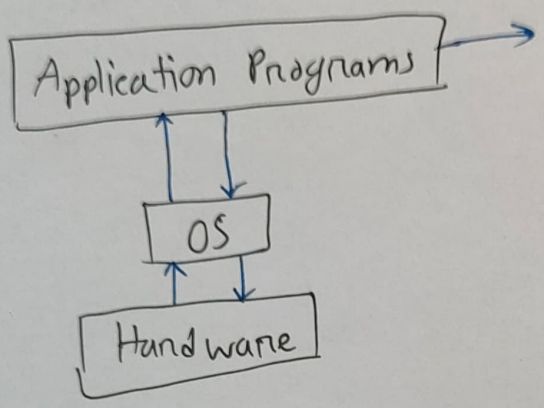
CSE 323 / L-2 / 15.02.2024 /

Chapter - 1

Introduction



(*) Computer system can be divided into four components:

(i) Hardwar ⇒ basic computing resources
- CPU, memory, I/O device

(ii) Operating System
- controls and coordinates use of hardware among various applications and users.

(iii) Application programs
- Define the ways in which the system resources are used to solve the computing problems of user

(iv) Users
⇒ People, machine, other computers.

What is an operating system?

⇒ A program that acts as an intermediary between a user of a computer and the computer hardware.

⇒ main goal,

- use the computer hardware in an efficient manner.

OS known as:

Logical Storage

Hardware + File System

(i) Resource allocator

- allocate memory for program
- allocate ALU for a program
- allocate I/O devices for program

} manage all resource & decides between conflicting request for efficient and fair use.

(ii) Control Program

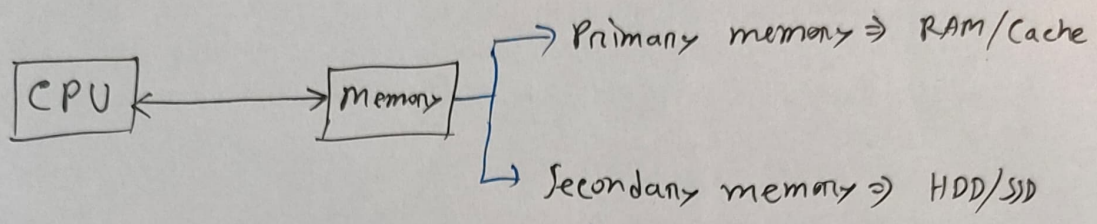
- Prevent Error : check if it is safe to run multiple unit of a single program at a time or there is a dependency.
- Error free efficient run.

kernel:

- Core part of the OS
- Loaded to the RAM just after the bootloader and start execution.
- control everything in the system.
- Application software of OS ⇌ kernel ⇌ Hardware

- critical part of kernel loaded into separate ~~area~~ area, which is protected from access of other's.

- this program runing all the time in the computer.



↳ For access, first data will transfer to Primary then CPU.

HDD/SSD → RAM/Cache ⟷ CPU

Normally OS manges all the address of user program. But, at the startup, OS itself need to be load. Then who carry the address of OS (kernel)?

Before load OS, we need to use some feature (HDD, I/O Device) to load OS.

⇒ These feature are stored in BIOS, integrated in the ~~small size of ROM~~ motherboard.

↳ small size of a ROM (EEPROM) restricted use, can't write

BIOS, need to know the address of kernel and very simple driver of display, Input device.

⇒ This program known as Bootstrap Program.

Fetch / Execute Cycle :

(i) Instruction Fetch (IF)

- From Primary memory (Cache) to Register

(ii) Instruction Decode (ID)

(iii) Data Fetch (DF)

- Data must be fetch to register

(iv) Instruction Execution (EX)

(v) Return Result (RR)

- Return to Accumulator register
- then move to specific address according to instruction.

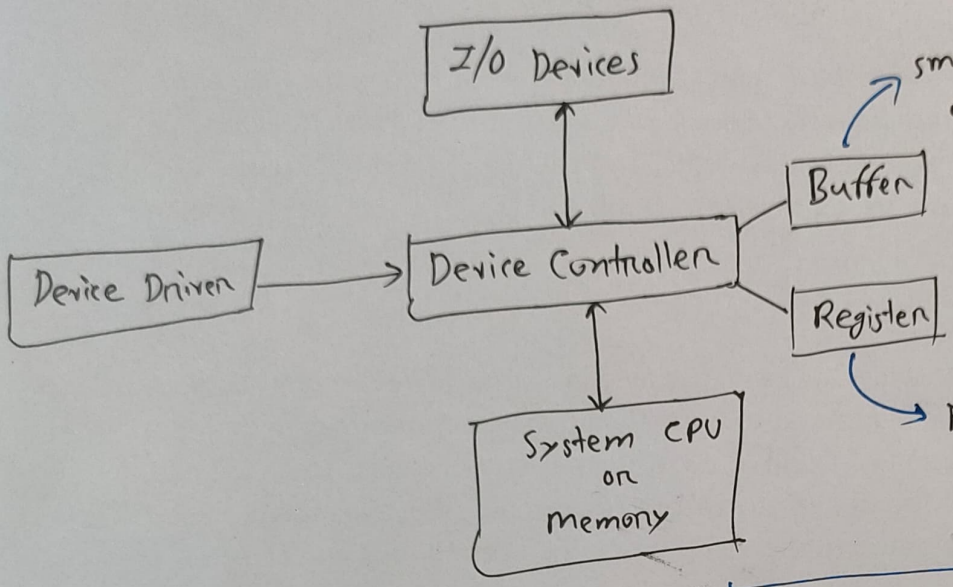
L-3 / 17.02.2024 /

Computer System Structure

⇒ CPUs, device controllers connect through common bus providing access to shared memory.

I/O Operation:

- All kind of external device



⇒ More details step on Slide Page-12

- I/O Devices and the CPU can execute concurrently.
- Each device controllen has a local buffen.
- Each device controllen has an operating system device driven
- data transfen from local buffen
- Device controllen cause an interrupt aften finishing execute.

Slide Page - 12, 13 ***

Interrupt:

- a signal, send by Handwarre
- CPU immediately stops what ~~he~~ it was doing and transfen execution to a fixed location.

starting address of service routine for the interrupt.

- then CPU resumes the interrupted computation.

* Interrupt Handler:
  - interrupt-request line, a wire connected to the hardware.
  - after executing an instruction, CPU check the line, if any signal exist or not.
  - if signal exist,
    read the number from interrupt signal
    jump to the address and start executing
  - Then clear the interrupt and return to the interrupted execution.

* Multilevel interrupts
  - interrupt signal during a critical process can make problem.
  - thats why divided to high and low priority interrupts.

* Interrupt handling features
  - ability to defer interrupt handling during critical processing
  - an efficient way to dispatch to the proper interrupt handler for a device
  - multilevel interrupt, high and low priority interrupts, CPU respond with the appropriate degree of urgency.

Interrupt Request Line:

- nonmaskable ⇒ reserved for events such as unrecoverable memory errors. (High Priority)
- maskable ⇒ used by device controllers to request service. can be wait to execute critical process.

Vectored Interrupt Handler

- every device has interrupt request line, they all are chained in single line ~~and~~
- interrupt vector points ~~the~~ to the head of a list of interrupt handler.
- when an interrupt is raised, Interrupt handler check the list one by one until found the request.
- briefly, reduce the need for a single interrupt handler to search all possible sources of interrupts to determine which one needs service.

Slide Page-19 ***

CSE 323 / L-4 / 22.02.2024

- Interrupt Handler: (Hardware)
  - Interrupt request line, provide the signal
  - Interrupt request immediately stop execution on the CPU
    - If critical process is running then it will check, is it possible to serve later on not.
    - multilevel interrupt. decide the importantness of an interrupt.
- There are two interrupt line,
  - maskable ⇒ can be wait for critical process
  - nonmaskable ⇒ no wait, need to run immediately.
- Interrupt can request by hardware or software.
  - kernel interrupt ~~driven~~
  - hardware → caused by device
  - software → exception, trap, system call, infinite loop.

Slide Page- 20

- System Daemons
  - Services provided outside of the kernel.

Storage Structure

Slide Page- 23-25 ✕ no need

- HDD ⇒ Hard Disk Driven

  NVM ⇒ Non-Volatile-Memory

- Storage system organization depends on,
  - speed
  - cost
  - volotility

Slide Page-26

- There are individual registers set for each ALU (Core)
- CLK speed depend on Registers access time.

  Both need to be sync each other.
- Multi-Programming (Batch System)
  - CPU should not be idle
  - CPU schedulers of OS do this job, kept busy all the time.
    - when a job has to wait, OS switches to another job and keep checking is the job is ready for execute.

Multi-tasking (Time sharing)

- CPU switches jobs so frequently that user can interact with each job while running.

⇒ - Response time < 1 second

- at least one process need to execute in memory
- if multiple job ready to run, schedule them
- virtual memory – if RAM is full.

In RAM, a segment is fixed for OS.

-other segament used for various process.

L-5/26.02.2024/

Interrupt Driven:

- If there is no processes to enecute, no I/O devices to service, no users to respond, OS will sit quietly.

Dual Mode

- User Mode ⇒ when running user program
- kernel Mode ⇒ when running kernel program

⇒ Mode bit

- provided by hardware, defined CPU's running mode

mode bit = 1, user mode

= 0, kernel mode

- Mode switch cased by hardware on software
  - hardware → I/O Devices
  - software → System Call (i.e.: printf())

- No program can access resources directly.
  OS access these and provide to program for maintaining security and error.

- Timer:
  - CPU switches mode very frequently. if CPU doesn't switches mode, then, there is a high chance to be error.
  - if CPU stays in user mode for a long time, then kernel/OS stop the program and take back the control.
  - OS can also be stack in this position, it's big problem.
  - There is a time limit to stay on user mode.
  - Before turning over control to the user, OS ensure that the timer is set to interrupt. After that time either OS give more time or show a fetal error.

- Process management activities (OS)
  - creating or deleting user or system process
  - suspending or resuming
  - process synchronization
  - communication
  - deadlock handling

* Memory management Unit (mmu)

- all or part of the instruction or instruction data must be on memory

⇒ activities

- keep track of currently being used memory.
- Deciding memory movement, in or out
- Allocate or deallocate memory space as needed.

* Storage management (File/Directory System)

↳ Logical storage

- creating and deleting files and directories
- maniplalate files and directories
- mapping files onto secondary storage
- Backup files onto stable (non-volatile) storage media

* Mass-Storage management (Secondary memory management)

- Free-space management
- Storage allocation
- Disk scheduling

Caching

- mediator of CPU and RAM
- multilevel caching
- fasten memory, CPU always checked first.
  - if not then data copied to ~~cha~~ cache and then cpu access it from cache.

Protection & Security

- protection of resource wage
- defence against internal & external attack
- decide which user can do what?
  - read, write protection

L-6 / 28.02.2024

Computer System Architecture

Multiprocessors

- known as parallel system, tightly-coupled system

⇒ Advantage

- Increased throughput ⇒ How much process can run in a certain time
- Economy of scale ⇒ Duel core can give us output of 2 computer, but price increase only 20% about.
- Increased reliability ⇒ if a core fail, other core still work

⇒ two types

- Asymmetric M.P. ⇒ at least one processor are dedicated for special task.
- Symmetric M.P ⇒ can do all type of task

Multi-Core / Multi chip

- contain multiple CPU/ALU. Each ALU have own register set.
- L1 cache are dedicated to each ALU.
- OS manage/schedule the instruction and select in which core it will be executed

How OS involve in multicore architecture?

NUMA

- Non-Uniform Memory Access System
- CPU's are connected by a shared system interconnect, so that all CPUs share one physical address space.
- every CPU have its own local memory. CPU can access it very fast. Also CPU are interconnected
- But if a CPU needs to access another CPU's local memory, then it will be slow process.

Slide Page-44

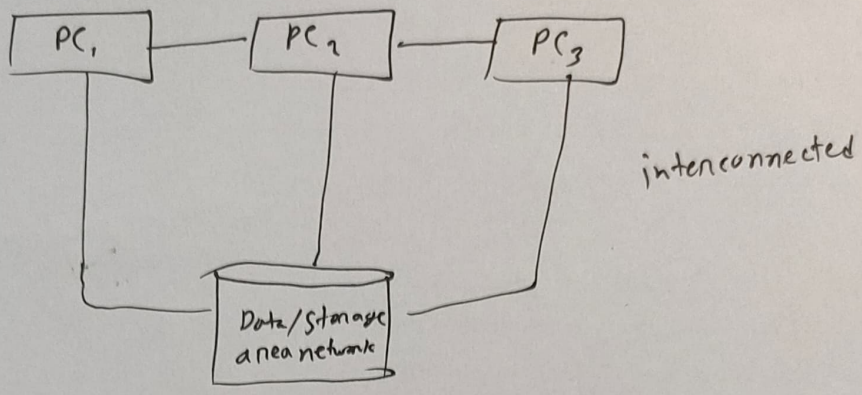
Clustered System:

- multiple system working together
- sharing storage via SAN (storage area network)
- provides high availability service

⇒ Two types:

- Asymmetric ⇒ has one machine is hot-standby mode
- Symmetric ⇒ multiple nodes running applications and monitor each other.

⇒ Like multiple computer using one data server.



Computer System Environments

⇒ Traditional

- stand-alone general purpose machine
  - security is not much important
- portals ⇒ provide web access to internal system
- Network computers (thin clients) ⇒ web terminals
- Wireless network
- firewall ⇒ for protect home computer from internet attack

⇒ Mobile Device

Advantage than a traditional Laptop

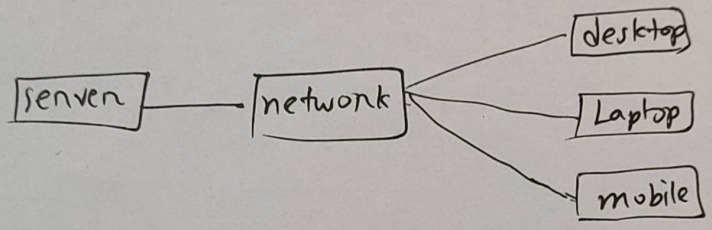
- more OS features (GPS, gyroscope)
- augmented reality apps
- IEEE 802.11 wireless / ~~cellular~~ cellular data network
- Most famous Apple IOS and Google Android

⇒ Client server

- responding ~~req~~ request generated by clients
- compute-server system
  - provides an ~~int~~ interface to client to request service (database)

~~- File system~~

- File server system
  - provides interface for client to store and retrieve files



⇒ P2P ⇒ Peer to Peer

- Torrent system, VoIP
- each act as client, server or both
- connected in P2P network

⇒ Cloud computing

- Delivers computing, storage, even apps as a service across a network
- virtualization

⇒ Types

- Public Cloud ⇒ available to anyone willing to pay
- Private Cloud ⇒ for company own use
- Hybrid Cloud ⇒ both public & private

⇒ Saas ⇒ Software as a Services ⇒ online software (word)

⇒ Paas ⇒ Platform as a service ⇒ database server, ready for application use via internet

⇒ Iaas ⇒ Infrastructure as a service ⇒ Storage available over internet (GDrive)

⇒ Virtualization

- allow OS to run as application with in other OS
- Emulation
  - when source CPU type different from target type
  - Rosetta - run ZBM CPU on Intel CPU
  - emu8086
- Virtualization
  - OS need to be similar
  - VMware
  - VMM, provides virtualization services

Slide Page - 56

  - based on software

⇒ Real time embedded system

- Washing machine (Real time OS)
- an OS embedded to machine, controll machine
- special purpose processor used

L-7/02.03.2024

Kernel Data Structure

Divided into two spaces

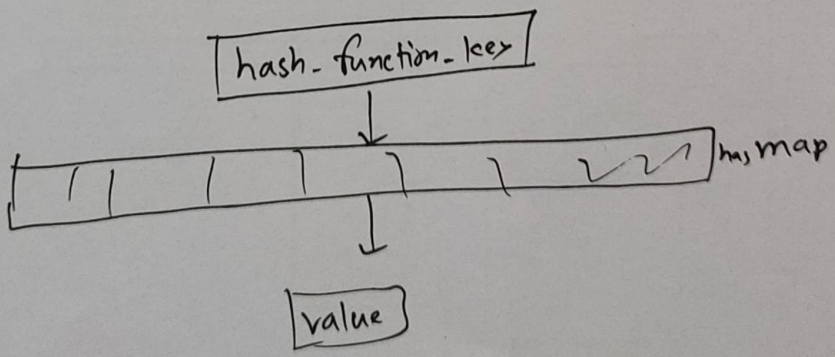
- User space ⇒ Code & Data
- Kernel Space ⇒ Process info, Open File Table, Process Table.

Some data structure (keep info. of process)

- Singly linked list
- Doubly linked list
- Circular linked list
- Binary tree (keep track of child process)

⇒ Hash Function

- can be used to map data of arbitrary size to data of a fixed size.
- can create a Hash Map



- used for security, to store passwords

Bitmap

- is a string of n binary digits that can be used to represent the status of n items.

- 0 for available
- 1 for unavailable

⇒ $\overrightarrow{001011101}$

⇒ Resource 2, 4, 5, 6, 8 unavailable

others are available

Quiz-1

09.03.2024

Lecture Slide - 1

Conceptual Question

Answere directly no need to overload description

⇒ Why & How Related Question

2 to 3 Question

30-40 minute time