Chapter 1:

- Turing Model:

The idea of a universal Computational device was first described by Alan Turing in 1937. He propsed all computation could be performed on a special machine, now called a Turing machine.

   - Data Processors:

     Input Data ◊ Computer ◊ Output Data o

   - Programmable data processors:

     Turing model is a better model for a general-purpose computer. This model adds the program element – Set of instructions telling computer what to do

     Input Data ◊ Computer ◊ Output Data Program o

   - The Universal Turing machine: A machine that can do any computation if the appropriate program is provided. Turing machine is capable of computing anything that is computable
- Von Neumann model:
   - Four Subsystems
      - Memory: Storage area where programs and data are stored during processing.
      - Arithmetic Logic unit (ALU): where calculation and logical operations occur.
      - Control Unit: Controls the operations of memory, ALU & input/output system
      - Input / Output: input subsystem accepts input data and the program from outside the computer, while the output system sends results of processing to the outside world. Includes secondary storage devices such as disk or tape too.
   - Stored program concept:

     Von Neumann model states that the program must be stored in memory. Totally different from the architecture of early computers in which only the data was stored in memory: Programs for their tasks were implemented by manipulationg sets of switches or changing wiring.

   - Sequential execution of instructions:

     A program in Von Neumann model is made of a finite number of instructions. The Control unit fetches 1 instruction from memory, decodes it, then executes it. Instructions are executed sequentially.

- Computer components:
   - Computer Hardware
   - Data:
      - Storing data & Organising data
   - Computer Software:
      - Programs must be stored in the memory

        - Sequence of instructions: Program must consist of a sequence of instructions, each operating on one or more data items.
    - Algorithms : A step-by-step solution to a problem is called
    - Languages : FORTRAN and COBOL
    - Software engineering: are principle that the program must follow strictly
    - Operating systems
- History: (There are some question about which have been noted below)
    - Mechanical Machines: Pre-1930
        - 17th century: Blaise Pascal invented Pascaline, a mechanical calculator for addition and subtraction operations. 20th Century, Niklaus Wirth invented a programming language and named it Pascal
    - Electronic Computers: 1930 – 1950
    - Computer Generations: 1950 – Now
    - Social and ethical issues:
        - Social:
            - Dependency
            - Social Justice: Computers created a service only for middle class or high income people. Low income are deprived.
            - Digital Divide: Covers issues of dependency and social justice.
        - Ethical:
            - Privacy: Hackers, etc
            - Copyright: Who owns data?
            - Computer crime: Viruses, Hackers, etc.

Chapter 2:

- **A number system (or numeral system)** is a system that uses distinct symbols to represent a number.
- In a **positional number system**, the position a symbol occupies in the number determines the value it represents. Each position has a place value associated with it.
  - In the decimal system, the base b = 10 and we use ten symbols to represent numbers. The symbols in this system are often referred to as decimal digits or just digits.
  - In the binary system, the base b = 2 and we use only two symbols to represent numbers. The symbols in this system are often referred to as binary digits or bits.
  - In a hexadecimal system, the base = 16 and we use 16 symbols to represent numbers. The symbols in this system are often referred to as hexadecimal digits.
  - In an octal system, the base = 8 and we use eight symbols to represent numbers. The symbols in this system are often referred to as octal digits.
- A **nonpositional number system** uses a limited number of symbols in which each symbol has a value. However, the position a symbol occupies in the number normally bears no relation to its value: the value of each symbol is normally fixed.
  - Ex: Roman number system: I V X L C D M

Chapter 3:

- Unsigned representation:
  - Can never be negative and can only take 0 or positive values
    - 1. Integer is changed to binary
    - 2. If number of bits is less than n, 0's are added to the of the binary integer so there is a total of n bits. If more than n, integer cannot be stored and **overflow**[1] occurs.
- Sign and magnitude:
  - Not commonly used to store integers, more to store part of a real number in a computer.
  - Two zeros, positive and negative.
- Floating Point Representation:
  - Made up of 3 parts, a sign, a shifter and a fixed point number
    - Sign: Positive or negative
    - Shifter: How many points for the decimal
    - Fixed Point Number: position of the decimal is fixed
- Normalization: To make the fixed part of the representation uniform.
- Sign Exponent & Mantissa:

  o Sign of the number stored using 1 bit

  o Exponent defines shifting of the decimal point

  o Mantissa is the binary integer to the right of the decimal point, treated like an integer together with the sign and stored in sign and magnitude representation

- Storing Audio:
  - Audio is representation of sound or music.
  - Audio is analog whereas text is digital.
  - Sampling:
    - Select only a finite number of points on the analog signal, measure their values, and record them
  - Quantization:
    - Value measured for each sample is a real number. This means we can store 40000 real values for each one second sample
    - Refers to a process that round the value of a sample to the closest integer value.;
  - Encoding:
    - The quantized sample values need to be encoded as bit patters.
    - Some systems assign positive and negative values to samples, some just shift the curve to the positive part and assign only positive values.

---

1 Overflow occurs when there are insufficient bits in a binary number representation to portray the result of an arithmetic operation. Overflow occurs because computer arithmetic is not closed with respect to addition, subtraction, multiplication, or division. Overflow cannot occur in addition (subtraction), if the operands have different (resp. identical) signs.
To detect and compensate for overflow, one needs n+1 bits if an n-bit number representation is employed.

        - Bit per sample:
            - System needs to decide how many bits are allocated for each sample.
        - Bit rate:
            - The number of samples per second.
        - Standarads for audio encoding
- Storing images:
    - Raster Graphics are used when we need to store an analog image like a photograph. Which consists of analog data, like audio information. The data must be sampled, in this case scanning. Samples are pixels.
    - Resolution:
        - The amount of pixels needed to record for each square or linear inch.
        - The scanning rate in image processing is called resolution.
    - Color depth:
        - The number of buts used to represent a pixel
        - True color – 24 bits.
        - 3 x 8 bitdecimal numbers (RGB & 255)
        - Indexed Colour only uses a portion of True colour's colour, then indexes them.
    - Vectors graphics:
        - Does not store the bit pattern for each pixel. Image is decomposed into a combination of geometrical shapes, such as lines, squares, or circles.
- Storing Video:
    - A video is a representation of image frames over time to create the illusion of motion.

<u>Caution:</u> How is a bit pattern length related to the number of symbols the bit pattern can represent?

Each bit can represent two symbols - often depicted as 0 or 1, or OFF and ON, or FALSE and TRUE. (Similarily the system of arabic numerals has a DIGIT than represent TEN symbols - 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.)

In the system of arabic numerals, a length of 1 symbol only allows you to represent 0 to 9 or 10 states (or 10 to the first power states.) A length of 2 symbols (digits) allows you to represent 10 to the 2nd power states - 0 to 99, or 100 states.

Similarily with bits. One bit allows you to represent 2 states. Two bits allow you to represent 2 to the 2nd power states (or 4.) 8 bits allows you to represent 256 states (2 to the 8th power.)

Chapter 4:

Chapter 5:

- CPU:
  - ALU: Performs logic, shift and arithmetic operations on data.
    - ♣ Logic: NOT, AND, OR and XOR.
    - ♣ Logical shift patters shift bit patterns to the left or right, while arithmetic operations are applied to integers.
  - Registers: Fast stand alone storage locations holding data temporarily.
    - ♣ Data registers: Today, computers have dozens of registers inside the CPU.
    - ♣ Instruction registers: Fetching instructions one by one from memory, storing them in the Instruction register,
    - ♣ Program Counter: Keeps track of the instruction currently being executed. & Increments to point to the address of the next instruction.
  - Control unit: Controls the operation of each subsystem.
    - ♣ Memory: Second major subsystem in a computer
    - ♣ Memory addresses are defined using unsigned binary integers.
    - ♣ Data is transferred to and from memory in groups of bits called words.
    - ♣ Types:
      - RAM: A data item can be accessed randomly – using the address of the memory location without the need to access all data items before it volatile
      - SRAM: Using traditional flip-flop gates to hold data. Fast but expensive.
      - DRAM: Uses capacitors, for data storages. If Charged, state is 1. Celss need to be refreshed periodically. Slow but inexpensive
      - ROM: CPU can read from but not write to. NON volatile, used for programs or data that must not be erased or changed
      - PROM: Blank when computer is shipped, User with equipment can store programs on it. When stored, behaves liekROM and cannot be overwritten.
      - EPROM: PROM that can be erased with a special device applying ultraviolet light. Has to be removed.
      - EEPROM: EPROM that can be erased with electronic impulses without being removed.
- Subsystem interconnection:
  - Connecting CPU and memory
    - ♣ Data Bus: made of several connections, each carrying 1 bit at a time. Number of connections depends on size of word used by the computer.
    - ♣ Address Bus: Allows access to a particular word in memory. The number of connections in the address bus depends on the address space of the memory.
    - ♣ Control Bus: Carries communication between CPU and memory. The number of connections used in the control bus depends on total number of control commands a computer needs.
  - Connection I/O Devices:
    - ♣ Cannot be directly connected to buses that CPU and memory use.

        - Controllers:
            - Controllers or interfaces bridge gap between nature of IO device and cpu and memory. Serial Controller only has one data wire, while parallel controller has several data connections so several bits can be transferred at a a time.
    - Program execution:
        - Machine Cycle:
            - Consists of three phases, fetch, decode and execute.
                - Fetch: Control unit orders the system to copy next instruction into the instruction register in the CPU. Address of the instruction is to be held in the program counter registry
                - Decode: When instruction is in the instruction register, is it decoded by the control unit. Result of this is the binary code for someoperation that the system will perform.
                - Execute: After the instruction is decoded, control unit sends the task to a component in the cpu.
        - I O Operation:
            - Programmed IO:
                - Synchronization is primitive, CPU waits for the IO device.
                - Transfer of data between IO device and CPU is done by an instruction in the program.CPU time is wasted as it constantly checks the status of the IO device.
            - Interrupt Drive IO
                - CPU informs the IO Deivce that a transfer will happen, but does test the status of the IO deivce continuously.
                - IO Deivce interrupts the CPU when it is ready.
                - CPU time is not wasted.
            - Direct Memory Access:
                - Transfers a large block between a high speed IO device with passing it through the CPU.
                - Requires a DMA controller that relieves CPU of some of its functions.
                - Has registers to hold a block of data before and after memory transfer.
        - Different Architectures:
            - CISC:
                - Complex instruction set computer
                - Large set of instructions, including complex ones.
                - Single instruction for both simple and complex tasks
                - Objection is the overhead associated with microprogramming and access to micromemory.
            - RISC
                - Reduced instruction set computer.

        - Small set of instructions that do minimum number of single operations
        - Complex operations are simulated using a subset of simple instructions
        - More difficult and time consuming.
    - Pipelining:
        - Improves throughput of the three phases of instructions.
        - Control unit can do two or three of these phases simultaneously. Next instruction can start before previous one completes.
    - Parallel Processing:
        - Have multiple control units, multiple ALU's, multiple memory units.
- ♣ SISD organisation:
    - A single instruction stream, single data stream
    - Represents computer with one control unit, ALU and memory unit
    - Instructions executed sequentially.
- ♣ SIMD Organisation:
    - Single instruction stream, multiple data stream
    - Represents a computer with one control unit, multiple processing units and one memory unit.
    - All processor units receive the same instruction from the control unit

Chapter 6:

- Networks – Interconnection of a set of devices capable of communication.
  - LAN:
    - Usually privately owned
    - Each host has an indentifier.
  - WAN:
    - Wider geographical span.
  - Internetwork:
    - Point to point dedicated WAN
  - Internet:
    - Two or more networks that can communicate with each other.
  - Hardware and software:
    - Both hardware and sofrware is needed.
  - Protocol Layering:
    - Defines the rules that both sender and received and all intermediate devices need to follow.
    - Simple – One simple protocaol
    - Complex – Protocol at each layer – protocol layering!
  - TCP / IP
    - Set of protocols organised in different layers.
  - 5 Layers
    - 5. Application Layer:
      - Provides services to the user. Communication is provided with a logical connection
      - Traditional Paradigm – Client server paradigm.
      - New paradigm – Peer to peer.
        - Centralised – Directory system of the peers and what is offered uses the client server paradigm. Storing and downloading of files are done unsing the P2P diagram. Servers are vulnerable to attack.
        - Decentralised – Does not depend on a centralised directory system. DHT is commonly use (Distributed Hash Table)
    - 4. Transport Layer:
      - Provides services to the application layer and receives services from the network layer.
      - Acts as a liaison between a client program and a server program.
      - Uses UDP and TCP
      - Addressing: Port Numbers
    - 3. Network Layer:
      - Responsible for packetizing: Encapsulating payload at source and decapsulating at destination.
      - May be encapsulated in several network layer packets.
      - Responsible for host to host delivery of messages.

                - Packet Delivery: Unreliable and connectionless.
                    - Packets can be easily lost or corrupted
                    - Treats each packet independently
                - Routing: Routing the packet from the source to destination
                - IPV4 – 32 bit address
                - IPV6 – 128 Bit Address
            - 2. Data Link
                - Territories of networks that when connected make up the internet.
                - Receive services and provide services to the network layer.
                - Host to host.
                - Nodes and links – Data from one point needs to pass through many networks to reach another point. End hosts are nodes and networks in between are links.
            - 1. Physical
                - Communication is node to node
                - Role is to transfer the bits received from the data link later and convert them to electromagnetic signals for transmission.
                - Data Signals
                    - Analog and digital
                    - Analog – Data that is continuous
                    - Digital – take on discrete values
        - Transmission media:
            - Guided media: Twisted pair cable, coaxial cable, and fibre optic cable.
                - Coaxial – Central core conductor and covered insulation and an out conductor.
                - Fibre – Made of glass or plastic and transmits signals in the form of light. This technology uses the property of a beam of light that is either refracted when encounter a medium of less dense. Covering a glass or plastic medium by another less dense medium. Covering a glass or plastic medium by another less dense medium, guiding the light through the medium
            - Unguided media
                - Radio Waves – Electromagnetic waves ranging in frequencies between 3kHz and 1GHz are normally called radio waves.
                - Microwaves – Have frequencies between 1 and 300 Ghz. Unidirectional
                - Infrared waves – with frequencies from 300GHz to 400 THz. Can be used for short range communication.

Chapter 7:

- Operating System
- Interface between hardware of computer and user and facilitates the execution of other programs and access to hardware and software resources o Efficient use of hardware
- Ease of use of resources
- Bootstrap process:
- Small section of memory is made of ROM and holds a small program called the bootstrap program.
- Batch systems:
- Designed in 1950s to control mainframe computers. Each program to be executed was called a job.
- Very simple. Only ensured that all the computer resources were transferred from one job to the next
- Time sharing systems:
- Multiprogramming was introduced
- Resources could be shared between different jobs, with each job allocated a portion of time to use a resource.
- Improved the efficiency o Required more Complex OS's, had to do scheduling – allocating resources to different programs and deciding which program should use which resource, and when.
- A job is a program to be run, a process is a program in memory waiting for resources.
- Personal Systems:
- Single user operating systems such as DOS were introduced.
- Parallel Systems
- Multiple CPUs on one machine. Each CPU can be used to serve one program or part of a program, meaning many tasks can be accomplished in parallel instead of serially.
- OS needs to be more complex.
- Distributed Systems:
- Job done on one computer can now be shared between computers far apart.
- Program can be run partially on one computer and partially on another if they're connected through an internetwork
- Combine features of the previous generation with new duties such as controlling security.
- Real time systems:
- Expected to do a task within a specific time constraint.
- Used with real time applications o Traffic control, patient monitoring or military control systems are examples.
- Application program sometimes be an embedded system such as a component of a large system.
- Components:
- User Interface:
    - ♣ Accepts requests from users and interprets them for the rest of the operating system. Sometimes called a shell, or a window.
- Memory Interface:
    - ♣ Memory management is a responsibility of a modern computer system.

- ♣ OS can be divided into categories of memory management.
- ♣ Monoprogramming:

- Memory is allocated to a single process.
- Only a small part needed to hold the OS.
- When memory size is less than size of program, the program cannot be run.
- When one program is run, no other program can be executed.
- Inefficient use of Memory and CPU Time

- ♣ Multiprogramming:

- Enables more than one program in memory at a time.
- Executed concurrently
- CPU rapidly switches between programs.
- Nonswapping: Partitioning and paging
- Partitioning: Memory divided into variable length sections, each holding a program. Priority levels can be set on the CPU for the most CPU time allocated. Improves efficiency, but size pf the partitions needs to be determined beforehand by memory manager. Holes can occur (Unused locations.).
- Paging: Improves the efficiency of partitioning. Memory is divided into equal sections called frames. Programs are divided into equally sized sections called pages. If program has 3 pages, it occupies 3 frames in memory.
- Swapping: Demand paging & demand segmentation
- Demand paging: Pages can be loaded into memory one by one, executed, and replaced by another page, instead of the whole program loaded into memory at once, like Paging. Consecutive pages from the same program do not need to be loaded into the same frame.
- Demand Segmentation: Program is divided into segments matching the programmer's view. Loaded into memory, executed, and replaced by another module from the same or different program.
- Can be combine to further improve the efficiency of the system. Memory is divided into frames, and module is divided into pages should the segment be too large to fit into available free space in memory.
- Virtual Memory:
- Storage on disk to store a portion of the memory used by a program, example, 30MB program, 10MB in Memory, 20MB on disk.

- ♣ Process Manager:

- Program Job and process refer to the instructions.
- Program: Nonactive set of instructions stored on disk. May or may not become a job
- Job: Program becomes a job when selected for execution. Until it has finished running and becomes a program again. Every job is a program, but not every program is a job
- Process: A program in execution. A program that has started but not finished. A process may be executing or waiting for CPU time. As long as the job is in memory it is a process. Every Process is a job, but not every job is a process.

- ♣ Schedulers:

- Job scheduler: Moves a job from the hold state to the readym state or running state to terminated state.

- Process Scheduler: Moves a process from the running state to the waiting state when the process is waiting for some even to happen. Moves from the waiting state to the ready state when the event has occurred. Moves from the running state to the ready state if the process' time allotment has expired.
    - ♣ Queuing:
- Handles multiple processes and jobs waiting to use the CPU resources.
- Has many queues, job queue holds the jobs waiting for memory, Ready Queue holds the processes in memory, ready to be run waiting for the CPU. The I O Queue holds processes waiting for an IO Device.
    - ♣ Process synchronisation:
- Synchronise different processes with different resources. Whenever resources can be used by more than one user or process, a deadlock or starvation can occur…
- Deadlock –When a process A is using File 1, and requests File 2, while Process B is using File 2 and requests File 1. Occurs when the OS foes not put resource restrictions on processes.
- Mutual exclusion: Only one process can hold a resource
- Resource Holding: A process holds a resource even though it cannot use it until other resources are available
- No pre-emption: OS cannot temporarily reallocate a resource
- Circular waiting: All processes and Resources involved for a loop.
- All four conditions need to occur to form deadlock.
- Starvation: Opposite of deadlock, Can happen when the OS puts too many resource restrictions on a process.
    - ♣ Device Manager:
- Monitors ever IO device constantly to ensure that the is functioning properly. Also needs to know when a device has finished serving a process and ready to serve the next process in queue.
- Device manager also maintains a queue for each IO device or one or more queues for similar IO Devices.
- Device Manager controls the different policies for accessing IO devices.
    - ♣ File manager:
- Used to control access to files. Access is only permitted by applications and or users. And the type of access varies.
- File manager supervises the creation, deletion and modification of files,
- Controls naming of files
- Supervises storage of files, how and where they're stored.
- Responsible for archiving and backups
- OS:
    - ♣ UNIX
- Multiuser, multiprocessing, portable OS.
- Designed to Facilitate programming, text processing and communication.
- Consists of 4 major components, kernel, shell, standard set of utilities, and application programs.
- Kernel – The heart of UNIX. Contains the most basic parts of the OS, Memory, process, device and file management.
- Shell - most visible to the user. Receives and interprets commands entered by the user. Most important component. Has several different Shells

- Utilities – A standard UNIX program that provides a support process for users. Such as text editors, search programs, and sort program.
- Applications – Programs not forming a standard part of the OS Distro.
    - ♣ Linux:
- Kernel – Kernel is responsible for all duties attributed to the kernel.
- System Libraries – hold a set of functions used by application programs including the shell, to interact with the kernel.
- System utilities – Individual programs using the services provided by the system libraries to perform management tasks.
- Supports the standard internet protocols, 3 Layers, socket interface, protocol drivers and network device drivers
    - ♣ Windows:
- Design Goals: extensibility, portability, reliability, compatibility and performance.
- Extensibility: Designed as a modular architecture with several layers. Higher layers can be changed with time, without affecting lower layers
- Portability: Mostly written in C or C++ and code is independent of the machine language of the computer
- Reliability: Designed to handle error conditions including protection from malicious software.
- Compatibility: Windows was designed to run programs written for other operating systems and earlier versions of windows.
- Performance: Designed to have a fast response time to applications running on top os the OS.
- Layered architecture:
- HAL – Hardware abstraction layer: Hides hardware difference from upper layers;
- Kernel – Heart of the operating system, is an object oriented piece of software that sees any entity as an object.
- Executive: Provides services for the whole OS. Made up of six subsystems, object manager, security reference monitor, process manager, virtual memory manager, local procedure call facility and the IO manager.
- Environmental subsystems: Allows Windows to run application programs designed for windows, other OS's or earlier versions of windows. Native subsystem designed for windows is called Win32.