Operating systems

Favorite	□
Archived	□
Fleeting	□
Area/Resource	
Project	
Type	
Review Date	
Image	
URL	
Created	@February 8, 2023 10:48 AM
Updated	@February 14, 2023 1:14 PM
Root Area	
Project Area	
Updated (short)	02/14/2023
Pulls	
Resource Pulls	
Project Archived	
URL Base	
🍳 Recipe Divider	🥗🥗🥗 RECIPE BOOK PROPERTIES 🥗🥗🥗
🍳 Recipe Tags	
📚 Book Divider	📚📚📚 BOOK TRACKER PROPERTIES 📚📚📚
📚 Author	
📚 Date Started	

📚 Date Finished	
📚 Book Status	
📚 Rating	

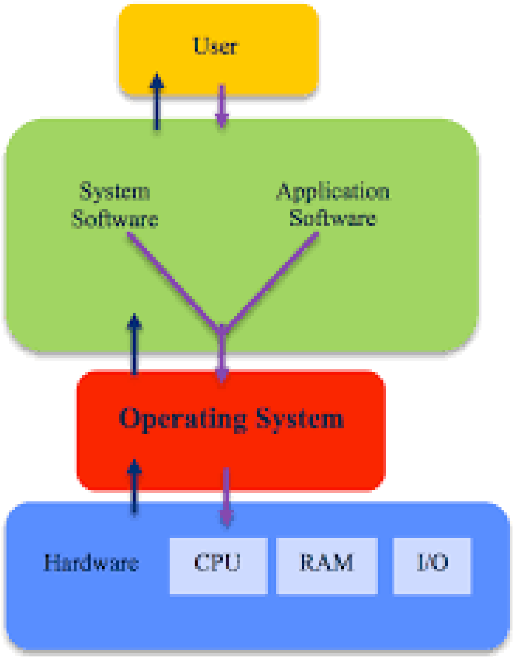
> 💡 This is the operating systems course offered by University of Massachusetts. The course outline and topics include:

- Processes and Threads
- Memory Management
- Storage and File systems
- Distributed systems

Introduction

What is an operating system?

> 💡 An operating system is a collection of software that manages computer hardware resources and provides services for computer programs. The OS acts as a bridge between the computer hardware and the applications running on the computer, providing a user interface and executing user-issued commands and instructions. The OS handles the complex instructions and operations needed for an app to interact with the hardware of a computer. Abstracts away a lot of complexity for users to have an easier time.



💡 Before diving into operating systems it is important to have a general understand of some key hardware concepts.

Hardware basics

Types of OS:

- Batch OS
- Time sharing OS
- Distributed OS
- Network OS
- Real Time OS
- Multi Programming/Processing/Tasking OS

Functions of OS

- Interface between User and Hardware

- Allocation of Resources
- Management of Memory, Security, etc

> 💡 How can the resources available be managed in a way that everything is functioning smoothly.

Goals of OS

- Convenience
    - Convenient to use
- Efficiency

> ✏️ Efficiency in terms of an Operating System is the ability of the system to allocate resources in an optimal manner allowing for tasks to be handled effectively, and efficiently. An efficient OS can handle tasks quickly, handle multiple tasks simultaneously and minimize the consumption of resources while providing a good user experience.

- Both

Important terms

Bootstrap program

> 💡 The initial program that runs when a computer is powered up or rebooted. It is stored in the ROM (Read Only Memory). It must know how to load the OS and start executing that system. It must locate and load into memory the OS kernel.

- The kernel is the heart of the OS

Interrupt

> 💡 The occurrence of an event is usually signaled by Interrupt from Hardware or Software. Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by the way of the system bus.

System call (monitor call)

> 💡 Software may trigger an interrupt by executing a special operation called system call. If the software is triggering an interrupt it is called a system call.

What happens when a CPU is interrupted

> 💡 When a CPU is interrupted it stops what it is doing and immediately transfers execution to a fixed location. The fixed location usually contains the starting address where the service routine of the interrupt is located. On completion, the CPU resumes the interrupted computation.

Service routine: Every time an interrupt is triggered it is usually to trigger another process, the service routine is where the instructions of the interrupts process is located, so that the CPU can execute.

I/0 Structure

> 💡 Input output devices are the devices used to either giving input or giving output in a computer. Storage is one of many types of I/O devices within a computer. A large portion of operating system code is dedicated to managing I/O, both because of it’s importance to the reliability and performance of a system and because of the varying nature of the devices.

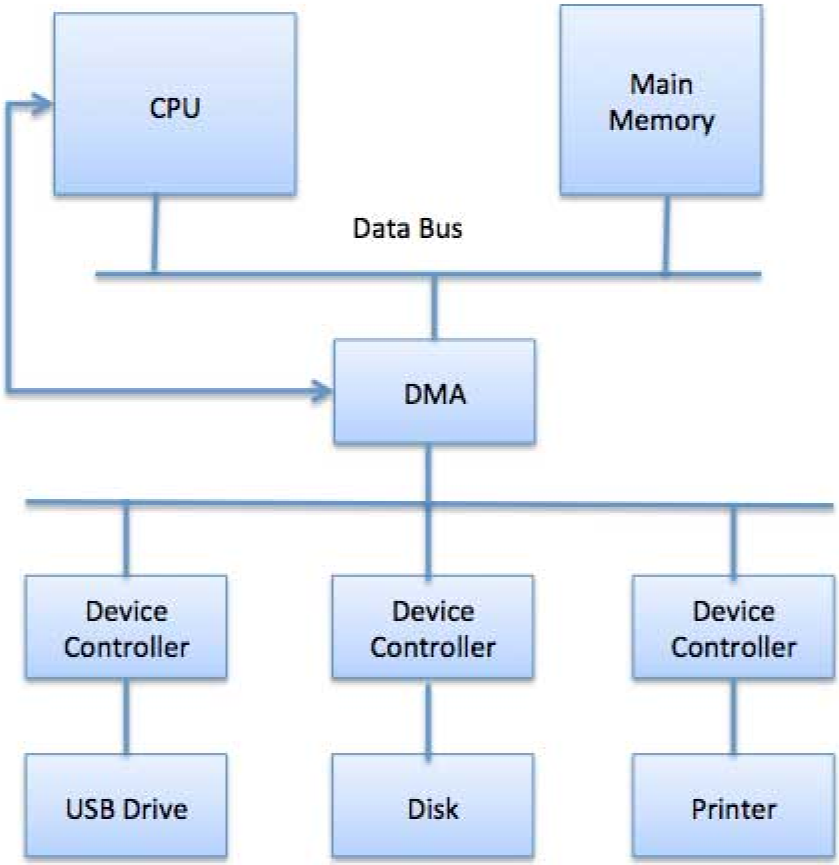
> ⚙️ A general purpose computer consists of one more CPUs and a number of I/O devices that communicate through a common bus.

Common bus

> ✏️ A shared common pathway which allows information to be transfered between devices within a computer.

Device controller

> 💡 Each device controller is in charge of a specific type of device. It mains a local buffer storage and a set of special purpose registers.



> 💡 Device controllers are a key component in hardware systems that allow the CPU to communicate with devices through the message bus. The device controller will allocate local buffer storage and make use of a register which is a smaller faster version of RAM that is located within or in close proximity to a device. Usually used to storage short term necessary information for the operation to continue.

Local buffer storage

> ✏️ Local buffer storage is a main memory storage that is allocated to a certain program or hardware. In the case of device controllers, buffer storage is allocated to always ensure there is sufficient memory to allow a device to function properly.

Registers

> ✏️ In the case of devices, the registers are used to contain the information specifying what actions need to be made to fulfill the operation inputted by the I/O device. It will store data addresses etc…

Device drivers

> 💡 The question arrises: How can an operating system allow a user to communicate with a device and allow the device to interact with the OS. A device driver is dedicated to each device. The device driver understands the device controller and presents a uniform interface to the device to the rest of the OS.

Working of an I/O operation

1. The device driver loads the appropriate registers within the device controller

2. The device controller in turn examines the contents of these registers to determine what action to take
3. The controller starts the transfer of data from the device to it’s local buffer
4. Once the transfer the of data is complete, the device controller informs the device driver via an interrupt that it has finished it’s operation
    a. Interrupts contain information, specifying how to address an interrupt
5. The device driver then returns control to the operating system

> ⚠️ This form of interrupt-driven I/O is fine for moving small amounts of data but can produce high overhead when used for bulk data movement. To solve this problem, **Direct Memory Access (DMA)** is used. Else the CPU is always being interrupted.

> ✏️ **Overhead:** Refers to the additional resources and processing time needed to perform a task. This additional resource is not directly atrributed to the execution of the task itself, but instead to the maintainance and managing the system used.

- Managing storage for many files
- Context thread switching

Direct Memory Access

> 💡 Since it is a major objective in hardware and OS design to reduce overhead for highly efficient systems, the strategy of continuously sending interrupts to the CPU to handle I/O device operations is not viable. Direct Memory Access is a strategy that allows device controllers to communicate directly with an allocated part of the main memory. This reduces overhead as it allows the transfer of information across devices without interrupting the CPU. While the device operation is occuring the CPU can perform other tasks.

Computer System Architecture

Categorize computer systems based on the number of General purpose processors:

1. Single processor systems
2. Multiprocessor systems
3. Clustered systems

Single processor systems

> 💡 One main CPU capable of executing a general purpose instruction set including instructions from user processes. Other special purpose processes are also present which perform device specific tasks. For devices such as keysboard there may be microprocessors to handle inputs. This is called a single system due to there being only one general purpose processor.

- In order to convert keystrokes to code, there is a microprocessor present on your keyboard to convert keystroke to code to be inputted into the system

Multiprocessor systems

Also known as parallel systems or tightly coupled systems

> 💡 Has two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices.

Advantages:

- Increased throughput
  - The amount of data that can be transfered from one location to another
- Economy of scale
  - Different processors sharing the resources within a sytem. It is more economic

  - Meaning that as the system increases in size the cost per unit of output decreases
- Increased reliability
  - A multiprocessor systems provides redundancy, as it would allow a processor to fail, and the work can be picked up by other processors

> ✏️ Throughput refers to the amount of work that can be completed over a period of time. It is a measure of a systems performance and efficiency. An example is the number of read/writes over a second, or amount of data that can be transfered over a network in a second.

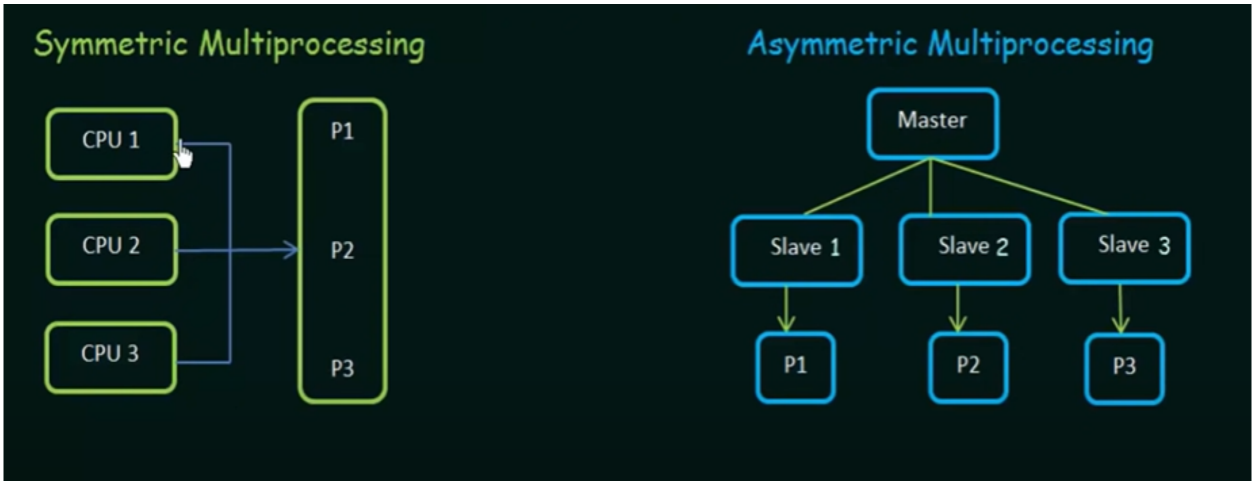
Types of multiprocessor systems

Symmetric Multiprocessing

> 💡 All CPUs are involved in performing a single task at a time. This concentration of processors significantly increases processing power which could increase the speed of performing tasks. It is scalable as you can easily add additional processors to the system. The disadvantages are this is a complex system, which increases in complexity as the number of processors increase, and can also lead to resource contention.

Assymetric Multiprocessing

💡 One of the processors will act as a master and the others will be slaves. The master processor will monitor the slaves and assign tasks or processes to processors. If something fails the master will take care of how to take care of the failure and distribute the load. This allows systems to handle different tasks with different needs. It is also cheaper to implement than a symmetric multiprocessing system. Disadvantages include resource underutilization as some processors will be assigned to tasks that do not require their full capacity, and the system is limited in scalability as it may require more hardware and software changes to add processors.



Clustered systems

💡 Similar to multiprocessor systems, clustered systems gather together multiple CPUs to accomplish computational work. The major difference is that they are composed of two or more individual systems coupled together. The complete system is coupled together, which forms a cluster.

- Provides high availability
  - More than one system which provides redundancy
- Can be structured asymmetrically or symmetrically

- Asymmetric: One machine in hot-standby mode (master), others run applications
- Symmetric: Two or more hosts run applications, and they monitor each other

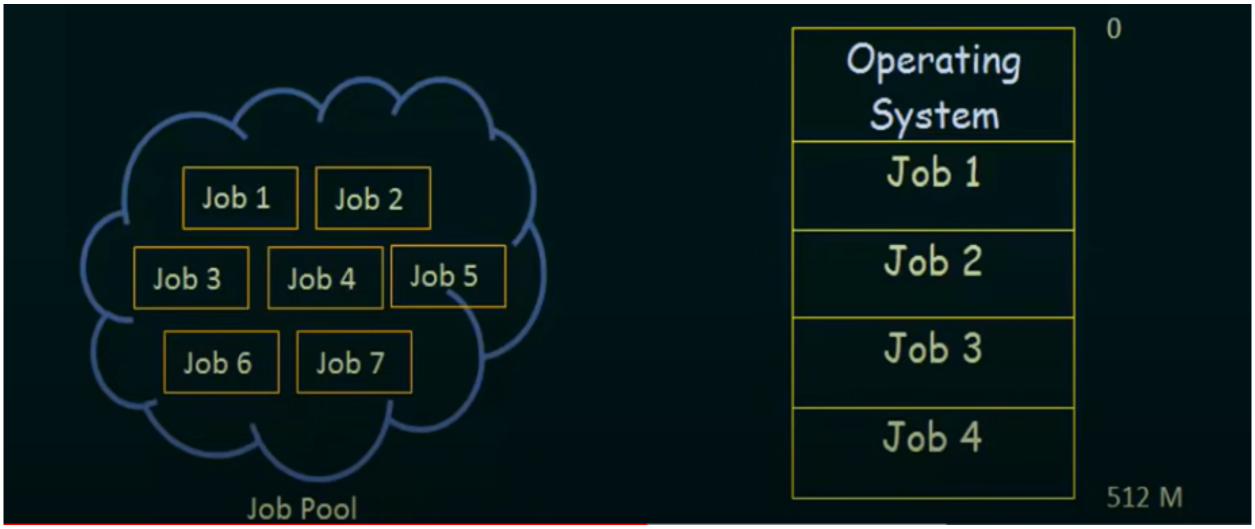
Operating system structure

Operatinf systems vary greatly in their makeup internally.

💡 Every OS has two commonalities: Multiprogramming & Multitasking (timesharing)

Multiprogramming

💡 The capability of running multiple programs by the CPU. A single user cannot, in general, keep either the CPU or the I/O devices busy at all times. Multiprogramming increases CPU utilization by organising jobs so that the CPU always has one to execute.



⚙️ A job is something to be execute which can contain code and data. All jobs are kept in a job pool. There is a memory layout within a multiprogramming system. All jobs can’t be loaded into the memory as it is limited. Without multiprogramming when a job is launched the CPU will only be available if and when the job is finished. Additionally, if the job requires the use of another device the CPU will become available but will remain idle. A multiprogramming system ensures that when the CPU is idle, it is being used to execute the next job.

- Multiprogram systems provide an environment in which the various system resources are utilized effectively, but they do not provide for user interaction with the computer system.

Time sharing (Multitasking)

💡 CPU executes multiple jobs by switching among them. The switches occur so quickly that the users can interact with each program while it is running. Time sharing requires an interactive computer system which provides direct communication between the user and the system. A time shared operating system allows many users to share the computer simultaneously.

- Since users take very long in terms of processing speeds, the time gap allows for time shared systems to switch jobs very quickly which gives the impression to a user that the system is completely available to them
- The system uses CPU scheduling and multiprogramming to provide each user with a small portion of time-shared computer
- Each user has at least one seperate program in memory

💡 A program loaded into memory and executing is called a **Process**

📝 An example of a time sharing system can be a server which needs to serve many requests simulatenously, or a cloud computer platform which needs to utilize the same resources to handle many user interactions.