disk management (chap11,12)** — manage files and disk storage for user data
  
  - **Process synchronization and deadlock (chap7,8)** — support concurrent execution of programs
## Mainframe System Summary

<table>
<thead>
<tr>
<th></th>
<th>Batch</th>
<th>Multi-programming</th>
<th>Time-sharing (Multi-tasking)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Model</strong></td>
<td>Single user</td>
<td>Multiple prog.</td>
<td>Multiple users</td>
</tr>
<tr>
<td></td>
<td>Single job</td>
<td></td>
<td>Multiple prog.</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Simple</td>
<td>Resource utilization</td>
<td>Interactive Response time</td>
</tr>
<tr>
<td><strong>OS features</strong></td>
<td>N.A</td>
<td>CPU scheduling</td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory Mgt.</td>
<td>Virtual memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I/O system</td>
<td>Synchronization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Deadlock</td>
</tr>
</tbody>
</table>
System Category

- Mainframe Systems

- Computer-system architecture
  - Desktop Systems: single processor
  - Parallel Systems: tightly coupled
  - Distributed Systems: loosely coupled

- Special-purpose Systems
Desktop Systems: Personal Computers

- Personal computers (PC) – computer system dedicated to a single user
- User convenience and responsiveness – GUI
- I/O devices – keyboards, mice, screens, printers
- Several different types of operating systems
  - Windows, MacOS, Unix, Linux
- Lack of file and OS protection from users
  - Worm, Virus
Parallel Systems

- A.k.a multiprocessor or tightly coupled system
  - More than one CPU/core in close communication
  - Usually communicate through shared memory

- Purposes
  - Throughput, Economical, Reliability
Parallel Systems

- **Symmetric multiprocessor system (SMP)**
  - Each processor runs the same OS
  - Most popular multiple-processor architecture
  - Require extensive synchronization to protect data integrity

- **Asymmetric multiprocessor system**
  - Each processor is assigned a specific task
  - One Master CPU & multiple slave CPUs
  - More common in extremely large systems
Multi-Core Processor

- A CPU with **multiple cores on the same die** (chip)
- On-chip communication is **faster** than between-chip communication
- One chip with multiple cores uses significantly **less power** than multiple single-core chips

**blade servers:**
Each blade-processor board boots independently and run its own OS
Many-Core Processor

- **Nvidia General-Purpose GPU**
  - First release in Apr. 2008
  - Utilize a graphics processing unit (GPU)
  - **Single Instruction Multiple Data**
  - 2,880 thread processor, **1.43 TGlops (x200 faster than a single Intel Core i7)**
  - 245 WATTS, Clock freq. 600~750 MHz
  - **$3000 USD**

- **Intel Xeon Phi**
  - First release in Nov. 2012
  - A coprocessor computer architecture based on Intel Many Integrated Core (MIC)
  - 61 cores, 1.2 TFlops, 300 WATTS

- **TILE64**
  - A mesh network of 64 "tiles"
  - Each tile houses a **general purpose** processor
Memory Access Architecture

- **Uniform Memory Access (UMA):**
  - Most commonly represented today by Symmetric Multiprocessor (SMP) machines
  - Identical processors
  - Equal access times to memory
  - Example: most commodity computers

- **Non-Uniform Memory Access (NUMA):**
  - Often made by physically linking two or more SMPs
  - One SMP can directly access memory of another SMP
  - Memory access across link is slower
  - Example: IBM Blade server
Distributed Systems

- Also known as *loosely coupled system*
  - Each processor has its own *local memory*
  - Processors communicate with one another through various communication lines (I/O bus or *network*)
  - Easy to *scale* to large number of nodes (hundreds of thousands, e.g. Internet)

- Purposes
  - Resource sharing
  - Load sharing
  - Reliability

- Architecture: peer-to-peer or client-server
Client-Server Distributed System

- Easier to manage and control resources
- But, server becomes the bottleneck and single failure point
Peer-to-Peer Distributed System

- Every machine is identical in its role in the distributed system – decentralized
- Example: ppStream, bitTorrent, Internet
Clustered Systems

Definition:
- Cluster computers share storage and are closely linked via a local area network (LAN) or a faster interconnect, such as InfiniBand (up to 300Gb/s).

- **Asymmetric clustering**: one server runs the application while other servers standby

- **Symmetric clustering**: two or more hosts are running application and are monitoring each other
System Architecture Summary

- Distributed
- Cluster
- Multi-processor
- Multi-core
- Single core

Tightly coupled

Loosely coupled
System Category

- Mainframe Systems
- Computer-system architecture
- Special-purpose Systems
  - Real-Time Systems
  - Multimedia Systems
  - Handheld Systems
Real-Time Operating Systems (Chap19)

- Well-defined fixed-time constraints
  - “Real-time” doesn’t mean speed, but *keeping deadlines*

- Guaranteed response and reaction times

- Often used as a control device in a dedicated application:
  - Scientific experiments, medical imaging systems, industrial control systems, weapon systems, etc

- Real-time requirement: *hard or soft*
Soft vs. Hard Real-Time

- **Soft** real-time requirements:
  - Missing the deadline is unwanted, but is not immediately critical
  - A critical real-time task gets *priority* over other tasks, and retains that priority until it completes
  - Examples: multimedia streaming

- **Hard** real-time requirements:
  - Missing the deadline results in *a fundamental failure*
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Examples: nuclear power plant controller
Multimedia Systems (Chap20)

- A wide range of applications including audio and video files (e.g. ppstream, online TV)

Issues:

- **Timing constraints**: 24~30 frames per second
- **On-demand/live streaming**: media file is only played but not stored
- **Compression**: due to the size and rate of multimedia systems
Handheld/Embedded Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones
- HW specialized OS

Issues

- Limited memory
- Slow processors
- Battery consumption
- Small display screens
Computer Systems

- Super Computer
- Mainframe
- PC
- Parallel System
- Cluster
- Distributed System

- computing power
- processor connectivity
- special purpose
Computer Systems

- Which system to use? How to use it?
- They have many things in common, but also with different design decisions for their OS.
Review Slides

- Mainframe system
  - Batch, **Multi-programming** and **Time-sharing**
- Tightly coupled system vs. loosely coupled system
- NUMA vs. UMA
- Distributed system:
  - Client-server vs. P2P
- Real-time system:
  - Soft vs. Hard real-time