CSC-203 Operating Systems[3]

Lecture: 1

Instructor: Sanjog Sigdel

Date: May 2, 2019

UNIT 1

Outlines

Historical Background

- Operating system evolution
- Hardware review
- Operating system structure
- Overview of operating system: batch system, multiprogramming, time-sharing, real-time, mainframe operating systems, personal computer operating systems, system calls.

UNIT 1

Historical Background

A modern computer consists of:

- One or more processors
- Main memory
- Disks
- Printers
- Various input/output devices

Managing all these components requires a layer of software – **the operating system**

Historical Background

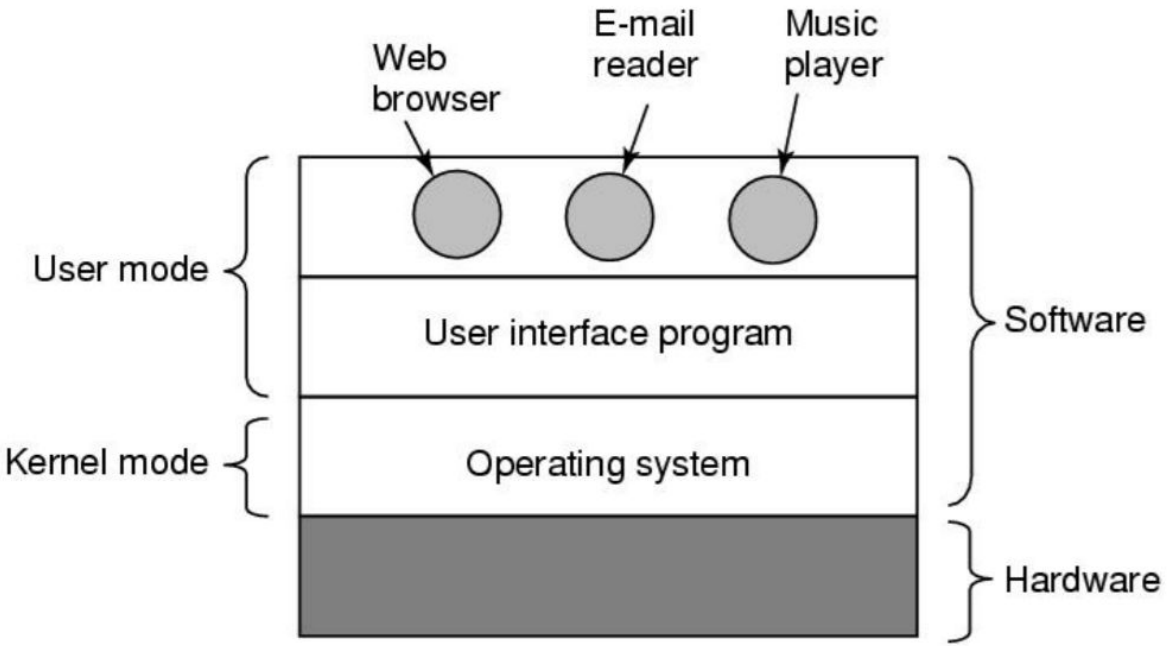
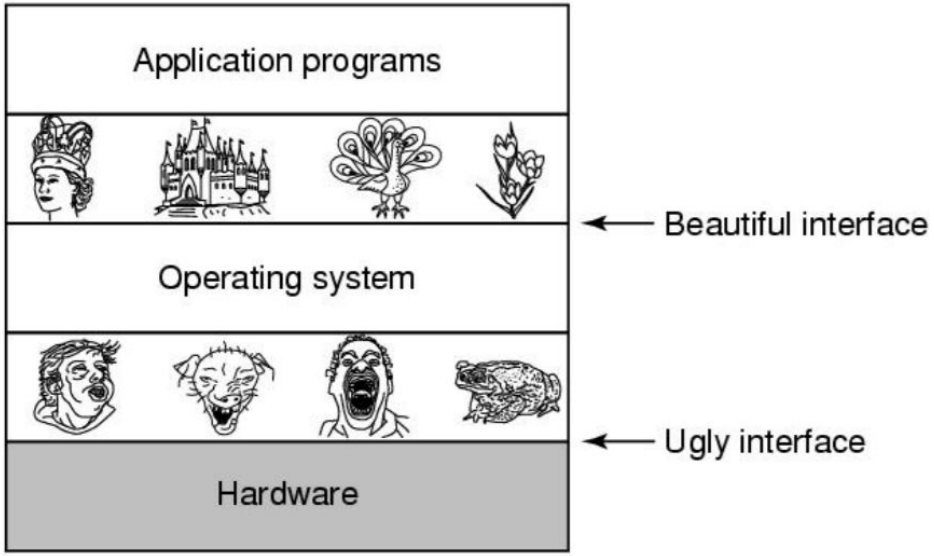


FIgure: Where the operating system fits in.

Historical Background

FIgure: Operating systems turn ugly hardware into beautiful abstractions.



Historical Background

What Is An Operating System?

- The collection of software that directs a computer's operations, controlling and scheduling the execution of other programs, and managing storage, input/output,and communication resources
- It is the software that runs in **kernel mode** - and even that is not always true. Part of the problem is that operating systems perform two essentially unrelated functions:
  - providing application programmers a clean abstract set of resources instead of the messy hardware ones
  - managing these hardware resources.

Historical Background

The Operating System as a Resource Manager

- Allow multiple programs to run at the same time
- Manage and protect memory, I/O devices, and other resources
- Includes multiplexing (sharing) resources in two different ways
  - In time (Sharing the Printer)
  - In space(Allocating Disk space)

Historical Background

History of Operating Systems

Generations:

- (1945–55) Vacuum Tubes
- (1955–65) Transistors and Batch Systems
- (1965–1980) ICs and Multiprogramming
- (1980–Present) Personal Computers

UNIT 1

Historical Background

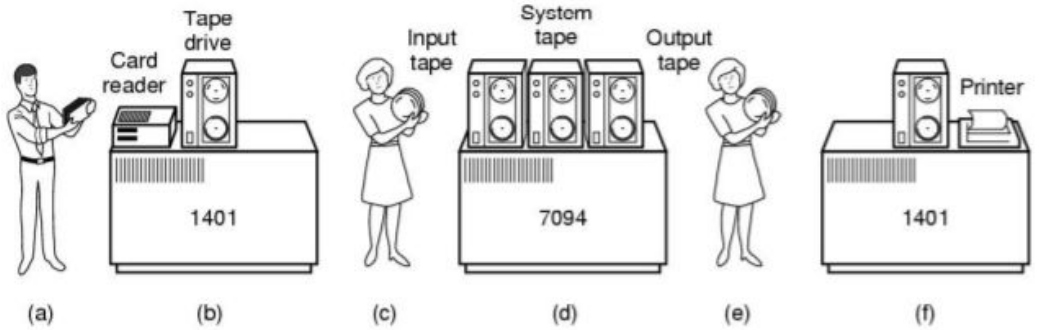


Figure 1-3. An early batch system.
(a) Programmers bring cards to 1401.
(b)1401 reads batch of jobs onto tape.

Historical Background

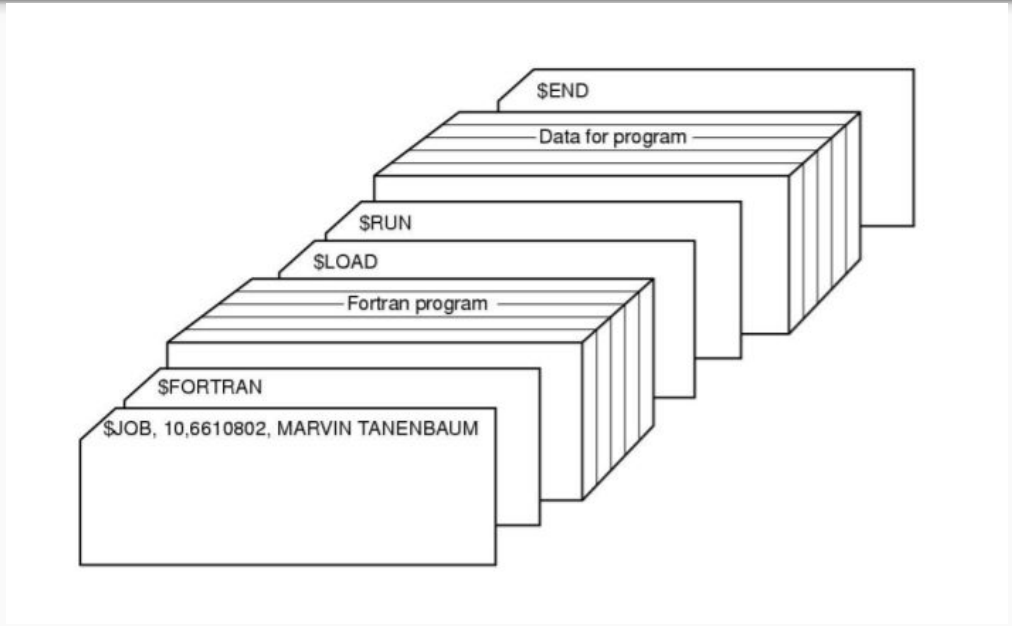


Figure Structure of a typical Fortran Job

Historical Background

ICs and Multiprogramming

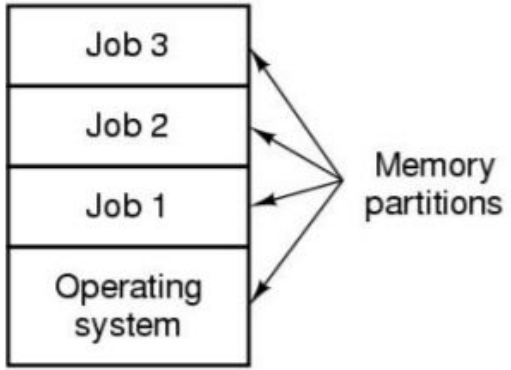
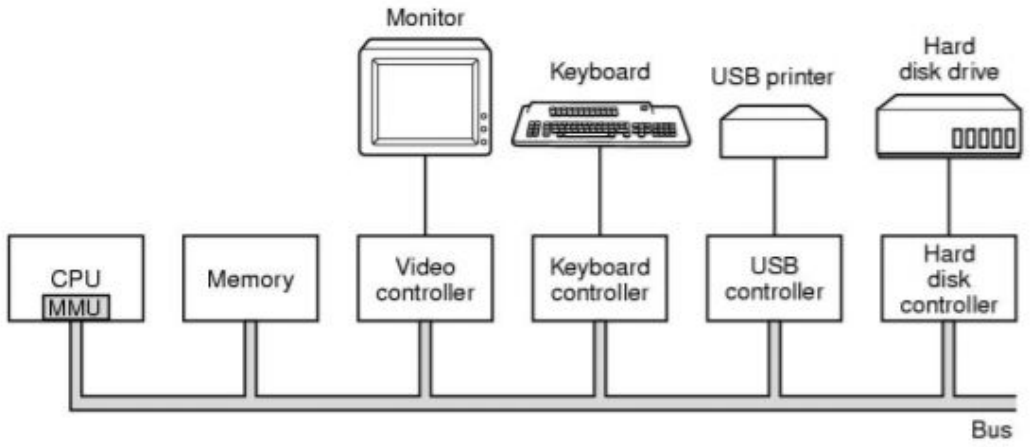


Figure- A multiprogramming system with three jobs in memory.

Historical Background

Computer Hardware Review

Figure- Some of the components of a simple personal computer.



Processors

Processor(CPU) is the **brain of the computer** which f**etches instruction from memory and executes them**. The basic cycle of every CPU is to f**etch the first instruction from memory**, **decode it to determine its type and operands**, **execute it**, and then **fetch, decode, and execute subsequent instructions**. The cycle is repeated until the program finishes. In this way, programs are carried out.

Historical Background

CPU Pipelining

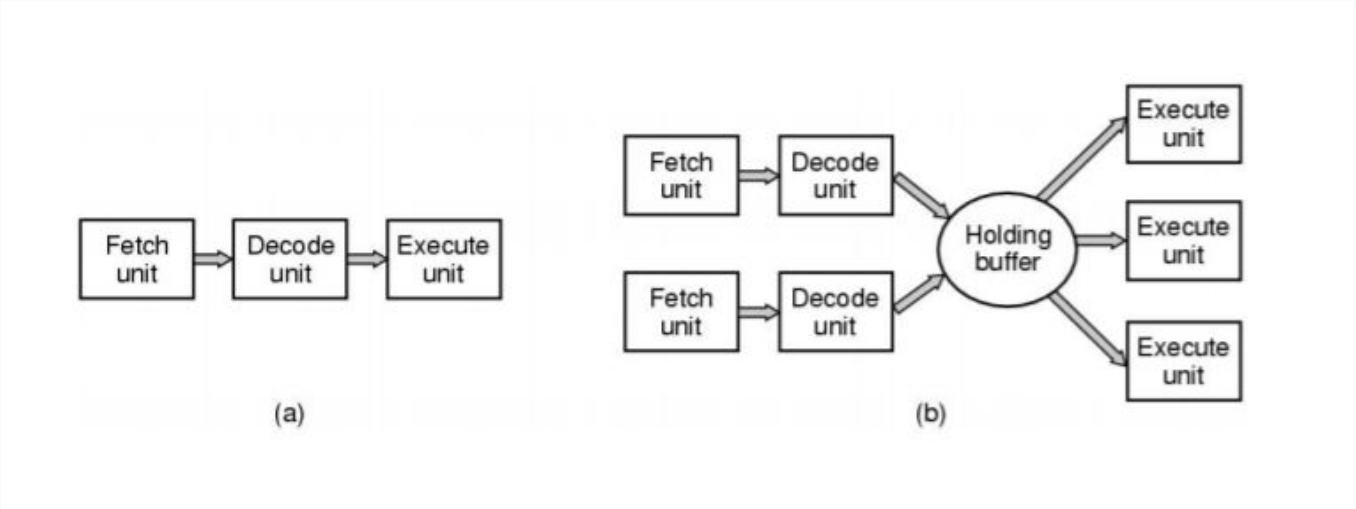


Figure- (a) A Three-stage pipeline. (b) A superscalar CPU

Historical Background

Moore's law states that "**the number of transistors on a chip doubles every 18 months.**"

This "law" is not some kind of law of physics, like conservation of momentum, but is an observation by Intel cofounder **Gordon Moore** of **how fast process engineers at the semiconductor companies are able to shrink their transistors.** Moore's law has held for over three decades now and is expected to hold for at least one more. After that, the number of atoms per transistor will become too small and quantum mechanics will start to play a big role, preventing further shrinkage of transistor sizes.

Historical Background

Intel Pentium 4 introduced this property, called **multithreading or hyperthreading** (Intel's name for it), to the x86 processor, and several other CPU chips also have it—including the SPARC, the Power5, the Intel Xeon, and the Intel Core family

Multithreading allow the CPU to **hold the state of two different threads and then switch back and forth on a nanosecond time scale**.
For example, if one of the processes needs to read a word from memory (which takes many clock cycles), a multithreaded CPU can just switch to another thread. **Multithreading does not offer true parallelism**. Only one process at a time is running, but thread-switching time is reduced to the order of a nanosecond.

Thread is a lightweight process, which, in turn, is a running program

Historical Background

Multithreaded and Multicore Chips

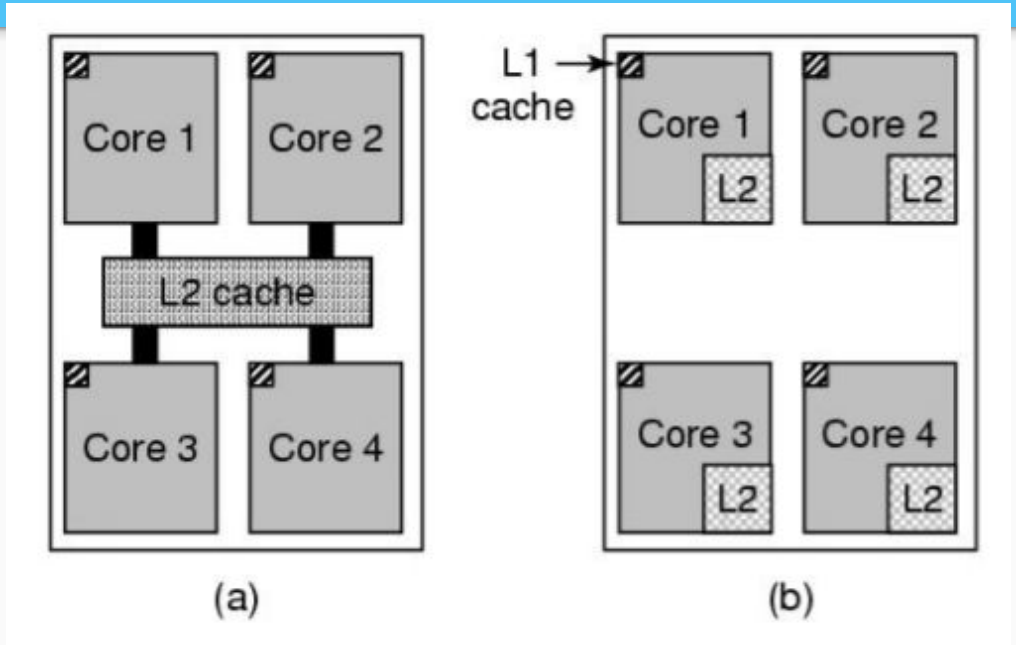


Figure- (a) A quad-core chip with a shared L2 cache. (b) A quad-core chip with separate L2 caches.

Historical Background

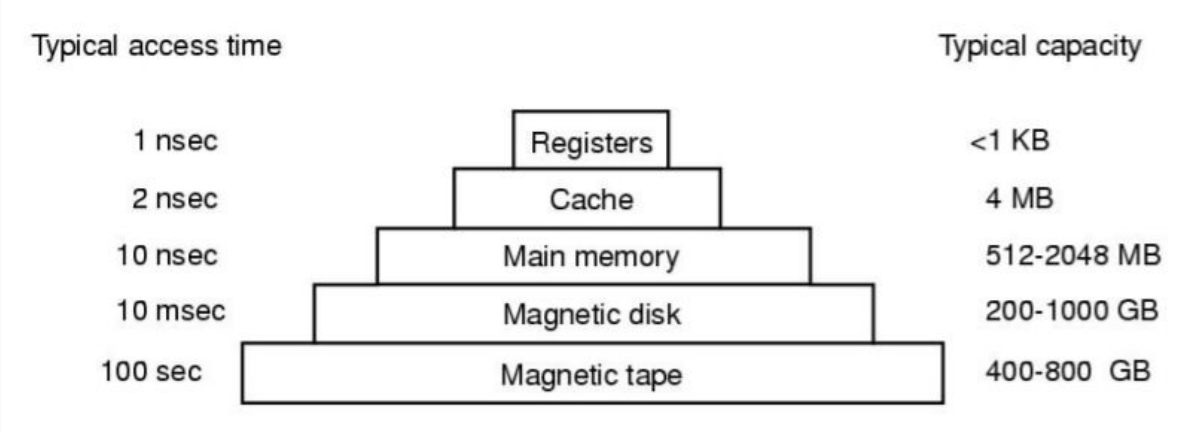


Figure- **A typical memory hierarchy. The numbers are very rough approximations.**

Historical Background

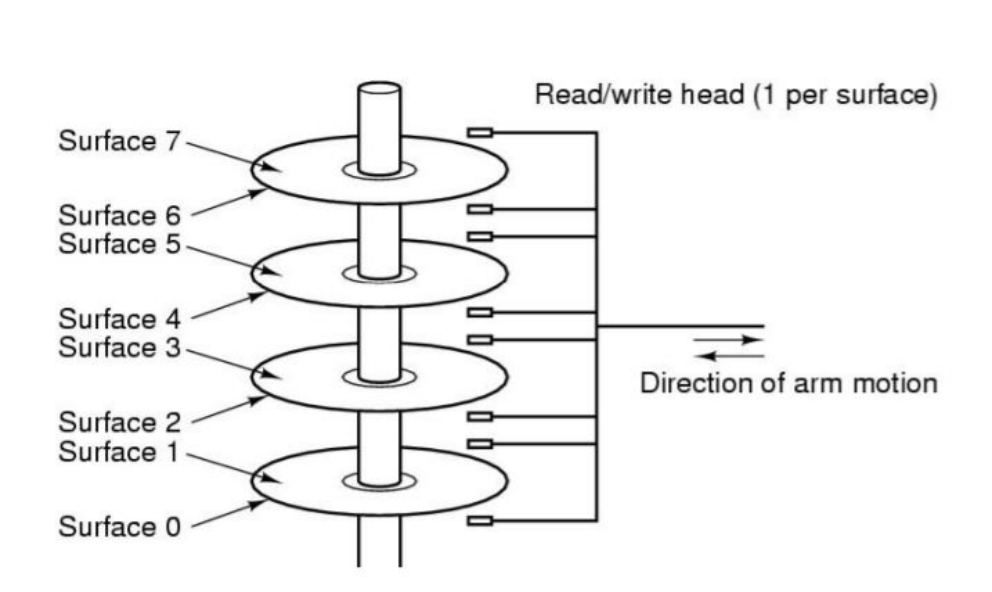


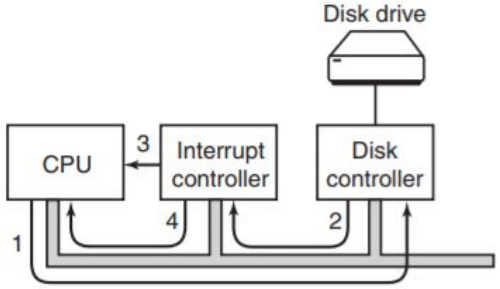
Figure- Structure of a disk drive

I/O Device

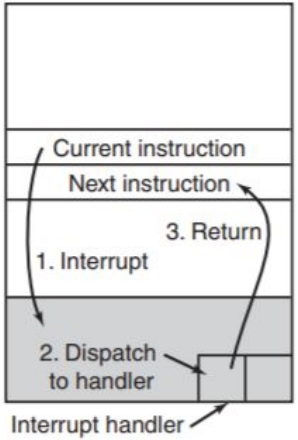
Busy Waiting

A user program issues a system call, which the kernel then translates into a procedure call to the appropriate driver. The driver then starts the I/O and sits in a tight loop continuously polling the device to see if it is done (usually there is some bit that indicates that the device is still busy). When the I/O has completed, the driver puts the data (if any) where they are needed and returns. The operating system then returns control to the caller

Historical Background



(a)



(b)

Figure 1-11. (a) The steps in starting an I/O device and getting an interrupt. (b) Interrupt processing involves taking the interrupt, running the interrupt handler, and returning to the user program.

Historical Background

Interrupt

In step 1, the driver tells the controller what to do by **writing into its device registers**. Then starts the device. When the controller has finished reading or writing the number of bytes it has been told to transfer, it s**ignals the interrupt controller chip using certain bus lines** in step2, if the interrupt controller is ready to accept the interrupt (which it may not be if it is busy handling a higher-priority one), it **asserts a pin on the CPU chip** telling it, in step 3. In step 4, **the interrupt controller** puts the **number of the device on the bus** so the CPU can read it and know which device has just finished (many devices may be running at the same time).

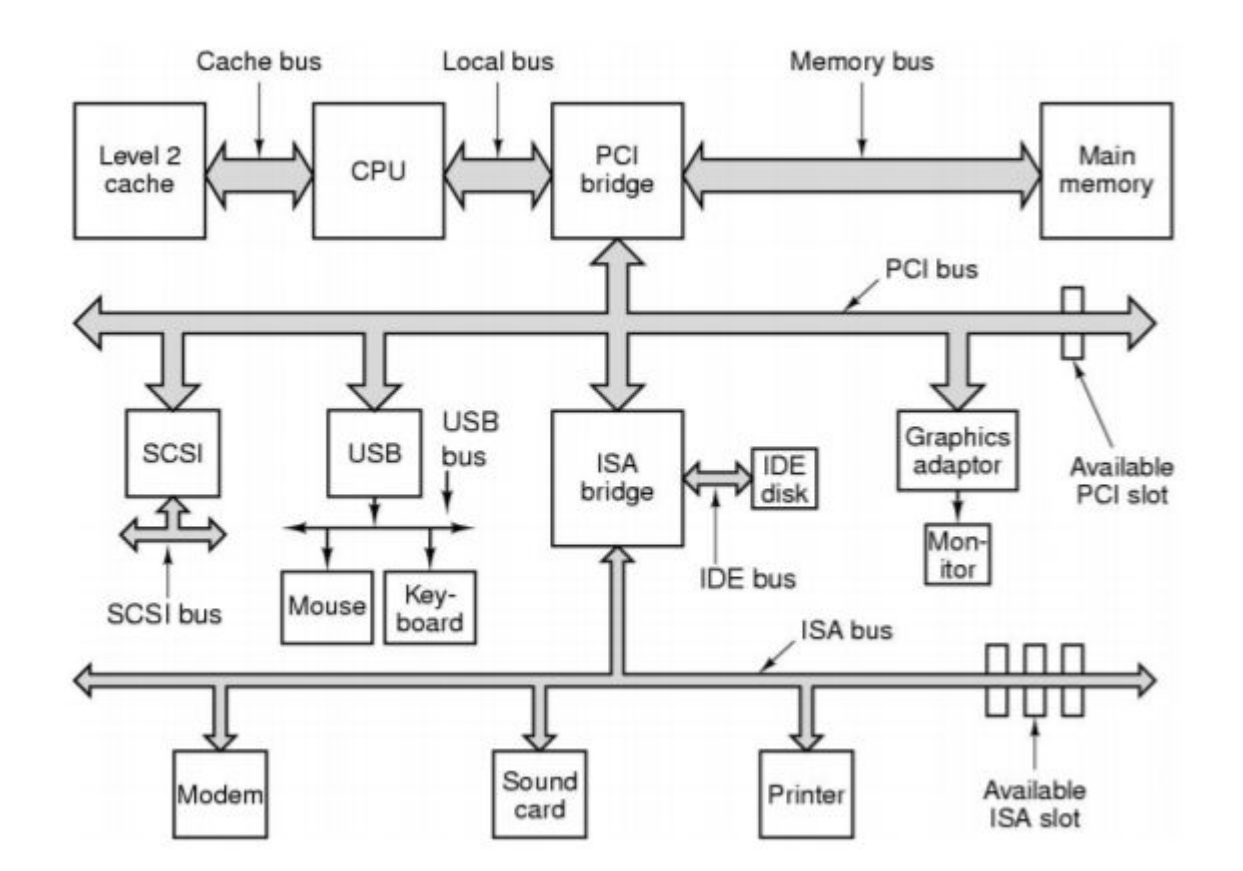


Figure: structure of a large Pentium system

Figure: structure of a large Pentium system

- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Handheld operating systems
- Embedded operating systems
- Sensor node operating systems
- Real-time operating systems
- Smart card operating systems

Historical Background

Operating System Variations

- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Handheld operating systems
- Embedded operating systems
- Sensor node operating systems
- Real-time operating systems
- Smart card operating systems

UNIT 1

Outlines

Historical Background

- Operating system evolution
- Hardware review
- Operating system structure
- Overview of operating system: batch system, multiprogramming, time-sharing, real-time, mainframe operating systems, personal computer operating systems, system calls.

Q&A

For Queries:
sigdelsanjog@gmail.com

Add [NISTBanepa] in Subject for any of your queries

THANK YOU