國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Institute of Artificial Intelligence Innovation
Department of Computer Science

Operating System

Lecture 00: Couse Overview & Historical Prospective

Shuo-Han Chen (陳碩漢),
shch@nycu.edu.tw

Wed. 10:10 - 12:00 EC115 +
Fri. 11:10 – 12:00 Online

Course Information

- Operating System
  - Mandarin Lecture + English Materials
  - Designed for students in graduate school
- Course Time and Place - Lecture (2 + 1 hours)
  - Wed. 10:10 - 12:00 EC115
  - Friday 11:10 – 12:00 Online
- Course Material & Video
  - Please check the MS Teams
  - Code: ig0604f
  - Visit the NYCU portal to apply MS365 account first and log into MS teams

Course Instructor & Teaching Assistant

- Course Instructor
  - Prof. Shuo-Han Chen (陳碩漢)
    - Office: EF373
    - Office Hours: By Request
    - Email: shch@nycu.edu.tw
- Teaching Assistant
  - 徐翊安
  - 簡子茸
  - Message TA through MS Teams message

Prerequisites

- Addend & Interact
- Comfortable with C/C++
- Already taken the Operating System Course
- Not afraid of English

- And, of course, willing to learn more about Operating System

Linux™

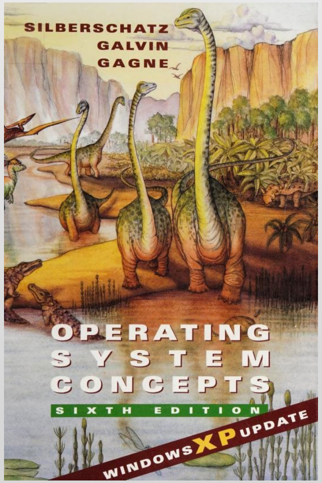
C/C++

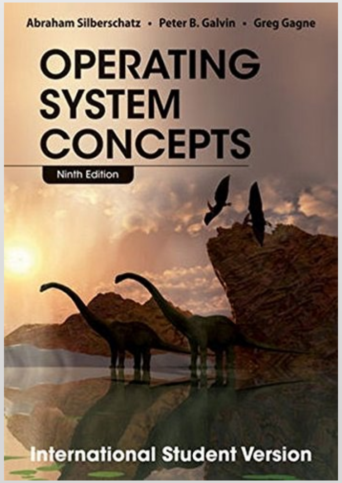
Course Assessment

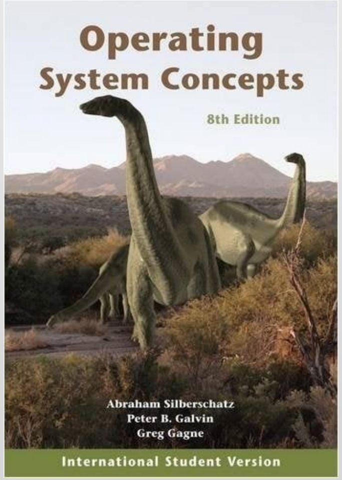
- **Assignment: 40 %**
  - 4 homework in total (2-people group homework, single-person group is allowed)
  - Upload your HW to the course GitLab (Account will be provided after collecting group info)
    - https://css-nachos.hopto.org/gitlab
    - https://css-nachos.hopto.org/jenkins
- **Midterm Exam: 30 %**
- **Final Exam: 25 %**
  - Closed-book book
  - Week 8 & Week 16 on Wed. 10:10 - 12:00
- **Course Attendance: 5% (By deduction)**
  - Response to course questionnaire, up to 2 times
  - Random roll call, up to 3 times

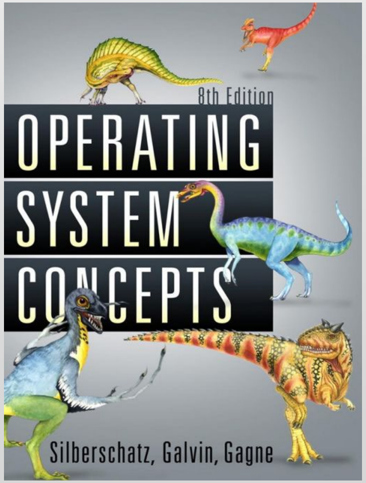
Textbook

- Textbook:
  - ”Operating System Concepts, 9^{th} Edition” by Silberschatz, Galvin, and Gagne. John Wiley & Sons, Inc









- Prerequisites: Data Structures, Computer Organization, and C++ Language

Homework : NachOS

- Features:
  - an educational OS developed at UC Berkeley
  - clean, simple to trace, compared with Linux
  - widely used by many universities in USA
  - you will add system call, memory manager, process scheduler and file system
- Pre-request Knowledge:
  - C++ Language
  - Linux coding environment
  - Code tracing

Grading Policy

- Correctness of the code - Verify through Jenkins automation testing
- Report
  - Team member information
  - Individual contribution
    - Name, Percentage, Briefly describe the contribution
  - Explanation of your implementation – As detail as possible
- Following rules will be strictly enforced
  - 0 will be given to cheaters (copying cats)
  - Late submission
    - 3 days: 90%
    - 1 week: 80%
    - 2 weeks: 70%
    - 3 weeks: 60%
    - Further, will not be accepted

Textbook Content

Course Content

PART 1	Overview
Lec01	Introduction
Lec02	System Structures (HW01)
PART 2	Process Management
PART 3	Process Coordination
PART 4	Memory Management
PART 5	Storage Management
PART 6	Protection And Security
PART 7	Distributed Systems
PART 8	Special Purpose Systems
PART 9	Case Studies

Course Content

PART 1	Overview
PART 2	Process Management
Lec03	Process Concept
Lec04	Multithreaded Programming
Lec05	Process Scheduling (HW03)
PART 3	Process Coordination
Lec06	Synchronization
Lec07	Deadlocks
PART 4	Memory Management
PART 5	Storage Management
PART 6	Protection And Security
PART 7	Distributed Systems

Course Content

PART 1	Overview
PART 2	Process Management
PART 3	Process Coordination
PART 4	Memory Management
Lec08	Memory-Management Strategies
Lec09	Virtual-Memory Management (HW02)
PART 5	Storage Management
Lec10	File System (HW04)
Lec11	Implementing File Systems
Lec12	Mass Storage Structure
Lec13	I/O Systems

Course Syllabus

- Introduction (Lec01-Lec02)
  - MP1
- Process (Lec03)
- Memory (Lec08-Lec09)
  - MP2
- Midterm
- Threading & CPU Scheduling (Lec04-Lec05)
  - MP3
- Synchronization & Deadlock (Lec06-Lec07)
- File System & I/O Systems (Lec10-Lec13)
  - MP4
- Final Exam

Course Schedule

W	Date	Lecture	Online	Homework
1	Sept. 4	Lec01: Couse Overview & Historical Prospective		
2	Sept. 11	Lec02: Introduction	V	
3	Sept. 18	Lec03: OS Structure	V	HW01
4	Sept. 25	Lec04: Processes Concept	V	
5	Oct. 2	Lec08: Memory Management	V	
6	Oct. 9	Lec09: Virtual Memory Management	V	HW02
7	Oct. 16	Lec05: Process Scheduling	V	
8	Oct. 23	School Midterm Exam		
9	Oct. 30	Lec06: Process Synchronization	V	
10	Nov. 6	Lec07: Deadlocks	V	HW03
11	Nov. 13	Lec10: File System Interface	V	
12	Nov. 20	School Event – No class		
13	Nov. 27	Lec11: File System Implementation	V	HW04
14	Dec. 4	Lec12: Mass Storage System	V	
15	Dec. 11	Lec13: IO Systems	V	
16	Dec. 18	School Final Exam		

Questionnaire & Join MS teams today

- Please help me understand you more

Questionnaire (before 09/13)

How to apply MS account

- And, Join MS teams right after the class
- Code: ig0604f

System Category

- Mainframe Systems
- Computer-system architecture
- Special-purpose Systems

System Category

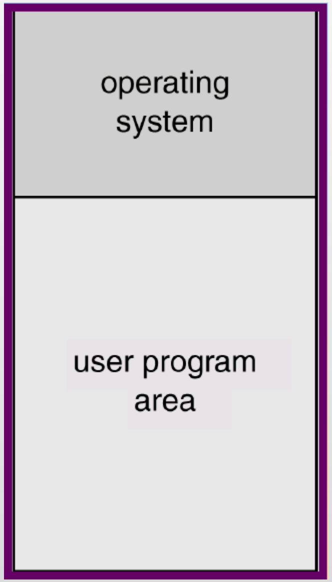
- Mainframe Systems
  - Batch
  - Multi-programming
  - Time-Sharing
- Computer-system architecture
- Special-purpose Systems

Mainframe Systems

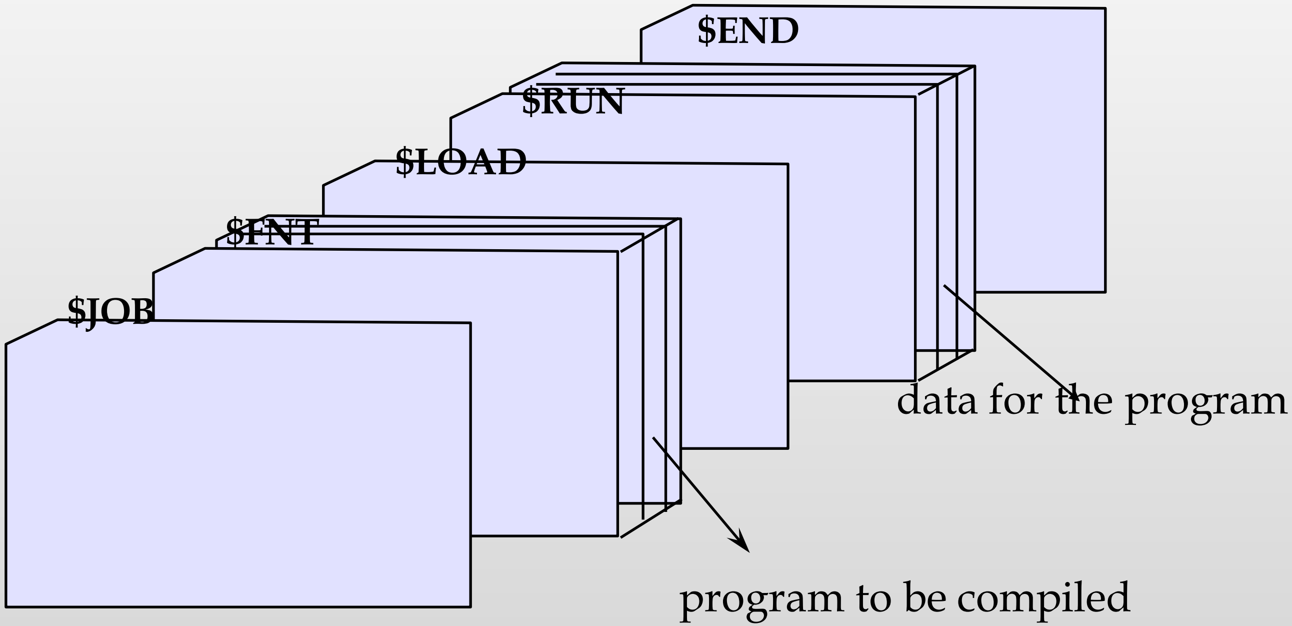
- One of the earliest computers
    - Slow I/O devices: card reader/printer, tape drivers
- Evolution:
    - Batch → Multi-programing → Time-shared
- Still exists in today's world…
    - For critical application with better **reliability & security**
    - Bulk data processing
    - Widely used in hospitals, banks

IBM 704 mainframe in 1954

Mainframe: Batch Systems



Memory layout



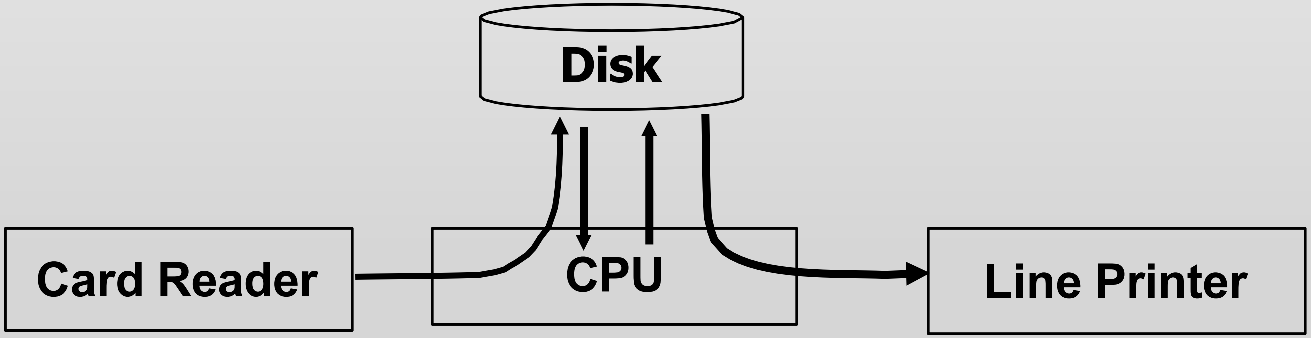
- Processing steps:
    - Users submit jobs (program, data, control card)
    - Operator sort jobs with similar requirements
    - OS simply transfer control from one job to the next

Mainframe: Batch Systems

- Drawbacks:
  - One job at a time
  - No interaction between users and jobs
  - CPU is often idle
    - I/O speed << CPU speed (at least 1:1000)
- OS doesn’t need to make any decision

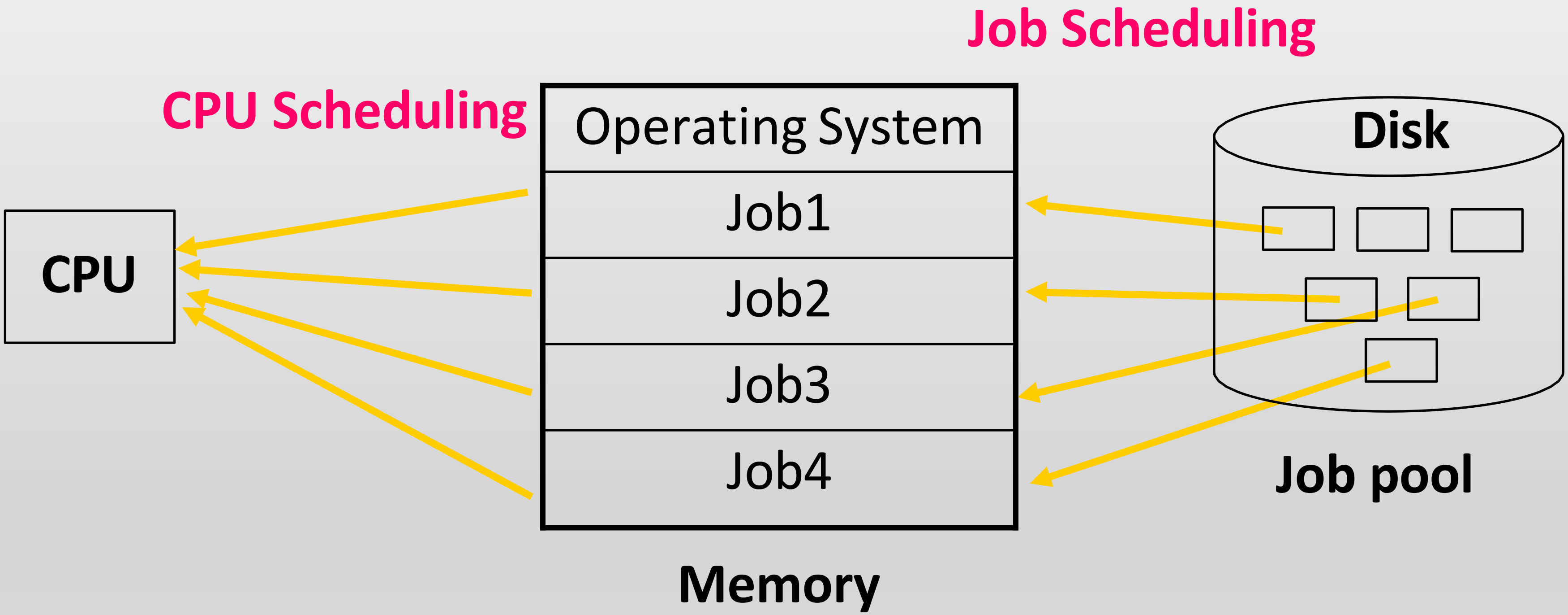
Mainframe: Multi-programming System

- 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



Mainframe: Multi-programming System

- OS tasks
  - Memory management (Lec09) - the system must allocate the memory to several jobs
  - CPU scheduling (Lec06) - the system must choose among several jobs ready to run.
  - I/O system (Lec13) - I/O runtime 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 (Lec10) - jobs swap in and out of memory to obtain reasonable response time
  - File system and disk management (Lec11, 12) - manage files and disk storage for user data
  - Process synchronization and deadlock (Lec07, 08) - support concurrent execution of programs

Mainframe System Summary

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

System Category

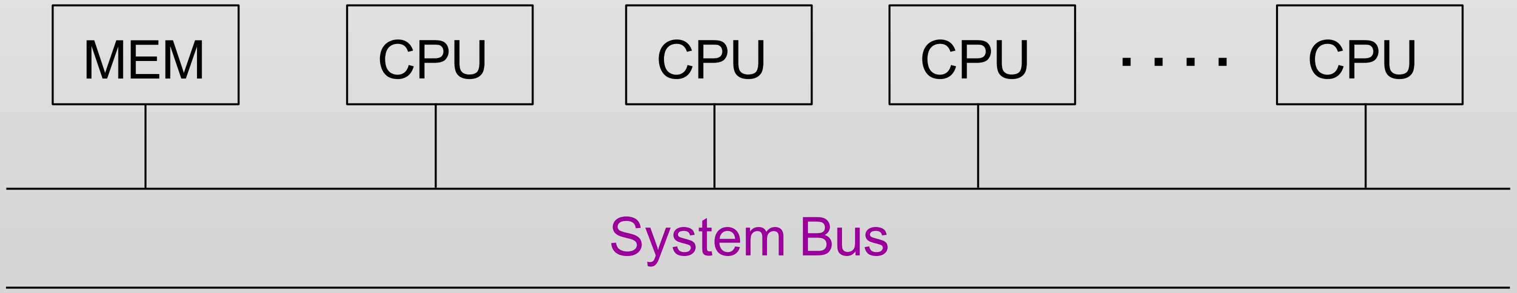
- Mainframe Systems
- Computer-system architecture
  - Desktop Systems: single processor
  - Parallel Systems: tightly coupled
  - Distributed Systems: loosely coupled
- Special-purpose Systems

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**

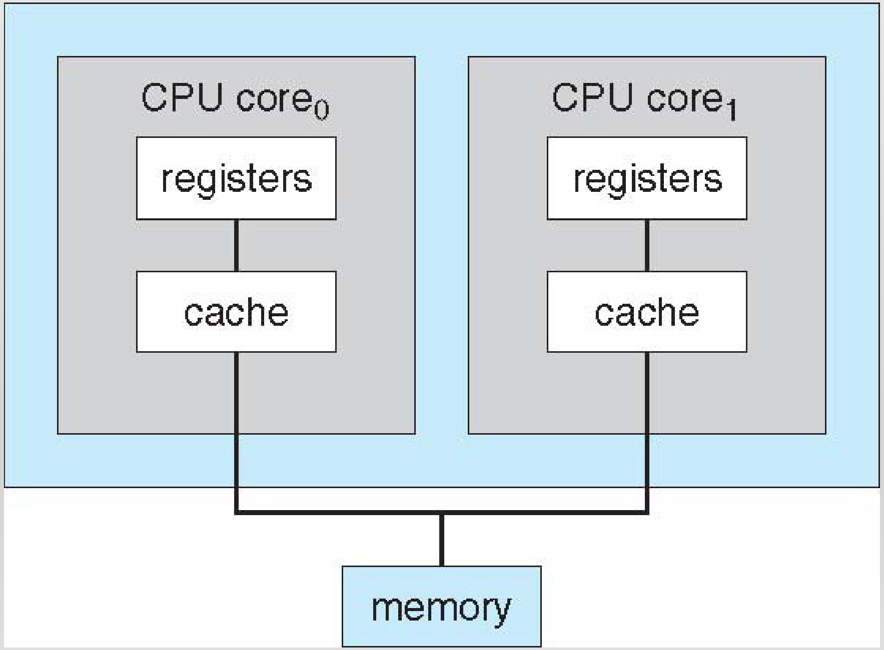


Parallel Systems

- 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

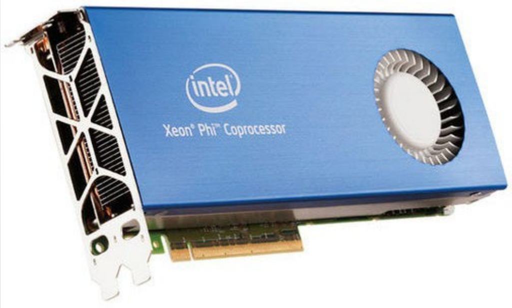


blade servers:
Each blade–processor board
boots independently and run its own OS

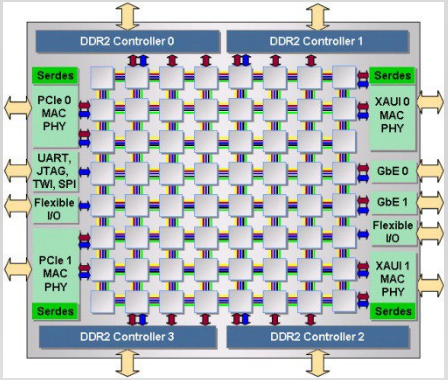
Many-Core Processor

- Nvidia General-Purpose GPU
  - First release in Apr. 2008
  - Utilize a graphics processing unit (GPU)
  - **Single Instruction Multiple Data**
  - 2,880 thread processor, **1.43TGlops (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 , 1.2TFlops, 300WATTS

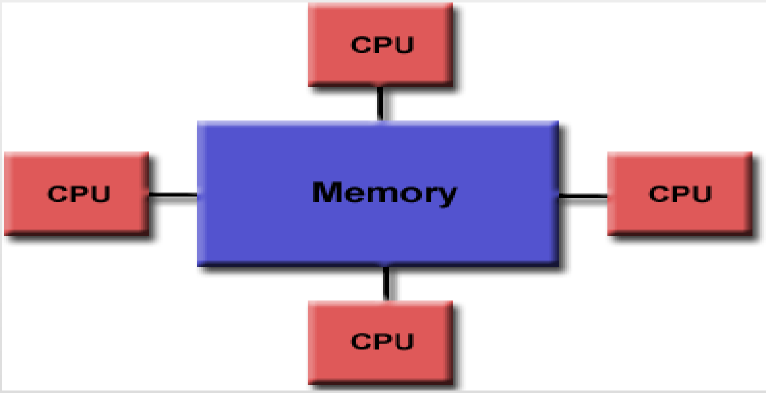


- TILE64
  - A mesh network of 64 "tiles“
  - Each tile houses a **general purpose** processor

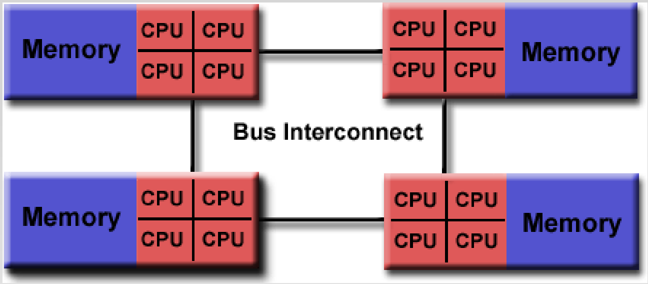


Memory Access Architecture

- 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

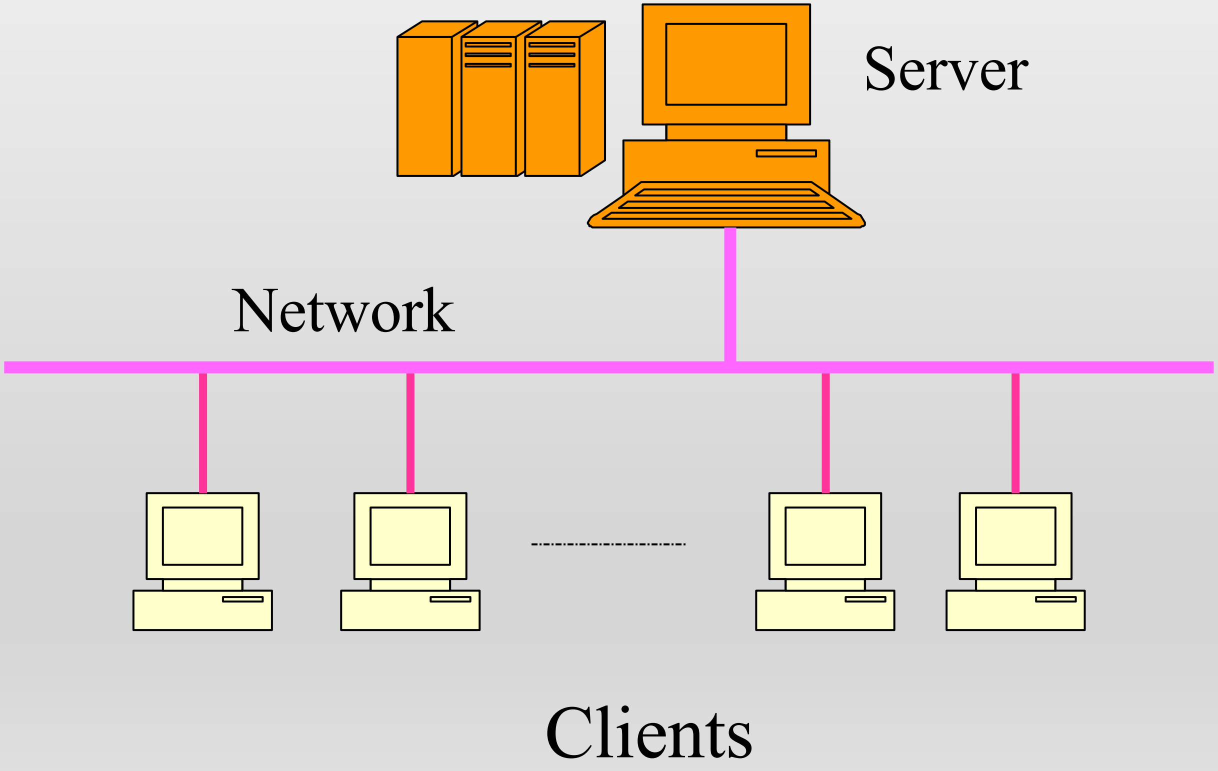


Distributed Systems

- 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

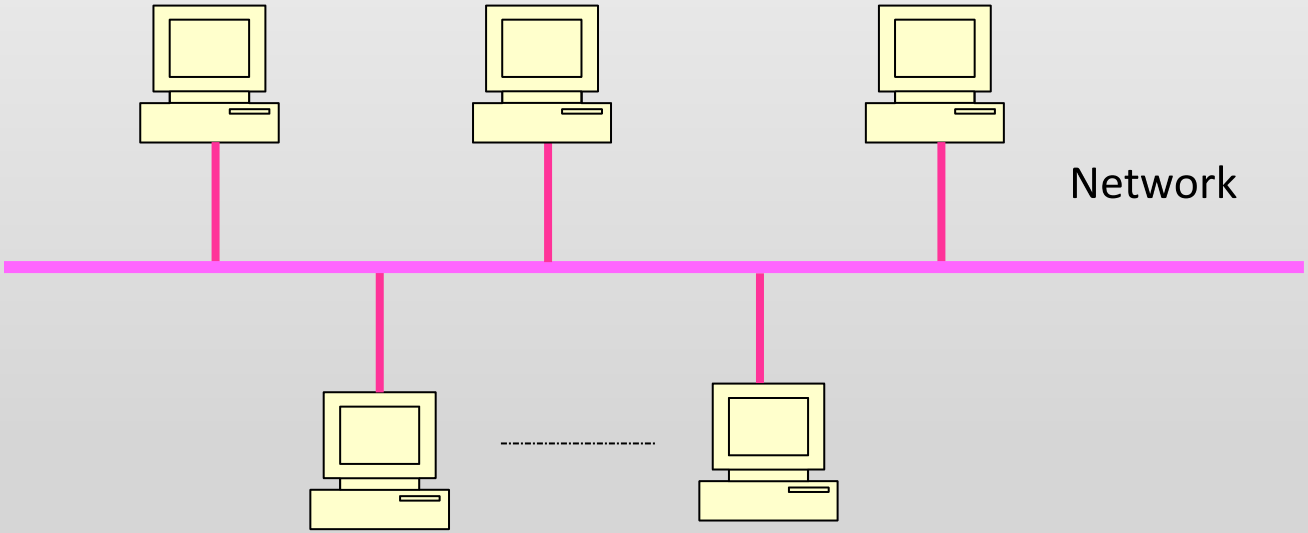
Client-Server Distributed System

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



Peer-to-Peer Distributed System

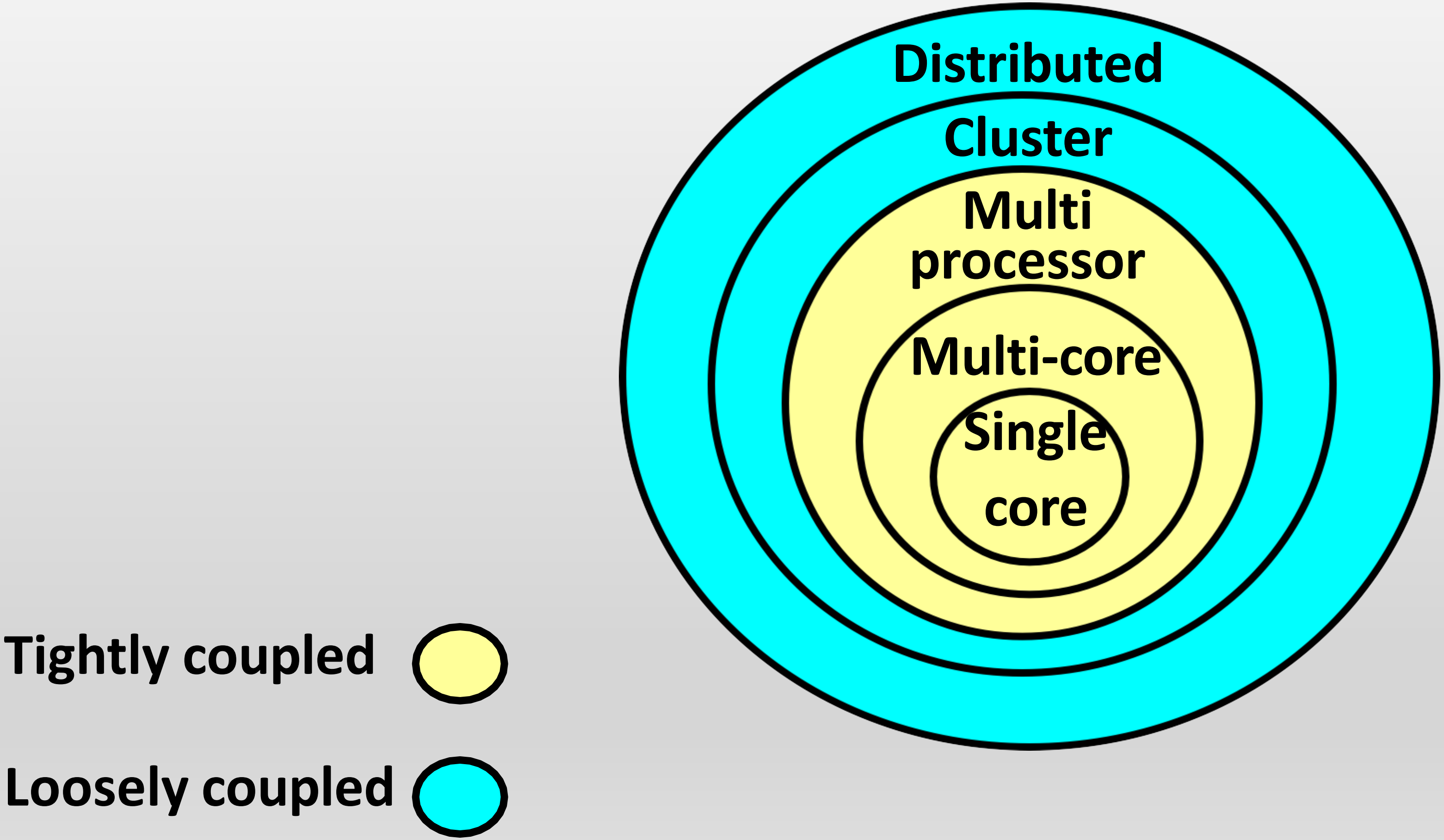
- Every machine is identical in its role in the distributed system – decentralized
- Example: ppStream, bitTorrent, Internet



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



System Category

- Mainframe Systems
- Computer-system architecture
- Special-purpose Systems
  - Real-Time Systems
  - Multimedia Systems
  - Handheld Systems

Real-Time Operating Systems

- 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

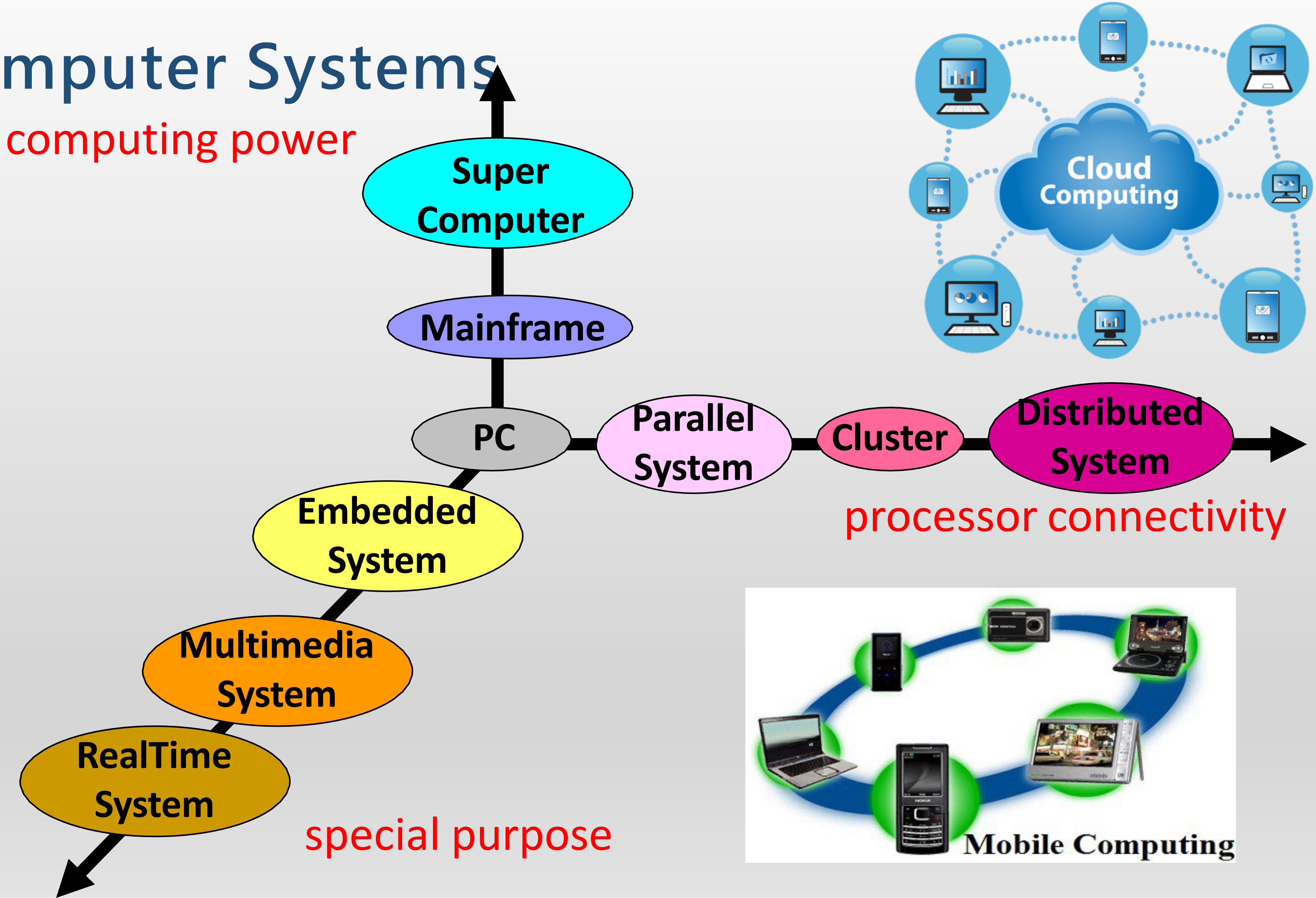
Multimedia Systems

- 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

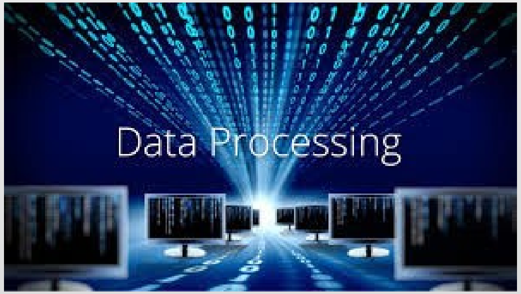
- Personal Digital Assistants (PDAs)
- Cellular telephones
- HW specialized OS
- Issues
  - Limited memory
  - Slow processors
  - Battery consumption
  - Small display screens

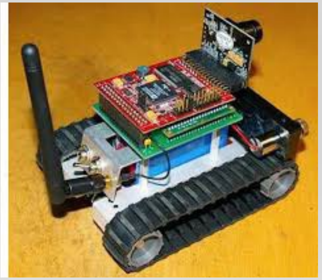
Computer Systems

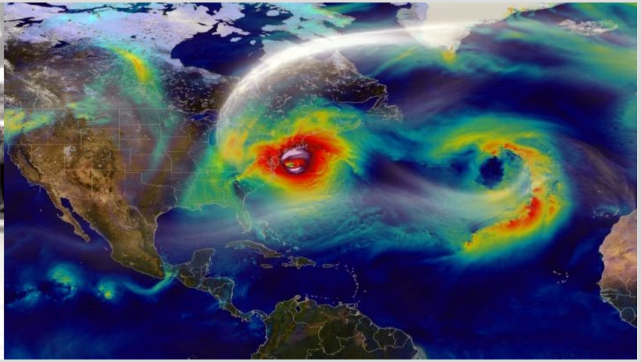


Computer Systems

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







Q & A

Thank you for your attention