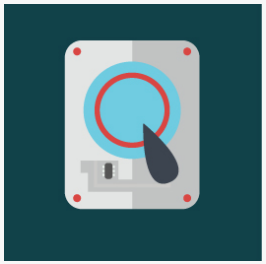
COMPS267F Chapter 2

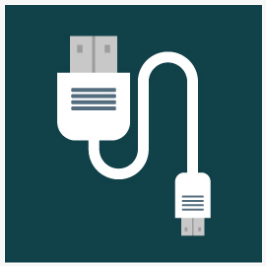
Computing Resources and Operating Systems

Dr. Andrew Kwok-Fai LUI

Aim of this Chapter

- Discusses the need of management of computer resources
  - The role of OS

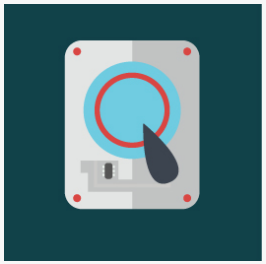


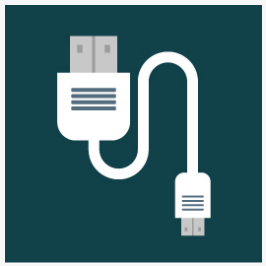


Resource Management

- Resources management is needed
  - Limited
  - Often non-sharable
  - One instance

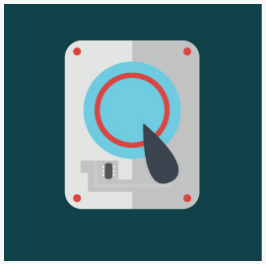
- Computing Resources belong to mainly 3 types
  - Computational
  - Storage
  - Input and Output

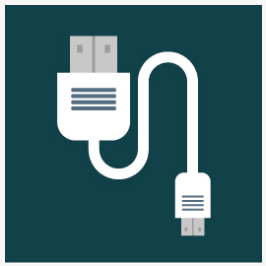




Resource Management in Multiprogramming Systems

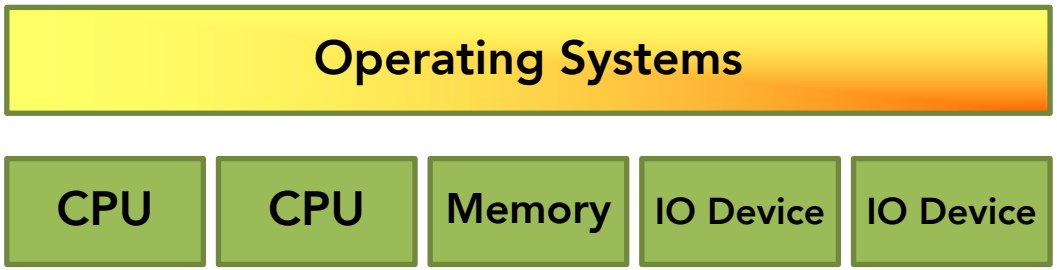
- One program running
- Multiple programs running
  - Programs compete for resources so that the execution can complete
- OS to manage the resources and allocates to different resource utilizers (user programs)





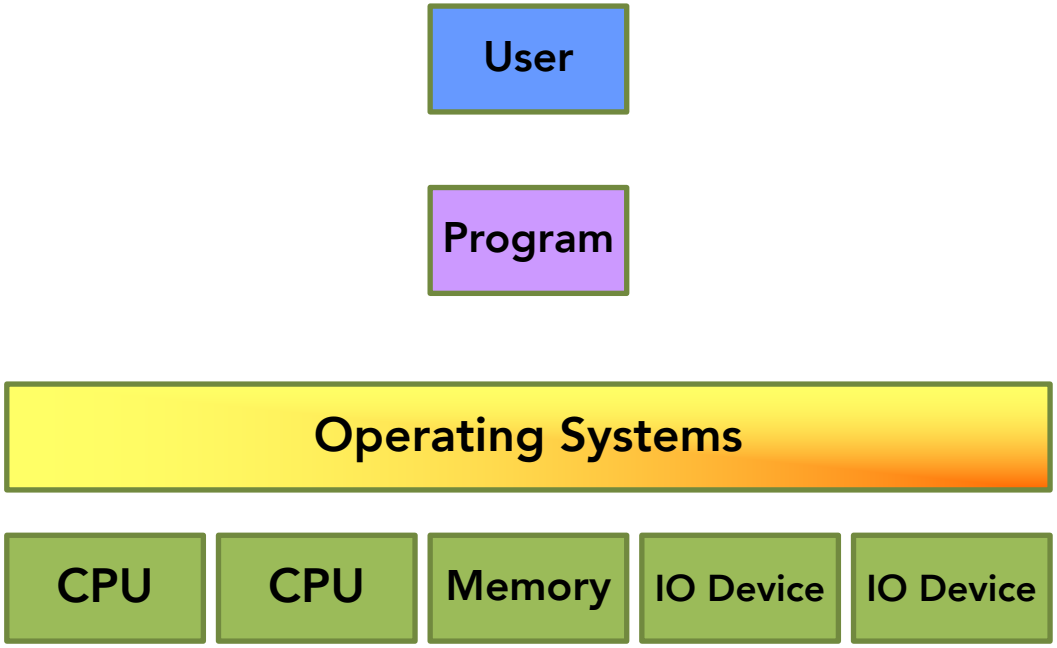
Purpose of Operating Systems

Manage Computing Resources



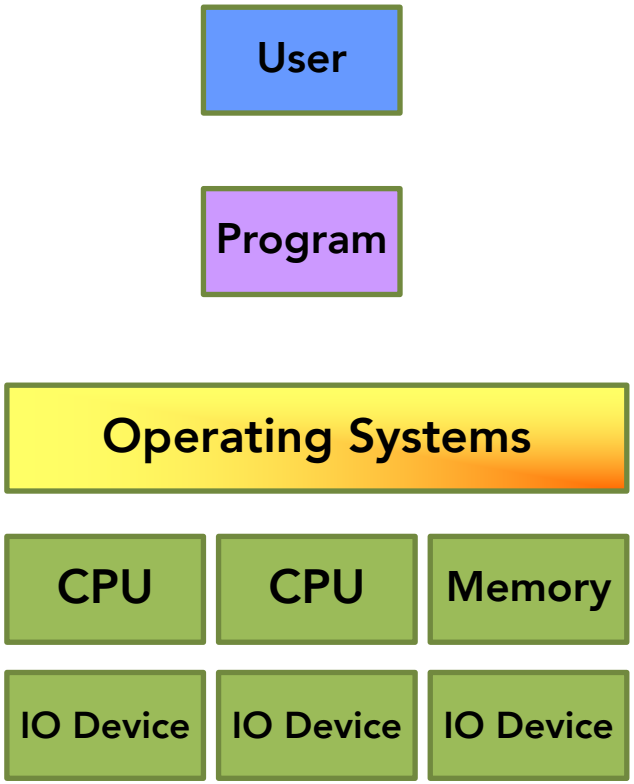
Purpose of Operating Systems

For Program Execution

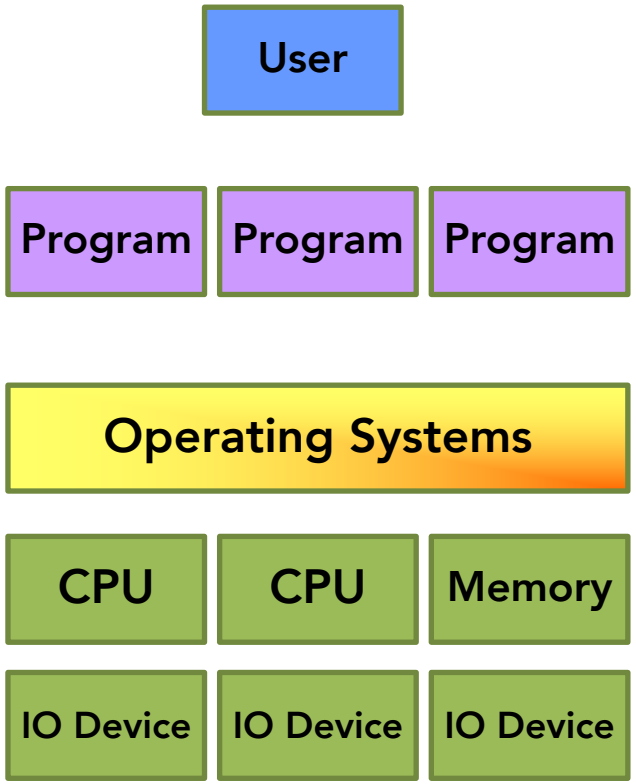


Types of Computing Systems

Single Programming

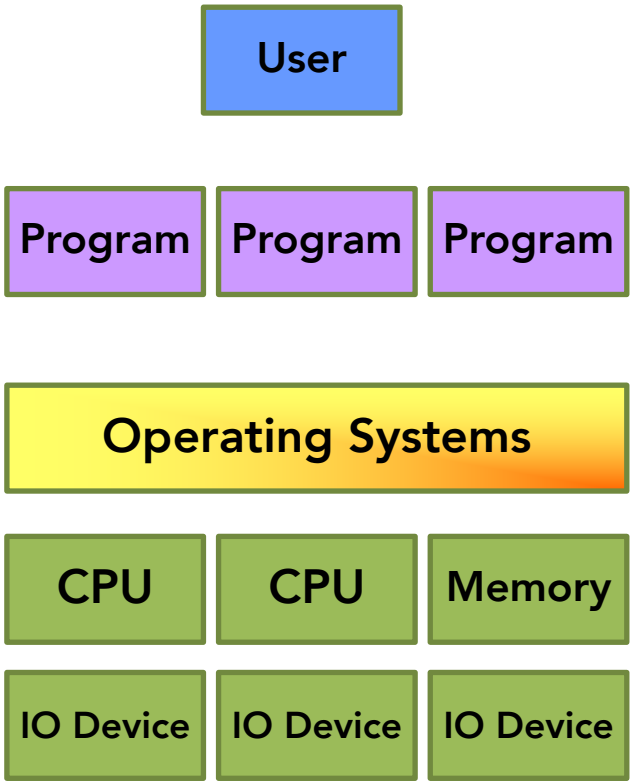


Multi-Programming

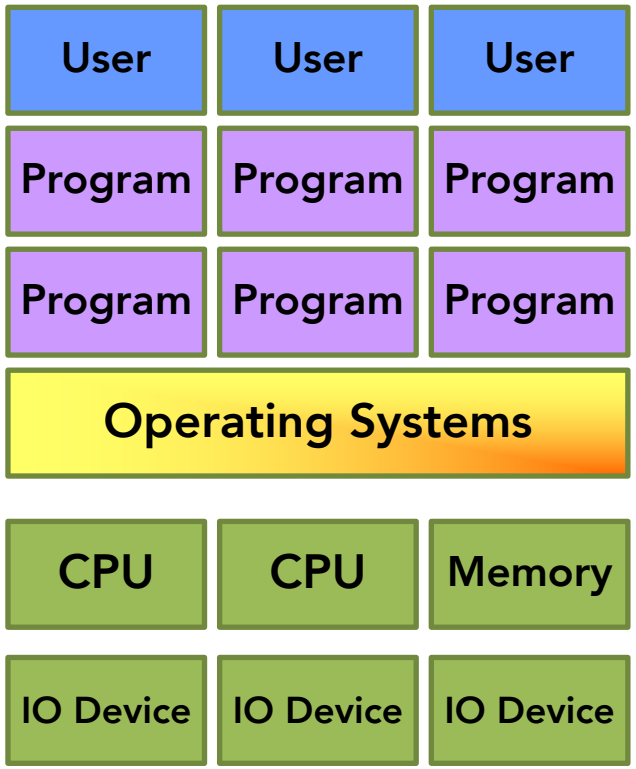


Types of Computing Systems

Single-User

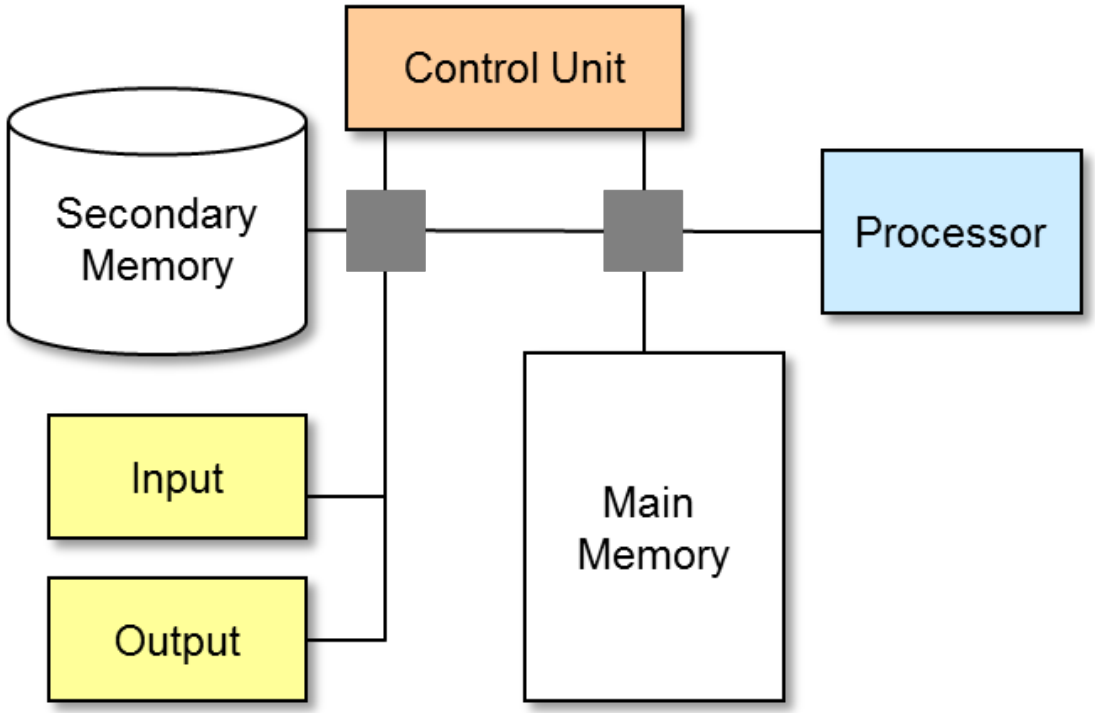


Multi-Users



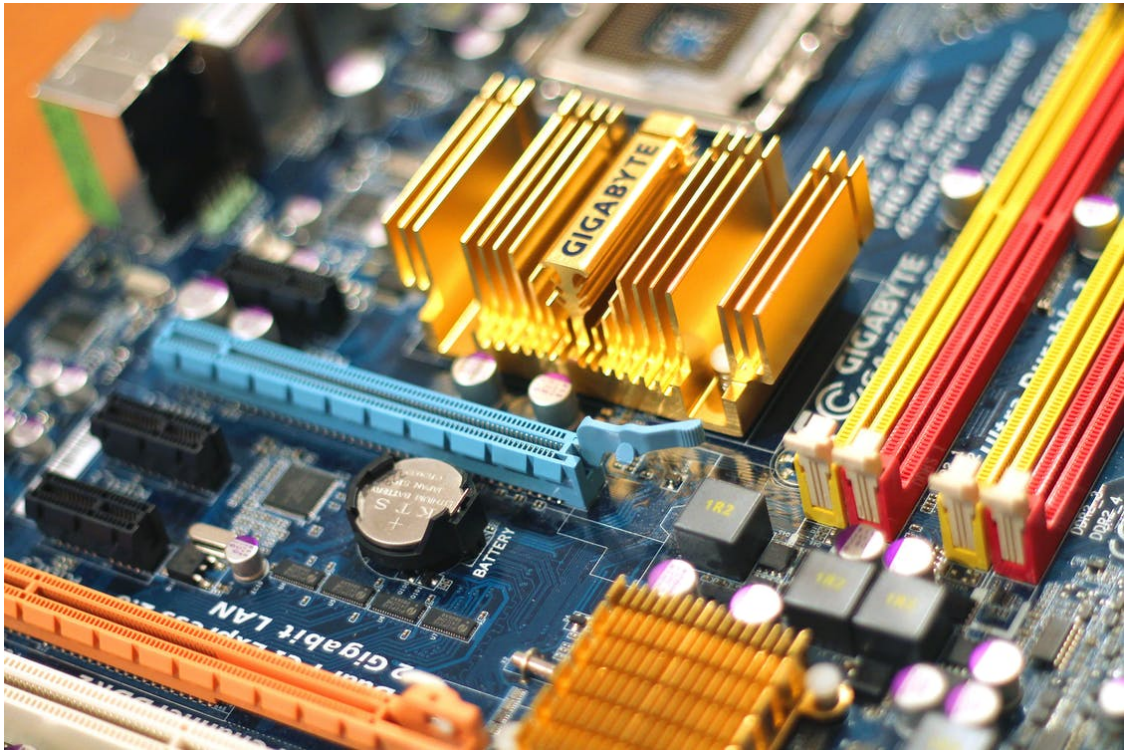
types of computer systems

Simplest Computing System



Computer Systems and Performance

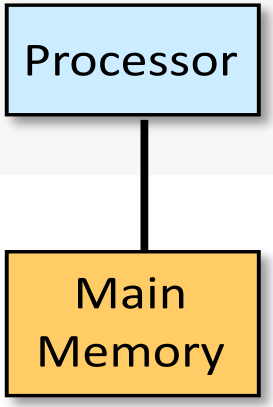
Different Architectures	Different Configurations of Processors , Memory, etc.	⇨	Different Performance Characteristics



Simplest Computing System

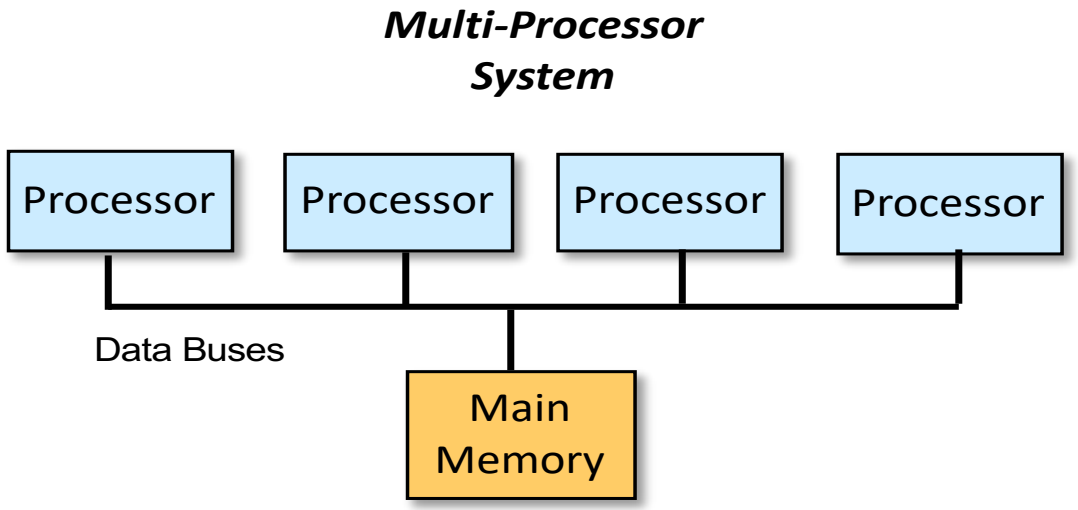
- Single processor systems
  - One processor connected to a main memory
    - Processor: executes instructions
    - Main Memory: stores programs and data

Single Processor System



Multi-Processor Computing Systems

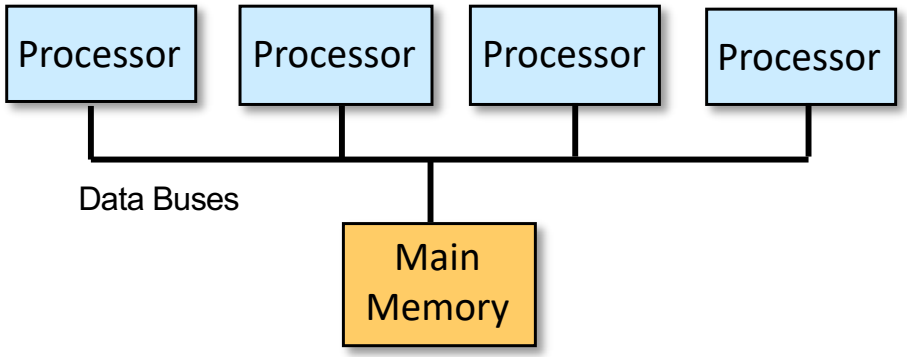
- Two or more processors
  - Executes multiple instructions at the same time
    - The instructions may come from the same program or different programs



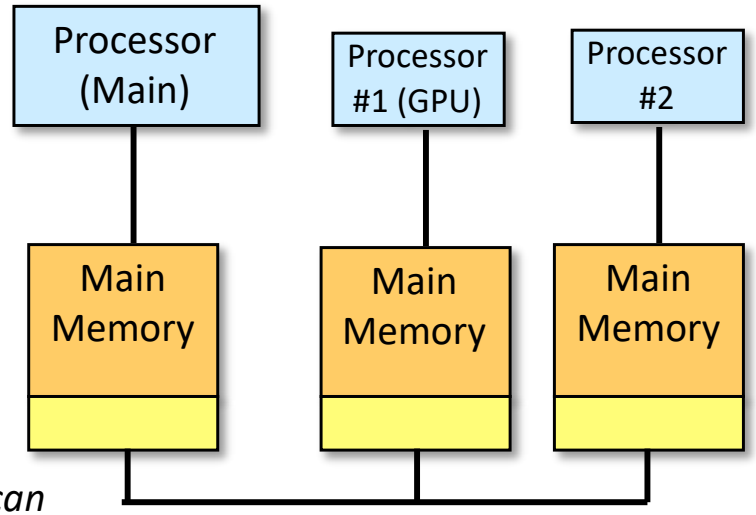
Multi-Processor Computing Systems

- Two types of multi-processor systems in terms of connection to the main memory

Symmetric Multi-Processing (SMP)



Asymmetric Multi-Processing (ASMP)



Part of the memory can be shared between the processors

Data may be exchanged through the shared memory

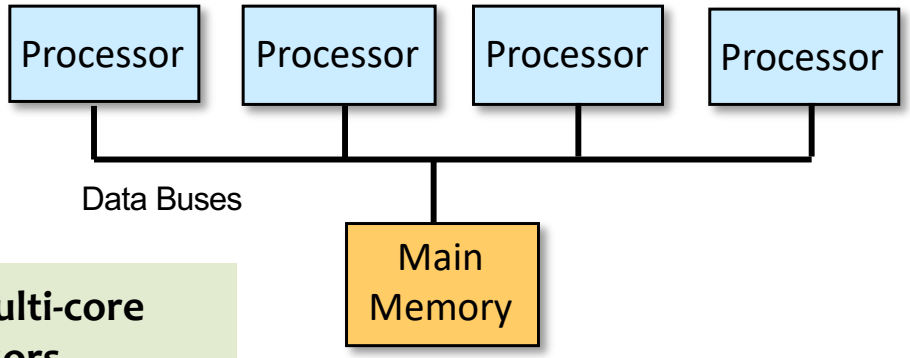
Multi-Processor Computing Systems

- Symmetric multi-processing (SMP)

Identical Processors

One OS Manages all

Symmetric Multi-Processing (SMP)



Example: Multi-core Processors

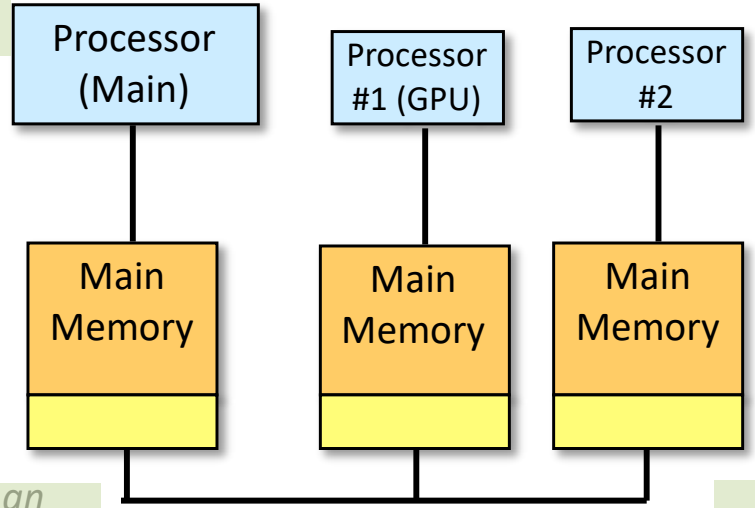
Multi-Processor Computing Systems

- Asymmetric multiprocessing (ASMP)

Specialized Processors

Example: GPU, FPU

Asymmetric Multi-Processing (ASMP)

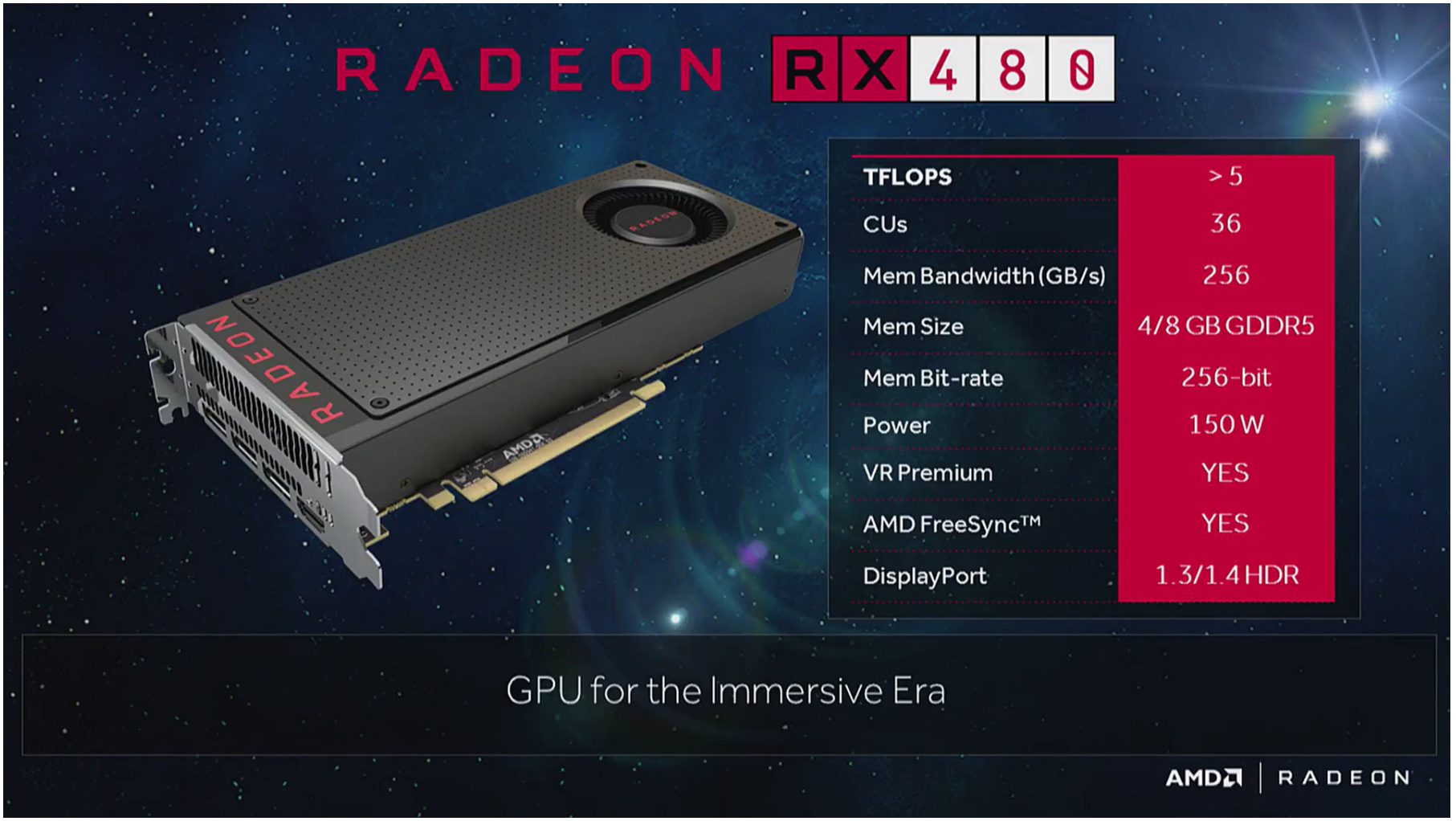


The main processor can access the main memory

Data may be exchanged through the shared memory

Other processors have their own main memory

Multi-Processor Computing Systems



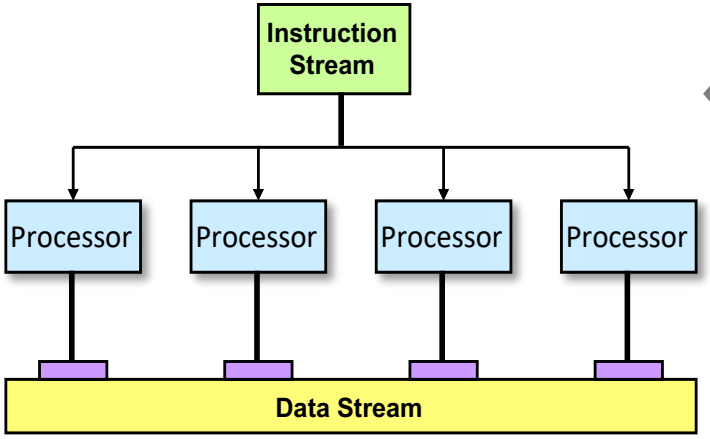
Multi-Processor Computing Systems

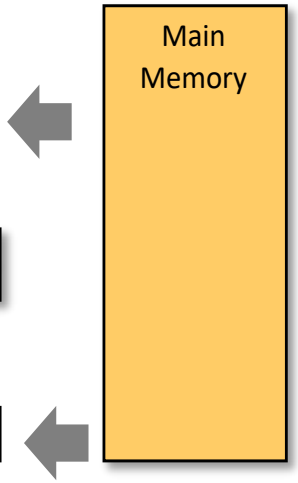
	SMP	***ASMP***
Processors	Should be of the same processor type	Can be different types of processors (i.e. CPU, GPU, etc)
Memory	One main memory system shared by all processors	Each process has its own memory. Data may be shared through specialized mechanism such as shared memory area
Operating Systems	Single operating system	Single operating system with each processor has their self-contained management process

Program Execution Models in Multi-Processor Systems

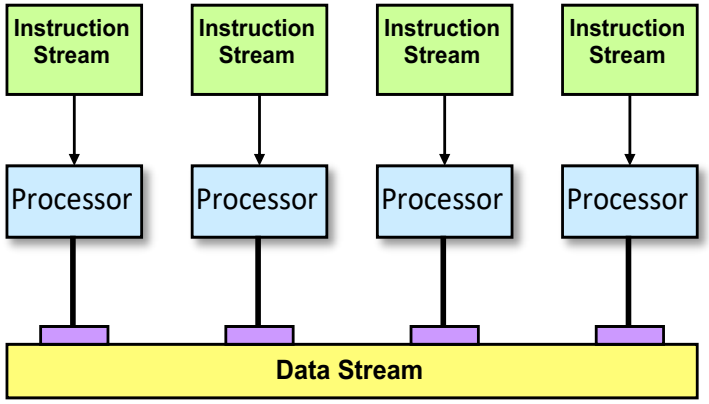
- Two types of multi-processor systems in terms of program execution models

Single Instruction stream Multiple Data stream (SIMD)





Multiple Instruction stream Multiple Data stream (MIMD)



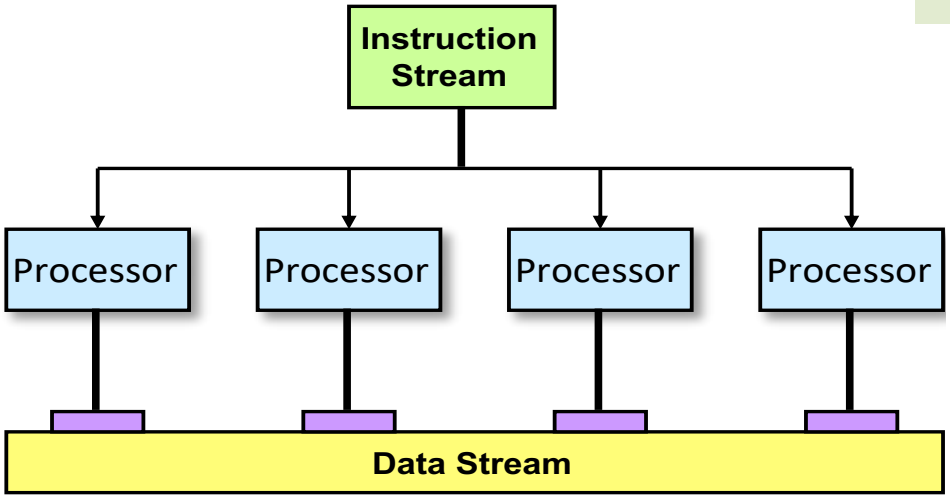
Program Execution Models in Multi-Processor Systems

- Single Instruction stream Multiple Data stream (SIMD)

Thinking Machine CM-2 has 64,000 processors

Single Instruction stream Multiple Data stream (SIMD)

Processors execute same instructions on different data



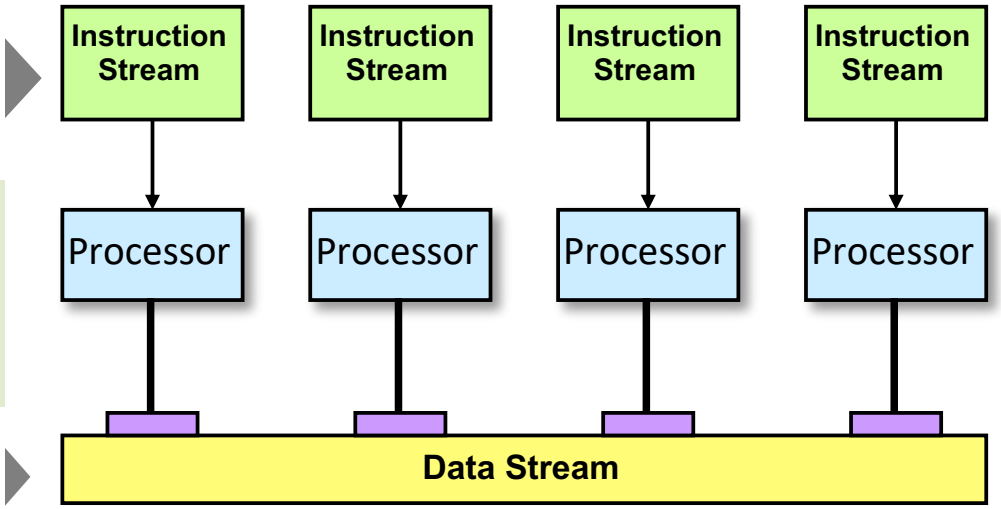
Suitable for certain problem types so called highly parallelizable problems: image processing, games

Program Execution Models in Multi-Processor Systems

- Multiple Instruction stream Multiple Data stream (MIMD)

Multiple Instruction stream Multiple Data stream (MIMD)

Executes its own instruction streams



Processors operate independently

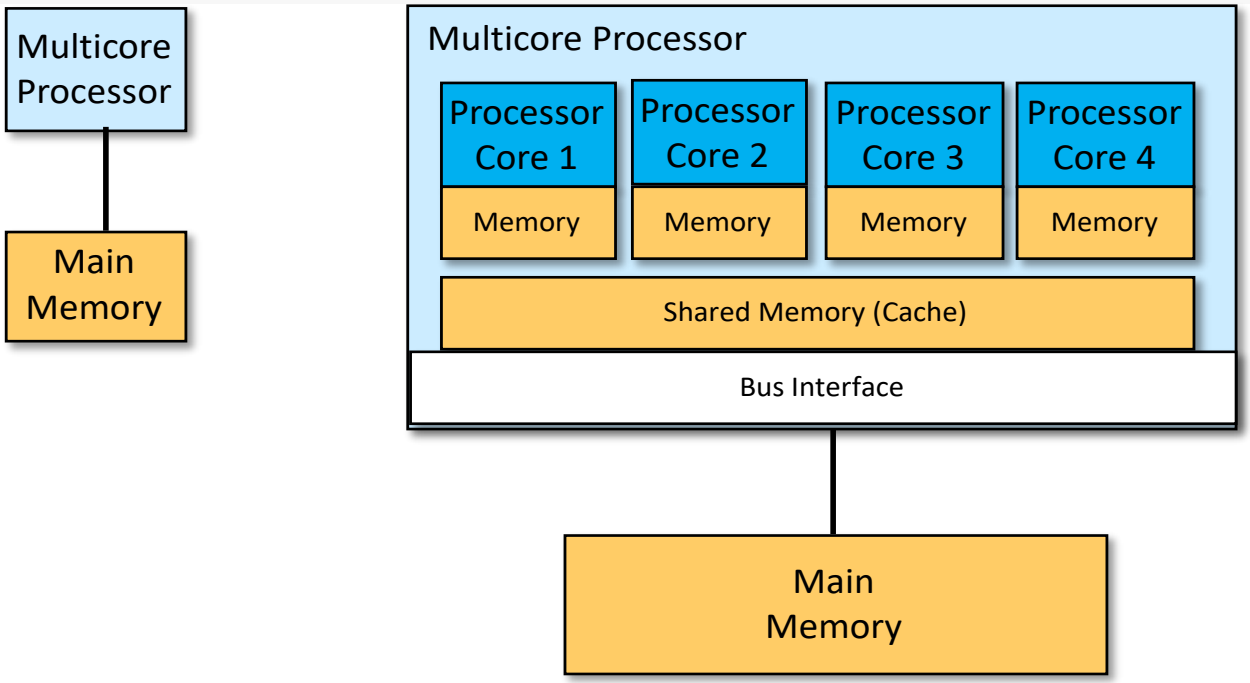
All processor can access the same memory space

The OS maintains coherency of memory

Multi-Core Processors

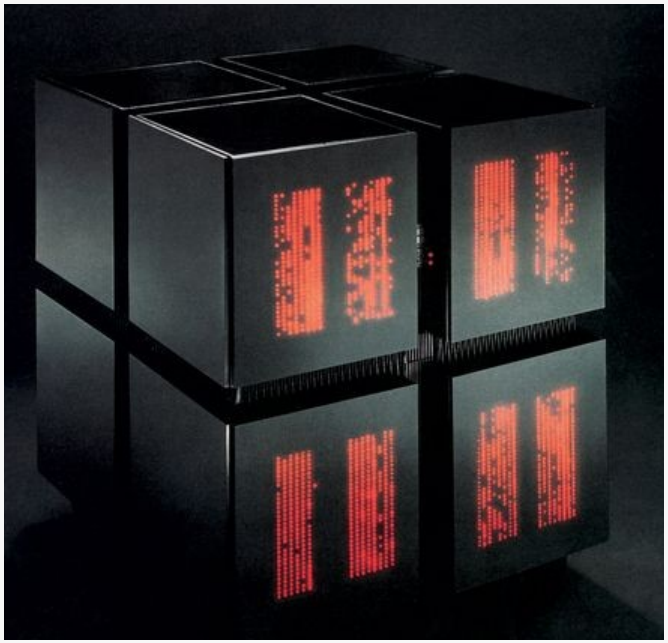
- Contain more than one processor within the same integrated chip
  - Circuitry to make components work together very efficiently

Computing System ***Computing System with a Multicore Processor***



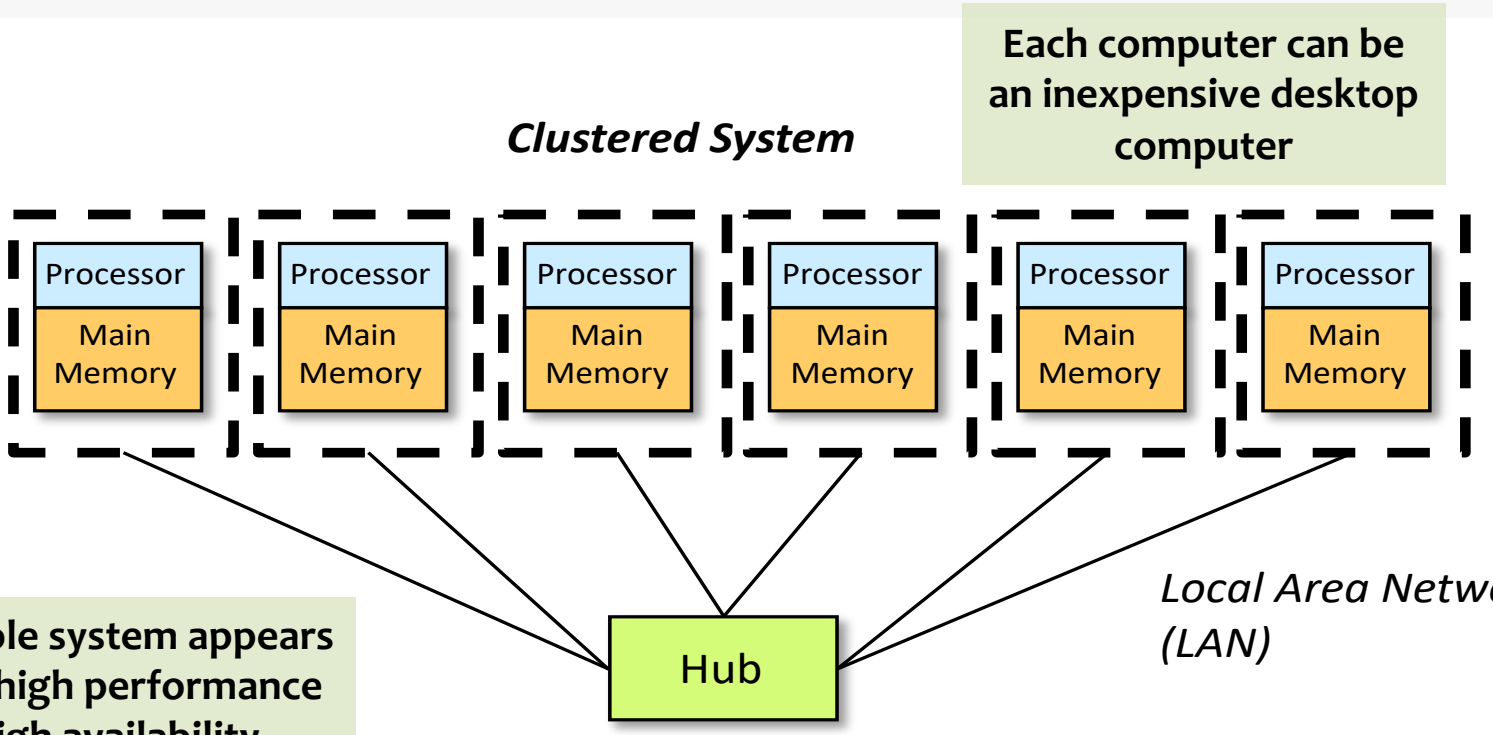
Clustered Systems

- Traditional multi-processor systems have processors and memory connected by internal bus
  - Specialized architecture to make them work smoothly
  - Harder if there are larger amount of processors
  - Costly to build



Clustered Systems

- A cost effective way is to use existing scalable technology - LAN



computing environments

Design of Operating Systems

- Computers operate in different environments
  - The usages require different performance characteristics
- Major types
  - Desktop computers
  - Mainframe computers
  - Mobile computing
  - Client-server computing
  - Peer-to-peer computing
  - Cloud computing

Desktop Computers

- Personal use for daily computing
  - Examples: web browsing, entertainment, games, simple office work
  - Not many jobs to do
  - Computing experience is important
- Performance target: Responsive to Users
  - Cheap processor is sufficient
  - Reasonably good and many I/O devices

Mainframe Systems

- Numerous tasks to complete within a period
  - Examples: banking systems to update accounts with interest, ERM to audit transactions, etc
- Performance target: Large throughput
  - Getting more jobs done
  - Large amount calculation
  - Large amount of data (transferred between storage and computation)
  - Input, output and data storage

Mainframe Systems

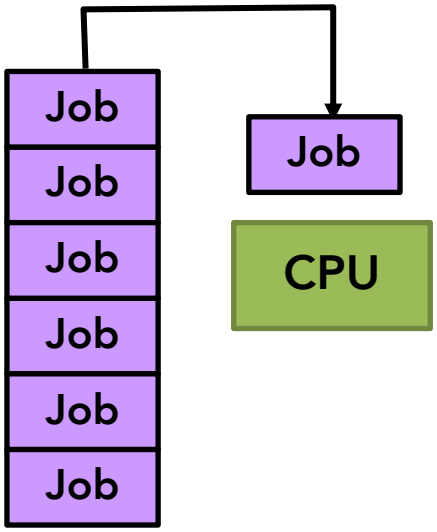
- Large-scale computer systems
  - Special housing, cooling and power

IBM 704

IBM System Z

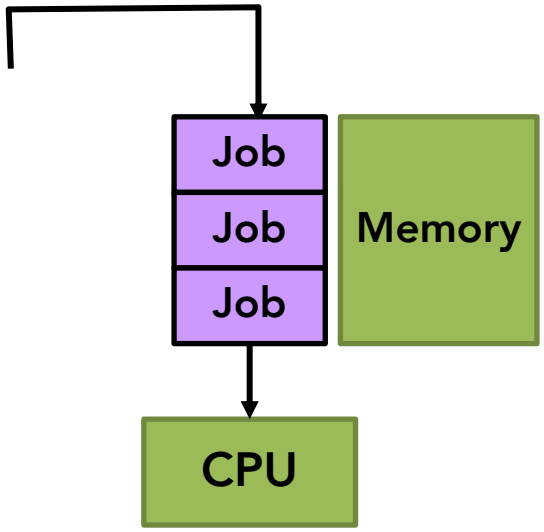
Types of Mainframes

Batch Systems



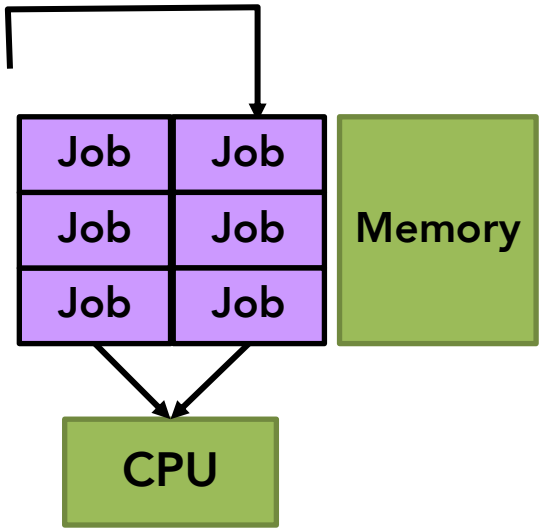
- *The Earlier Type*
- *Jobs are lined up and sent for processing*
- *The OS gives control to jobs one by one*

Multi-Programming



- *More than one program sharing the processor*
- *The programs loaded into the main memory*
- *Improve CPU utilization*
- *Need larger memory*

Time-Sharing



- *Multiple programs running at the same time apparently*
- *Needs switching from one job to another job quickly*

Mobile Systems

- Access to computing anywhere anytime
  - Examples: communication systems, installation on vehicles, personal assistants
- Performance requirements: network connectivity, fast response, good user interface
- Less powerful hardware
  - Network connectivity
  - Location awareness
  - Limited input and output
  - The OS has smaller footprint

Client-Server Systems

- A server provides services to many clients
  - A common configuration of computers working together
  - Example: the world-wide web
- Performance target (server): Large throughput and short response time
- Performance target (client): Similar to desktop or mobile usages

Peer-to-Peer Systems

- No centralized service point
  - All computers are simultaneously servers and clients
- Performance target: similar to mobile systems

Cloud Systems

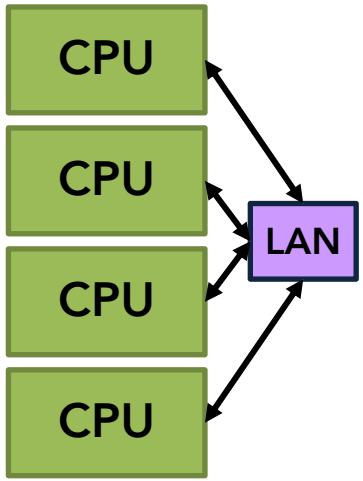
- Many jobs to be processed but mainframes are too expensive
  - Mainframes provides other features such as reliability and availability which are not needed sometimes
  - Examples: scientific and research work, simulations
- The answers are cluster systems and grid computing

Cluster Systems and Grid Computing

- Traditional multiprocessing systems are expensive
- Clusters provide a more economical solution
  - Connection of processor based on existing technology: LAN
  - Each processor has its own memory system
- Grid computing
  - Emphasis on high throughput
  - Usually based on open standard

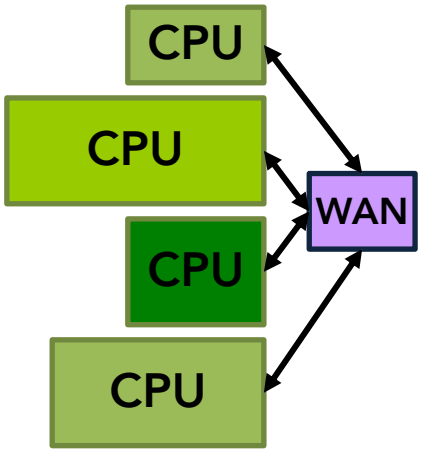
Cluster Systems, Grid Computing and Cloud

Clusters



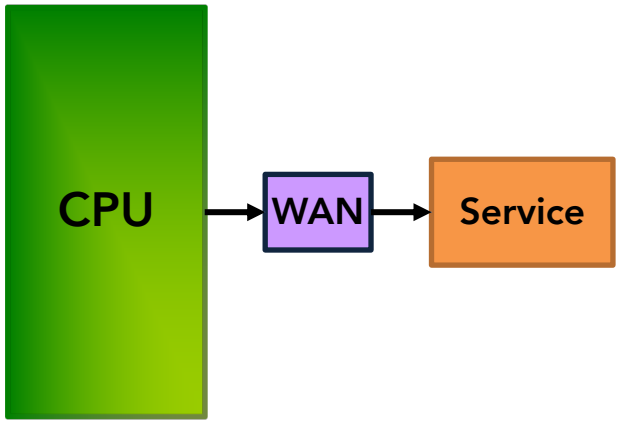
- *Tightly coupled*
- *Connected by LAN*
- *Same types of processors*
- *Local*

Grid



- *Loosely coupled*
- *Connected by WAN*
- *Various processors*
- *Non-Local*
- *Multi-parties*

Cloud



- *Service-oriented*
- *Unified interface*
- *Processors, connection not specified*
- *Single party*

Summary

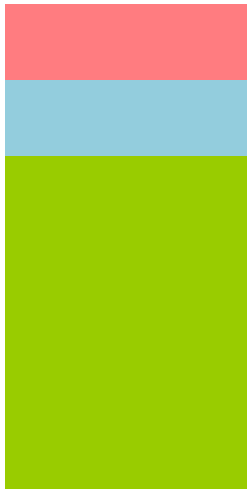
	Performance Requirements	***Hardware Configurations***
Desktop	Quick response to users Multi-programming	Flexible. Depending on user requirements
Mainframe	Large throughput	Large number of processors. Large I/O throughput for incoming and outgoing data.
Mobile	Network connectivity Quick response to users Good user interface	Flexible. Usable interface.
Client-Server	Internet connectivity Quick response to users (Server) Large throughput (Server) Multi-programming (Server)	Server should be powerful with large number of processors, large I/O throughput, and large amount of memory.
Peer-to-peer	Network connectivity	Flexible.
Cloud	Adaptable to different performance requirements Security	Similar to Servers. Flexible.

responsibilities and services of os

Responsibilities of the OS

- There are common responsibilities concerning resource management for various computing needs

Process management
Memory management
File Management
Input/Output Management
Secondary Management
Networking Support
System Protection
Command-Interpreter User Interface

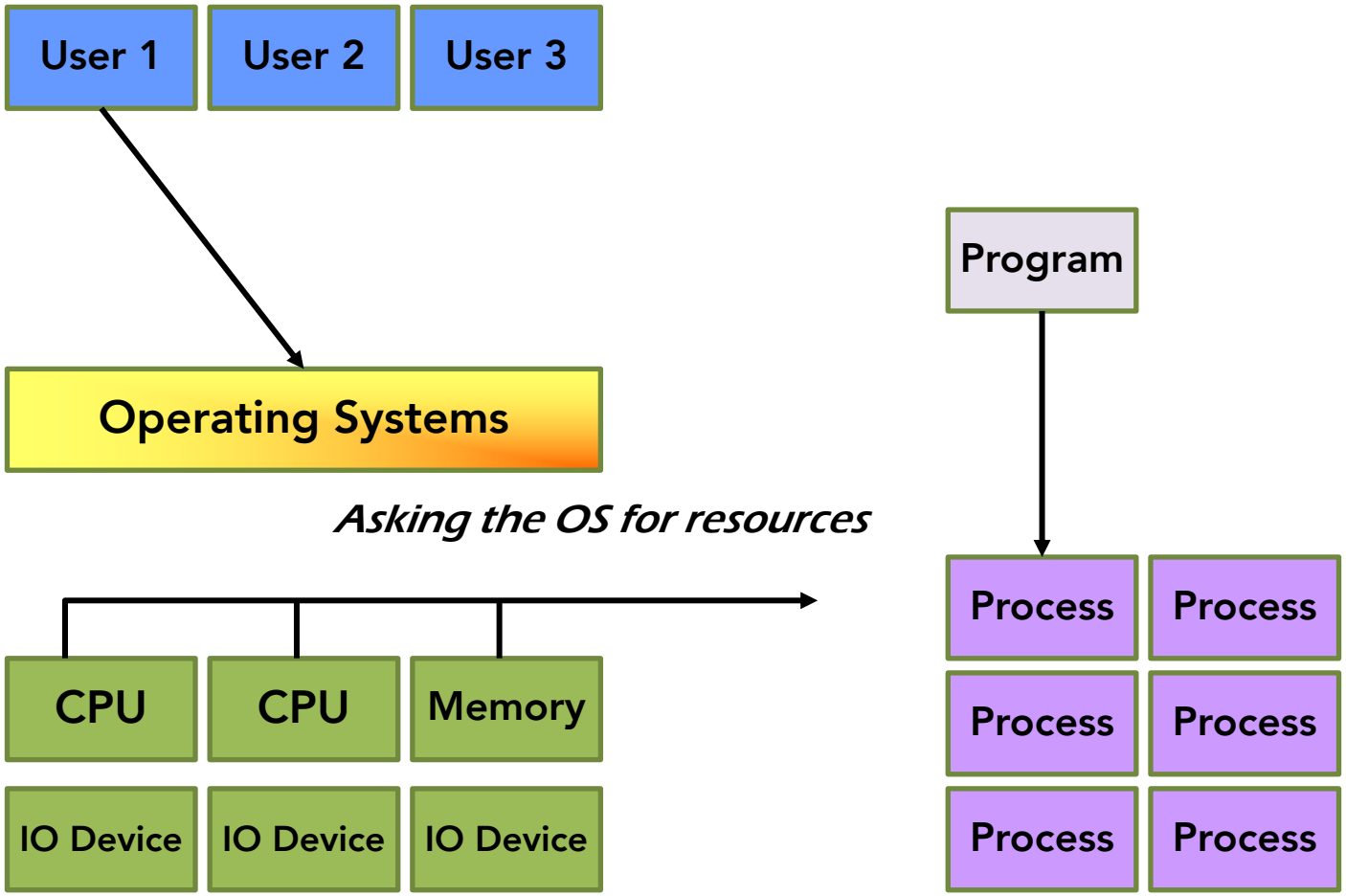


Process Management

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- A process is an executing program
  - Many processes spawned from one program
  - Has own resources
- Service provided by the OS
  - Process creation
  - Process termination
  - Process suspension and resumption
  - Inter-process coordination and communications

Resource Allocation to Processes



Main Memory Management

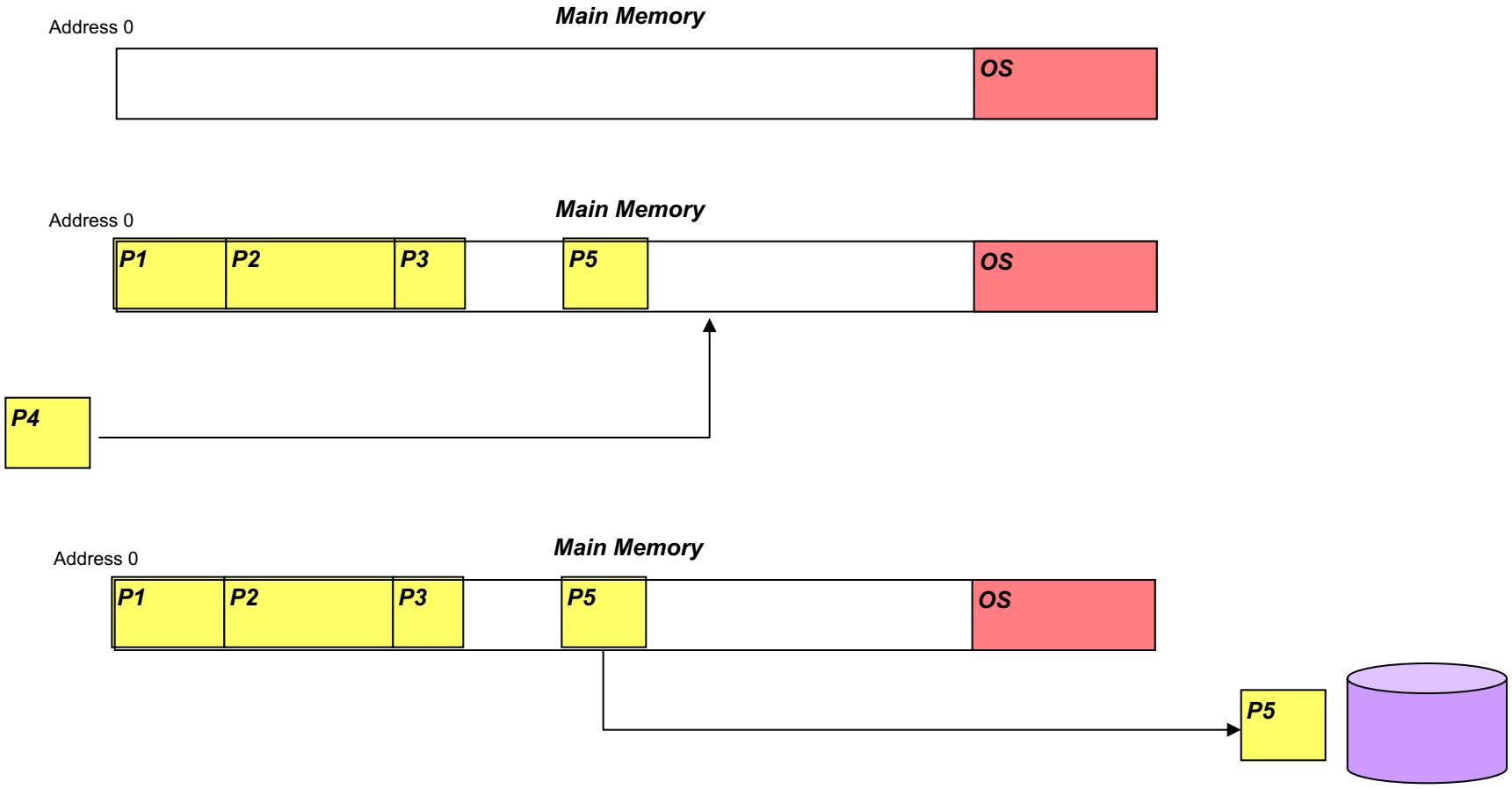
- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- CPU can directly access data in the main memory
  - Directly accessible
- The data in the main memory is addressed
  - Smallest unit is byte
  - One address for one word
    - Word size can be16, 32 and 64 bits
    - Smallest unit for read/write
  - Certain address ranges are special
    - Read only
    - Mapped to IO registers

Responsibilities of OS in Memory Management

- OS responsibilities
  - Free memory regions
  - Ownership of memory allocation
  - Allocate and de-allocate memory
    - Process creation
    - Dynamic memory allocation
  - Swapping in and out of the memory space associated with processes

Responsibilities of OS in Memory Management



File Management

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- File is a collection of data under a file name.
  - The data could in fact be program machine code.
- Files have attributes
  - Owner, permission, timestamp, and other indicators of its purposes.

File Management

- OS responsibilities
  - File creation, deletion, and attribute change
  - Directory management
  - File backup
  - File mapping of I/O devices

IO System Management

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- I/O devices are vastly different
  - Purposes
  - Properties: speed, control and access, data transfer
- Difficult to deal with disparity of devices
  - Challenging to write programs and control them

IO System Management

- A uniform interface hides away the differences
  - File/Stream is often the interface presented to programmers
  - Network connection, file on secondary devices, other device drivers are all access through files/streams

IO System Management

- OS responsibility
  - Uniform device driver interface
  - Specific drivers for I/O devices
  - Buffer for I/O devices

Secondary Storage Management

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- Hard disk is the major secondary storage used in current computer systems.
  - Supports the main memory by providing a backup
- OS is responsible
  - Free space management and storage allocation
  - Disk scheduling
  - Performance management

Networking Management

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- A modern computer needs connection to the Internet
  - Data can be shared and obtained.
- Distributed systems
  - Sharing the burden of executing a program between computers
  - Heterogeneous computers connected together in a single logical system

Protection System and Security

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- Control access of processes and users to various users and system resources.
  - Authorization, authentication, and auditing form the pillars of system security
  - Example: File access and execution.

Command Interpreter System

- Process management
- Memory management
- File Management
- Input/Output Management
- Secondary Management
- Networking Support
- System Protection
- Command-Interpreter User Interface

- A user interface
  - Executing programs and managing resources on the computer systems
  - Provided to programmers to get easy access to system resources
  - Read commands and interprets control statements, executes them
- Another name is shell

services provided by OS

Services provided by the OS

- Services are related to an environment for execution of programs
  - System calls
  - System programs
  - Coordination of process execution

System Calls

- An interface between running process and the services of operating systems
  - Call directly from a higher-level language programs – WIN32 API, .NET, ANSIC.
  - Abstraction of a set of procedures
    - Making one system call performs some procedure

System Calls

- Types of system calls
  - Process management
  - File management
  - Device management
  - Information maintenance
  - Communication between processes and across computers
- System calls are similar on common OS
  - Example: open, read, write, close, wait, exec, fork, exit, and kill
  - Linux has 319 system calls
  - FreeBSD has almost 330 system calls

System Programs

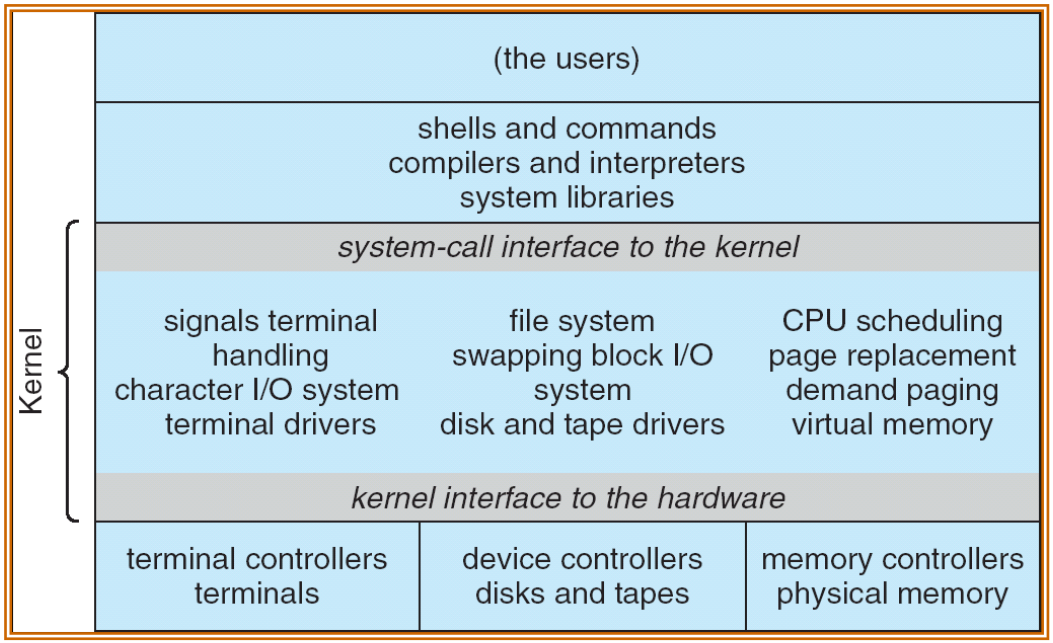
- System programs exist as files
  - User interface to underlying system calls
  - The command interpreter is the most important system program

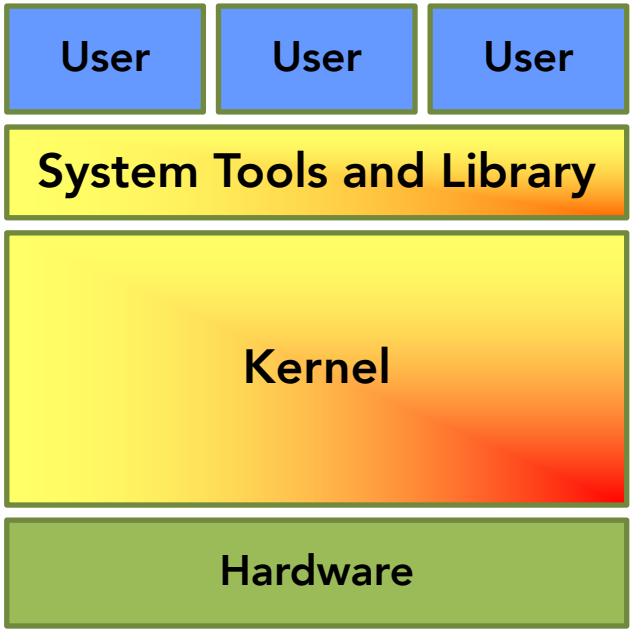
structure of os

Structure of OS

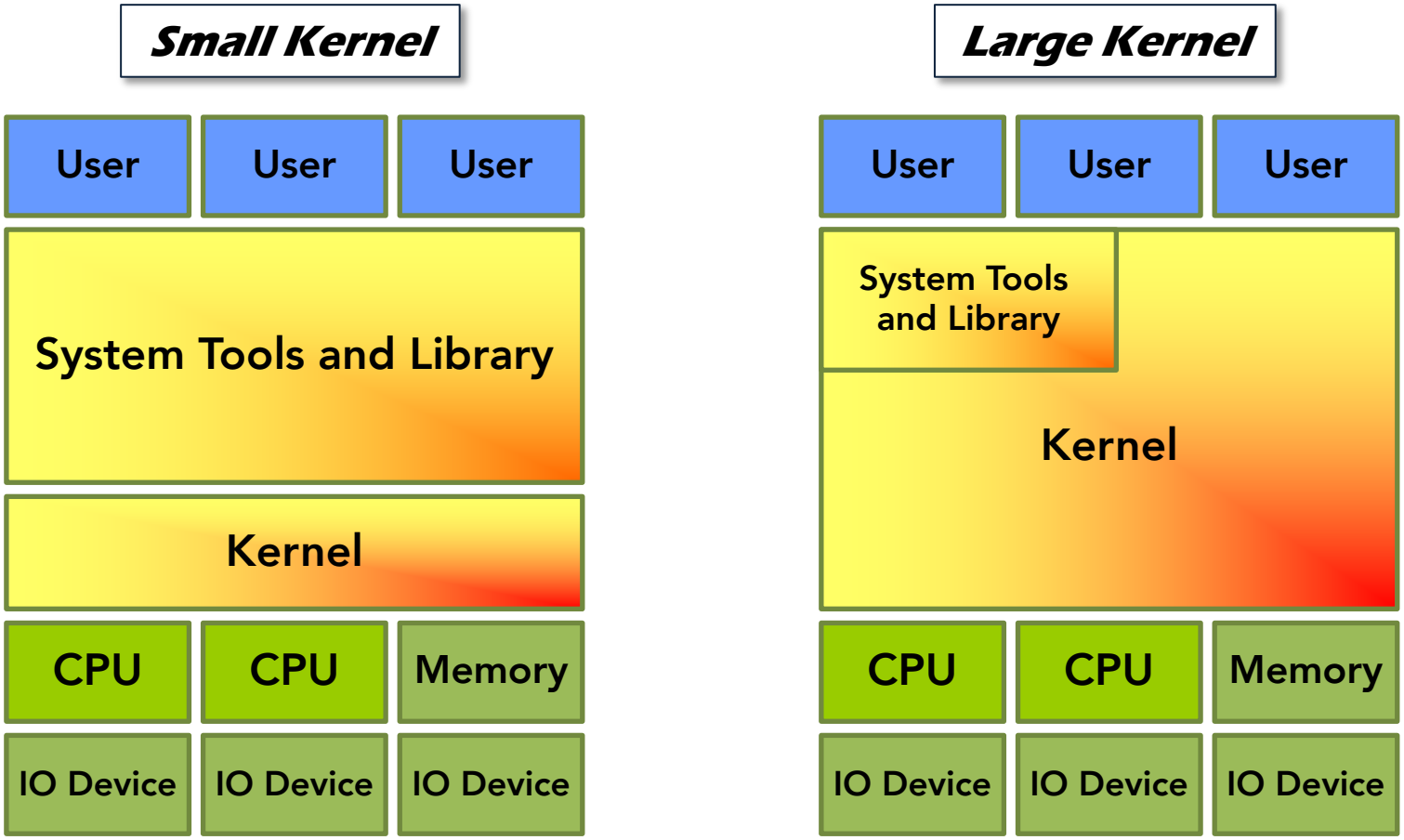
- The OS is a large system
  - Design practices applicable to large system development are also suitable to OS development
  - Conventional and modern software engineering practices
    - modular approach, layered approach, component-based approach
  - Layered approach is common
    - Stepwise development from the bottom up and better portability

System Programs





Layered Architecture



Which design is more efficient?

Which design is more resilience to change?

Micro-Kernels

- Kernel is the core of the operating system services.
  - Some OS designs supports many functionality and makes the kernel large and becoming difficult to manage
  - One approach is to reduce the function of kernel to bare minimal
- Micro-kernel approach
  - Minimizes the function suite to the essential minimal
  - Moves other programs to the system level and application level

SysGen

- System Generation (SYSGEN) is a process
  - Configures an OS for the use on a computer
    - Often interactively.
  - SYSGEN reads the files from the CDROM/USB/Storage and moves the software on the secondary storage on the computer
- SYSGEN probes the configuration of the computer for information of what devices are used and their attributes
  - CPU, memory, peripheral devices, and also user preferences

Acknowledgement

- Images and Graphics used in this set of slides
  - Designed by Photoroyalty - Freepik.com
  - https://www.pexels.com
  - https://www.flaticon.com