Tài Liệu CSI

CHƯƠNG 1: INSTRODUCTION (cậu lệnh)

Alu : subsystem performs calculations and logical operations.
Control unit : subsystem serves as a manager of the other subsystem
Memory : subsystem stores data and programs
An Algorithm : a step by step solution to a problem
Input/Output :ubsystem accepts data and programs and sends processing results to output devices
A program: in a set of instruction in a computer languages that tell the computer what to do with data.
Software engineeing : is the design and writing of a program in structered form
High level programing languages: separated the programming task from computer operation tasks.
Von Neumann : model is the basis for today's computers.
The jacquard loom : The first computing machine to use the idea of storage and programming
ABC. :The firth electronic special-purpose computer
Computer languages :FORTRAN and COBOL
Pascaline :A 1fith -century computing machine that could perform addition
EDVAC :One of the first computers based on the von Neumann model
Data and programs :According to the von Neumann model, ________ are stored in memory.
Software engineering : is the design and writing of a program in structured form.

CHUƠNG 2 : NUMBER SYSTEM

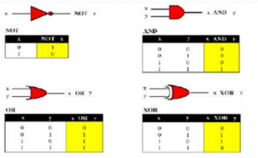
Octal number	: Cơ số 8	subtract from: trừ	divide: chia
Hexadecimal number	: Cơ số 16	add to : cộng	multiply: nhân

CHƯƠNG 3:DATA STORARE (Lưu trữ dữ liệu)

Data type: numbers, text, audio, image and video.
Storing Integers(lưu trữ số nguyên):ví dụ 125: -125
Storing Reals: integral part and a fractional part(Phần nguyên và phần phân số)
Storing Images(lưu trữ hình ảnh): raster graphics(bitmap)and vector graphics
Raster graphics(bitmap)
+ JPEG (Joint Photographic Experts Group) uses the True- Color scheme, but compresses the image to reduce the number of bits
+GIF (Graphic Interchange Format), on the other hand, uses the indexed color scheme.
+Softwares: Photoshop, PhotoImpact, Corel Painter
Vector graphics
Unsigned integer(Số nguyên không dấu) : nhận giá trị 0 và giá trị dương
Sign-and-Magnitude Representation(biểu diễn ký hiệu và đọ lớn)
Two's Complementing(bù 2): đến số 1 lật ngược bit 1->0 0-> 1 VD:00111000
11001000
Floating-Point Representation for a Real number(biểu diễn số phẩy động cho số thựcc

CHƯƠNG 4:

1. **LOGIC OPERATIONS** : AND ,NOT ,OR ,XOR (BOOLEAN algebra)



NOT

x	NOT x
0	1
1	0

AND

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

The **NOT** operator is a unary operator: it takes only one input. The output bit is the complement of the input.
The **AND** operator is a binary operator: it takes two inputs.The output bit is 1 if both inputs are 1s and the output is 0 in the other three cases
The **OR** operator is a binary operator: it takes two inputs.The output bit is 0 if both inputs are 0s and the output is 1 in other three cases.

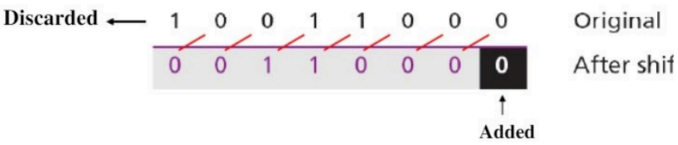
The **XOR** operator is a binary operator like the OR operator, with only one difference: the output is 0 if both inputs are 1s.

2. SHIFT OPERATIONS (TOÁN DỊCH CHUYỂN)

Shift operations move the bits in a pattern, changing the positions of the bits.

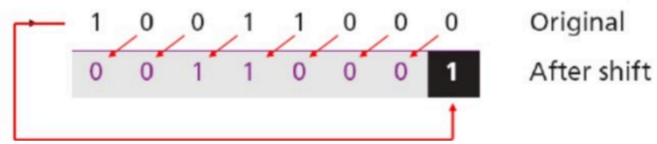
Có 2 loại : logical shift operations (toán tử dịch chuyển logic) and arthmetic shift operations(toán tử dịch chuyển số học)

Logical shift operations



Logical circular shift (Rotate)

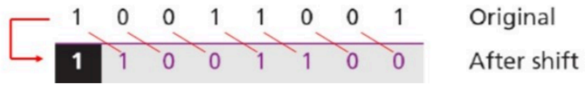
(Dich trai)



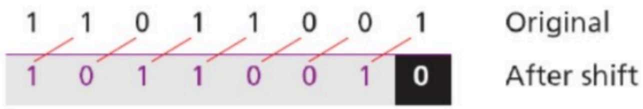
Arithmetic shift operations

Arithmetic shift operations assume that the bit pattern is asigned integer in **two's complement (bu 2)**format

Arithmetic right shift is used to divide an integer by two, while arithmetic left shift is used to multiply an integer by two.



Dich phai



CHƯƠNG 5 : COMPUTER ORGANAZATION(TỔ CHỨC MÁY TÍNH)

LÝ THUYẾT

1.LAN (mạng cục bộ):

A LAN is usually privately owned and connects some hosts in a single office, building, or campus.

A LAN can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video devices.

Each host in a LAN has an identifier, an address, that uniquely defines the host in the LAN.

A packet sent by a host to another host carries both the source host's and the destination host's addresses.

2.WAN (mạng diện rộng)

A WAN is also an interconnection of devices capable of communication.

A WAN has a wider geographical span, spanning a town, a state, a country, or even the world, however, a LAN is normally limited in size, spanning an office, a building, or a campus.

A WAN interconnects connecting devices such as switches, routers, or modems, however, a LAN interconnects hosts.

A WAN is normally created and run by communication companies and leased by an organization that uses it, however, a LAN is normally privately owned by the organization that uses it.

3.THE INTERNET

An Internet is two or more networks that can communicate with each other and is composed of thousands of interconnected networks.

The Internet is as several backbones, provider networks, and customer networks. Backbones at top level are large networks owned by some communication companies. Provider networks at second level use the services of the backbones for a fee.

Customer networks are networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services. Backbones and provider networks are also called Internet Service Providers (ISPs). The backbones are often referred to as international ISPs.

4.TCP/IP: gồm 5 player :

1. Tầng ứng dụng (Application):
2. Tầng giao vận (Transport):
3. Tầng mạng (Network):
4. Tầng vật lý (Physical):
5. Tần liên kết dữ liệu (Data link)

-Protocol Layering A protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively in Internet. we need a protocol at each layer, or protocol layering.

-The TCP/IP (Transmission Control Protocol / Internet Protocol) is a protocol suite (a set of protocols organized in different layers) used in the Internet today.

-It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality.

+Addressing and Packet Names IN TCP/IP:

-Any communication that involves two parties needs source and destination addresses. we normally have only four because the physical layer (data exchange is a bit) does not need addresses.

-There is a relationship between the layer, the address used in that layer, and the packet name at that layer.

a . Application player

-*We start from the fifth layer and move to the first layer.*

-The fifth layer of the TCP/IP protocol is called the application layer.The application layer provides services to the user. Communication is provided using a logical connection.

+Application-Layer Paradigms(mô hình tầng ứng dụng)

Two paradigms: -the client-server paradigm
- the peer-to-peer paradigm

+Applications of Standard Client-Server

World Wide Web (WWW) and its vehicle HyperText Transfer Protocol (HTTP), file transfer protocol (FTP), secure shell (SSH), email,

+DNS in the Internet

DNS is a protocol that can be used in different platforms. The domain name space (tree) was originally divided into three different sections generic domains, country domains, inverse domain.

Generic Domains : define registered hosts

b.Transport player

The transport layer in the TCP/IP suite is located between the application layer and the network layer. It provides services to the application layer and receives services from the network layer.

The transport layer acts as a liaison between a client program and a server program.

+ Process-to-Process Communication

-The Transport-layer protocol provides process-to-process communication . A process is an application-layer entity (running program) that uses the services of the transport layer.

-The network layer is responsible for communication at the computer level and can deliver the message only to the destination computer. A transport-layer protocol is responsible for delivery of the message to the appropriate process.

+Addressing: Port Numbers

-For communication, we must define the local host (IP), local process, remote host (IP), and remote process. To define the processes, we need second identifiers called port numbers. In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535 (16 bits).

-The client program defines itself with an ephemeral port number that is recommended to be greater than 1023 for some client/server programs to work properly. The server process must also define itself with a port number.

+Translayer-Layer Protocols: UDP,TCP

The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message using UDP takes much less interaction between the sender and receiver than using TCP.

-UDP packets, called user datagrams (format as below), have a fixed-size header of 8 byte and the total length needs to be less 65 535 bytes.

-Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol. TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service. At the transport layer, TCP groups a number of bytes together into a packet called a segment.

-TCP adds a header to each segment (for control purposes) and delivers the segment (format as below) to the network layer for transmission. The segments are encapsulated in an IP datagram and transmitted.

c. Network player

-The network layer in the TCP/IP protocol suite is responsible for the host-to-host delivery of messages.

-The network layer accepts a packet from a transport layer, encapsulates the packet in a datagram, and delivers the packet to the data-link layer.

-At the destination host , the datagram is de-capsulated, the packet is extracted and delivered to the corresponding transport layer.

+Network-Layer Protocols

-The main protocol is called the Internet Protocol (IP) . IPv4 and IPv6 are in use today.

-There are three common notations to show an IP address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16).

d . Data link player

The TCP/IP suite does not define any protocol in the data-link layer. This layer is the territories of networks that when connected make up the Internet. These networks, wired or wireless, receive services and provide services to the network layer.

Nodes and Links

-Communication at the data-link layer is node-to-node .Data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point. Theses LANs and WANs are connected by routers.

It is customary to refer to the two end hosts and the routers as nodes and the networks in between as links.

Wred LANs： Ethernet

Ethernet LAN was developed in 19fi0s by Robert Metcalfe and David Boggs. Standard Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps), and 10 Gigabit Ethernet (10 Gbps).

A frame carries some information such as the source address (48 bits), the destination address (48 bits), the type of data, the actual data, and some other control bits as a guard to help checking the integrity of data during transition.

Wireless Ethernet

Wireless Ethernet or WiFi is a wireless LAN. Two kinds of services: the basic service set (BSS) and the extended service set (ESS). The second service uses an extra device (access point or AP) that serves as a switch for connection to other LANs or WANs.

Cable Service

Cable networks were originally created to provide access to TV programs. Cable TV network can also support DSL technology that provides high-data-rate connections for residential subscribers over the local loop.

Wireless WAN:

WiMaThe **worldwide Interoperability Access (WiMax)** is the wireless version of DSL or Cable connection to the Internet. It provide two types of services (fixed WiMax) to connect the main station to fixed station or to mobile stations such as cellular phones

e .physical player

-The role of the physical layer is to transfer the bits received from the data-link layer and convert them to electromagnetic signals for transmission.
-After the bits are converted to signals, the signals are delivered to the transmission media.

CHƯƠNG 7 ; OPERATING SYSTEMS (HỆ ĐIỀU HÀNH)

1.A computer is a system composed of two major components: hardware and software

-Computer hardware is the physical equipment (thiết bị vật lý)
-Software is the collection of programs that allows the hardware to do its job. Computer software is divided into two broad categories: the operating system and application programs.(hệ điều hành và các chương trình ứng dụng)
-Application programs (chương trình ứng dung)use the computer hardware to solve users' problems. The operating system, on the other hand, controls the access to hardware by users.

1 số đinh nghĩa về hệ điều hành (operating system)

-An operating system is an interface between the hardware of a computer and the user (programs or humans).
Hệ điều hành là giao diện giữa phần cứng của máy tính và người dùng (chương trình hoặc con người).
-An operating system is a program (or a set of programs) that facilitates the execution of other programs.
-Hệ điều hành là một chương trình (hoặc một tập hợp các chương trình) hỗ trợ việc thực thi các chương trình khác.
-An operating system acts as a general manager supervising the activity of each component in the computer system.
-(Hệ điều hành đóng vai trò là tổng giám đốc, giám sát hoạt động của từng thành phần trong hệ thống máy tính)

Two major design goals of an operating system are:(2 mục tiêu)

❑ Efficient use of hardware.(sd phần cứng hiệu quả)
❑ Ease of use of resources.(dễ dàng sử dụng các nguồn tài nguyên)

2.EVOLUTION (TIẾN HÓA)

a.Batch systems (hệ thống hành loạt)

-Designed (thuyết kế) in the 1950s to control mainframe computers
-Each program to be executed was called a **job**(mỗi ct đc thực thi gọi là 1 công việc)
-A programmer who wished to execute a job sends a request to the operating system.(Một lập trình viên muốn thực hiện công việc sẽ gửi yêu cầu đến hệ điều hành.)

b.Time-sharing systems: (hệ thống chia sẻ thời gian)

-To use computer system resources efficiently, multiprogramming (đa chương trình) was introduced
-Multiprogramming brought the idea of time sharing: resources could be shared between different jobs, with each job being allocated a portion of time to use a resource.**(** với mỗi công việc được phân bổ một phần thời gian để sử dụng tài nguyên).

c.Personal systems

-When personal computers were introduced, there was a need for an operating system for this new type of computer.
-During this era, single-user operating systems such as DOS (Disk Operating System) were introduced

d.Parallel systems (hệ thống song song)

-The need for more speed and efficiency led to the design of parallel systems: multiple CPUs on the same machine.
-Each CPU can be used to serve one program or a part of a program, which means that many tasks can be accomplished in parallel instead of serially. The operating systems required for this are more complex than those that support single CPUs.

e.Distributed systems (hệ thống phân phối)

A job that was previously done on one computer can now be shared between computers that may be thousands of miles apart **(** được chia sẻ giữa các máy tính có thể cách nhau hàng ngàn dặm)**.** Distributed systems combine features of the previous generation with new duties such as controlling security.

f.Real-time systems (hệ thống thời gian thực)

-A real-time system (**RTOS**) is expected to do a task within a specific time constraint.(yêu câù thực hiện một nhiệm vụ trong một giới hạn thời gian cụ thể)
-They are used with real-time applications, which monitor, respond to or control external processes or environments.(giám sát ,phản hồi hoặc kiểm soát các quy trình hoặc môi trường bên ngoài.

3.COMPONENTS OF OS (THÀNH PHẦN CỦA HỆ ĐIỀU HÀNH)

A modern operating system has at least four duties: *memory manager(quản lý bộ nhớ)*, process manager (quản lý tiến trình), device manager (quản lý thiết bị) and file manager (quản lý tiệp tin)

a.User interface (giao diện người dùng)

-Each operating system has a user interface, a program that accepts requests from users (processes) and interprets them for the rest of the operating system.

-A user interface in some operating systems, such as UNIX, is called a shell. In others, it is called a window to denote that it is menu driven and has a GUI (graphical user interface) component.

b.Memory manager (quản lý bộ nhớ)

-One of the responsibilities of a modern computer system is memory management. Although the memory size of computers has increased tremendously in recent years, so has the size of the programs and data to be processed.

-Memory allocation must be managed to prevent applications from running out of memory. Operating systems can be divided into two broad categories of memory management: monoprogramming and multiprogramming.

+Monoprogramming (lập trình đơn)

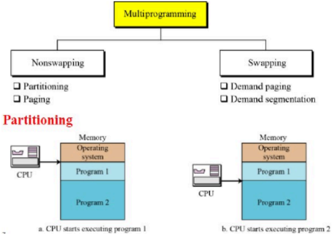
-In monoprogramming, most of the memory capacity is dedicated to a single program; only a small part is needed to hold the operating system.

-In this configuration, the whole program is in memory for execution. (Trong cấu hình này, toàn bộ chương trình nằm trong bộ nhớ để thực thi)**.** When the program finishes running, the program area is occupied by another program.

+Multiprogramming (đa chương trình)

- more than one program is in memory at the same time, and they are executed concurrently (nhiều chương trình nằm trong bộ nhớ cùng lúc và chúng được thực thi đồng thời), with the CPU switching rapidly between the programs.

+Categories of multiprogramming (thể loại đa chương trình)



+Virtual memory

Demand paging and demand segmentation mean that, when a program is being executed, part of the program is in memory and part is on disk

c.Process manager (quản lý tiến trình)

-A program is a non-active set of instructions stored on disk.

(Chương trình là một tập hợp các hướng dẫn không hoạt động được lưu trữ trên đĩa.**)**

-A program becomes a job from the moment it is selected for execution until it has finished running and becomes a program again.

(Một chương trình trở thành một công việc kể từ thời điểm nó được chọn để thực thi cho đến khi nó chạy xong và trở lại thành một chương trình.)

-A process is a program in execution. It is a program that has started but has not finished.

+State diagrams & Queuing

+Process synchronization (đồng bộ hóa tiến trình)

The whole idea behind process management is to **s**ynchronize different processes with different resources (đồng bộ hóa các quy trình khác nhau với các tài nguyên khác nhau). Whenever resources can be used by more than one user (or process, in this case), we can have two problematic situations: deadlock and starvation.

d.Device manager (trình quản lý các thiết bị)

-The device manager, or input/output manager, is responsible for access to input/ output devices. There are limitations on the number and speed of input/output devices in a computer system.

- The device manager monitors every input/output device constantly to ensure that the device is functioning properly.

-The device manager maintains a queue for each input/output device or one or more queues for similar input/output devices.

-The device manager controls the different policies for accessing input/output devices.

e,File manager (quản lí tệp tin)

Operating systems today use a file manager to control access to files (Sd quản lý tệp để kiểm soát quyền truy cập vào tệp)

- Controls access to files. (kiểm soát quyền truy cập vào thông tin)
- Supervises the creation, deletion, and modification of files. (giám sát việc tạo xóa và sửa đổi tệp tin)
- controls the naming of files. (kiểm soát việc đặt tên tệp tin
-supervises the storage of files. (kiểm soát việc lưu trữ các tệp tin)

is responsible for archiving and backups. (chịu trách nhiệm lưu trữ và sao lưu)

4.+A SURVEY OF OPERATING SYSTEMS (UNIX) khảo sát hệ điều hành

-Developed in 1969 by Thomson and Ritchie of the Computer Science Research Group at Bell Laboratories.

-UNIX has been a popular operating system among computer programmers and computer scientists.

-UNIX is a multiuser, multiprocessing, portable operating system. It is designed to facilitate programming, text processing and communication
(UNIX là một hệ điều hành di động, đa người dùng, đa xử lý. Nó được thiết kế để hỗ trợ lập trình, xử lý văn bản và giao tiếp)

+A SURVEY OF OPERATING SYSTEMS (LINUX)

-Linux In 1991, Linus Torvalds**,** a Finish student at the University of Helsinki at the time, developed a new operating system that is known today as **Linux.**

-The initial kernel, which was similar to a small subset of UNIX (một tập hợp con nhỏ của unix), has grown into a full-scale operating system today.

-The Linux 2.0 kernel, released in 199fi, was accepted as a commercial operating system: it has all features traditionally attributed to UNIX

+A SURVEY OF OPERATING SYSTEMS (LINUX)

Windows In the late 1980s Microsoft**,** under the leadership of Dave Cutler, started development of a new single-user operating system to replace MS-DOS (Microsoft Disk Operating System).

CHƯƠNG 8:ALGORITHMS (THUẬT TOÁN)

Algorithms: a step by step solution to a proble

1. THREE BASIC CONSTRUCTS (3 CẤU TRÚC CƠ BẢN)

a.Sequence (trình tự)

-The first construct is called the **sequence.**

-An algorithm, and eventually a program, is a sequence of instructions, which can be a simple instruction or either of the other two constructs. (Một thuật toán, và cuối cùng là một chương trình, là một chuỗi các lệnh, có thể là một lệnh đơn giản hoặc một trong hai cấu trúc còn lại.)

b.Decision selection(quyết định) (lựa chọn)

Some problems cannot be solved with only a sequence of simple instructions.we need to test a condition.

c.Repetition (sự lặp lại)

The same sequence of instructions must be repeated. We handle this with the repetition or loop construct.

2. ALGORITHM REPRESENTATION (biểu diễn thuật số)

-UML (Unified Modeling Language) is a pictorial representation of an algorithm. It hides all the details of an algorithm in an attempt to give the 'big picture' and to show how the algorithm flows from beginning to end.

(Ngôn ngữ mô hình hóa thống nhất (UML) là biểu diễn bằng hình ảnh của một thuật toán)

-Pseudocode is an English-language-like representation of an algorithm.

3. BASIC ALGORITHMS (thuật toán cơ bản)

Summation (tính tổng) phổ biến

A summation algorithm has three logical parts:

1. Initialization of the sum at the beginning.(Khởi tạo tổng lúc đầu)
2. The loop, which in each iteration adds a new integer to the sum.

(Vòng lặp, trong mỗi lần lặp sẽ thêm một số nguyên mới vào tổng.)

3.Return of the result after exiting from the loop.

(Trả về kết quả sau khi thoát khỏi vòng lặp.)

Smallest and largest (nhỏ nhất và lớn nhất)

If we put this construct in a loop, we can find the largest of a list of integers.

Finding the smallest integer among a list of integers is similar, with two minor differences.

First, we use a decision construct to find the smaller of two integers.

Second, we initialize with a very large integer instead of a very small one.(Thứ hai, chúng ta khởi tạo với một số nguyên rất lớn thay vì một số nguyên rất nhỏ)

Sorting (sắp xếp)

Sorting is the process by which data is arranged according to its values.
(Sắp xếp là quá trình sắp xếp dữ liệu theo các giá trị của nó.)

Common sorting algorithms: các thuật toán sx phổ biến

Bubble/Shell sort:
Insertion sort
Selection sort
Quick sort
Merge sort

Selection Sort: (lựa chọn sx)

The list to be sorted is divided into two sublists—sorted and unsorted— which are separated by an imaginary wall. We find the smallest element from the unsorted sublist and swap it with the element at the beginning of the unsorted sublist.

After each selection and swap, the imaginary wall between the two sublists moves one element ahead, increasing the number of sorted elements and decreasing the number of unsorted ones.

Bubble sorts:

In the bubble sort method, the list to be sorted is also divided into two sublists—sorted and unsorted. The smallest element is bubbled up from the unsorted sublist and moved to the sorted sublist.

After the smallest element has been moved to the sorted list, the wall moves one element ahead, increasing the number of sorted elements and decreasing the number of unsorted ones.
Each time an element moves from the unsorted sublist to the sorted sublist, one sort pass is completed

Insertion sort :the item that goes into the sorted list is always the first item in the unsorted list.
(Trong sắp xếp chèn, mục nằm trong danh sách đã sắp xếp luôn là mục đầu tiên trong danh sách chưa sắp xếp.)

Search Algorithms:

Có 2 danh sách tìm kiếm cơ bản : 1.**sequential (linear) search**
2.binary search

+**Sequential search (tuần tự)**can be used to locate an item in any list,
+whereas **binary search** requires the list first to be sorted (yêu câù danh sách phải tìm kiếm trước)

Sequential search (tuần tự) A linear search or sequential search is a method for finding an element within a list . It sequentially checks each element of the list until a match is found or the whole list has been searched.(là phương pháp tìm kiếm một phần tử trong danh sách. Nó tuần tự kiểm tra từng phần tử của danh sách cho đến khi tìm thấy kết quả khớp hoặc toàn bộ danh sách đã được tìm kiếm.)

Binary Search

Binary search, also known as half-interval search, logarithmic search, or binary chop, is a search algorithm that finds the position of a target value within a sorted array.

Binary search compares the target value to the middle element of the array.

CHƯƠNG 9 : PROGRAMMING LANGUAGES

1.EVOLATION

To write a program for a computer, we must use a computer language. A computer language is a set of predefined words that are combined into a program according to predefined rules (syntax). Over the years, computer languages have evolved from machine language to high- level languages.

a.Machine languages (ngôn ngữ máy)

-In the earliest days of computers, the only programming languages available were machine languages.
-Each computer had its own machine language, which was made of streams of 0s and 1s.
-Use eleven lines of code to read two integers, add them, and print the result. These lines of code, when written in machine language, make eleven lines of binary code, each of 16 bits, as shown in

b.Assembly languages.

-The next evolution in programming came with the idea of replacing binary code for instruction and addresses with symbols or mnemonics. Because they used symbols, these languages were first known as symbolic languages.

-The set of these mnemonic languages were later referred to as assembly languages. The assembly language for our hypothetical computer to replace the machine language

c. High-level languages

- They still required programmers to concentrate on the hardware they were using. Working with symbolic languages was also very tedious, because each machine instruction had to be individually coded.

-The desire to improve programmer efficiency and to change the focus from the computer to the problem being solved led to the development of high-level language

-Over the years, various languages, most notably BASIC, COBOL, Pascal, Ada, C, C+ +, and Java, were developed.

2.TRANSLATION (DỊCH THUẬT)

-To run the program on a computer, the program needs to be translated into the machine language of the computer on which it will run.

-The program in a high-level language is called the source program (chượgn trình nguồn).

-The translated program in machine language is called the object program (chương trình đối tượng)

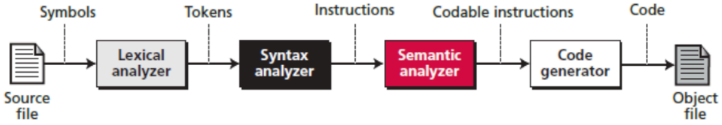
Two methods are used for translation: compilation (biên dịch) and interpretation (phiên dịch)

+Compilation : A compiler normally translates the whole source program into the object program.

+Interpretation : Some computer languages use an interpreter to translate the source program into the object program. Interpretation refers to the process of translating each line of the source program into the corresponding line of the object program and executing the line.

Translation process (quá trình dịch thuật)

Compilation and interpretation differ in that the first translates the whole source code before executing it, while the second translates and executes the source code a line at a time.



-Lexical analyzer (bộ phân tích từ vựng)reads the source code, symbol by symbol, and creates a list of tokens in the source language.

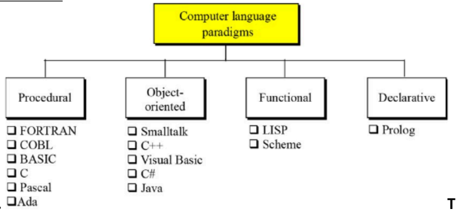
-Syntax analyzer (bộ phân tích cú pháp) parses a set of tokens to find instructions.

-Semantic analyzer (bộ phân tích ngữ nghĩa)checks the sentences created by the syntax analyzer to be sure that they contain no ambiguity.

-Code generator (trình tạo mã)After unambiguous instructions are created by the semantic analyzer, each instruction is converted to a set of machine language instructions for the computer on which the program will run.

1. PROGRAMMING PARADIGMS (mô hình lập trình)

We divide computer languages into four paradigms: procedural, object-oriented, functional, and declarative



a. The procedural paradigm

-In the procedural paradigm (or imperative paradigm) we can think of a program as an active agent that manipulates passive objects. We encounter many passive objects in our daily life: a stone, a book, a lamp, and so on. A passive object cannot initiate an action by itself, but it can receive actions from active agents.

b. The object-oriented paradigm

-The object-oriented paradigm deals with active objects instead of passive objects. We encounter many active objects in our daily life: a vehicle, an automatic door, a dishwasher, and so on.

-The actions to be performed on these objects are included in the object: the objects need only to receive the appropriate stimulus from outside to perform one of the actions.

c.The functional paradigm

-In the functional paradigm a program is considered a mathematical function. In this context, a function is a black box that maps a list of inputs to a list of outputs.

d.The declarative paradigm

-A declarative paradigm uses the principle of logical reasoning to answer queries. It is based on formal logic defined by Greek mathematicians and later developed into first-order predicate calculus.

3.COMMON CONCEPTS (khái niệm chung)

1.Identifiers

One feature present in all procedural languages, as well as in other languages, is the identifier—that is, the name of objects. Identifiers allow us to name objects in the program.

2.Data types (kiểu dữ liệu)

A data type defines a set of values and a set of operations that can be applied to those values. The set of values for each type is known as the domain for the type.

Most languages define two categories of data types: simple types (Primitive in Java) and composite types (Non-Primitive in Java).

3.Variables (biến)

Variables are names for memory locations. Although the addresses are used by the computer internally, it is very inconvenient for the programmer to use addresses.

A programmer can use a variable, such as count, to store the integer value of a count number received in a test.

2. Literals

A literal is a predetermined value used in a program. For example, if we need to calculate the area of circle when the value of the radius is stored in the variable r, we can use the expression 3.14 × r2, in which the approximate value of π (pi) is used as a literal.

5.Constants

-The use of literals is not considered good programming practice unless we are sure that the value of the literal will not change with time (such as the value of π in geometry).

-However, most literals may change value with time. For example, if a sales tax is 8 per cent this year, it may not be the same next year.

6.Inputs and Outputs

-Almost every program needs to read and/or write data. These operations can be quite complex, especially when we read and write large files. Most programming languages use a predefined function for input and output.

Input : Data is input by either a statement or a predefined function.

CHƯƠNG 10 :SOFTWARE ENGINEERING

1 - The software lifecycle

-The software development life cycle (SDLC), also referred to as the application development life-cycle, is a process for planning, creating, testing, and deploying an information system.

-The systems development life cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

-There are usually six stages in this cycle: requirement analysis, design, development and testing, implementation, documentation, and evaluation.

a.Development process models

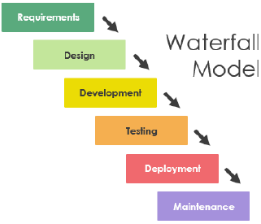
One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. The models specify the various stages of the process and the order in which they are carried out. The most used, popular and important SDLC models are given below:

-Waterfall model
-V model
-Incremental model
-RAD model
-Agile model
-Iterative model
-Spiral model
-Prototype model

b. Waterfall model

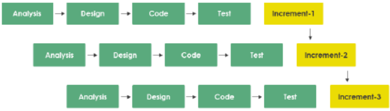
-One very popular model for the software development process is known as the waterfall model (Figure 8.2). In this model, the development process flows in only one direction.

-This means that a phase cannot be started until the previous phase is completed.



The incremental model

-In the incremental model, software is developed in a series of steps. The developers first complete a simplified version of the whole system. This version represents the entire system but does not include the details. Figure 10.3 shows the incremental model concept.



2- Analysis phase

-The development process starts with the analysis phase. This phase results in a specification document that shows what the software will do without specifying how it will be done.
-The analysis phase can use two separate approaches, depending on whether the implementation phase is done using a procedural programming language or an object-oriented language.

a. Procedure-oriented analysis

Procedure-oriented analysis—also called structured analysis or classical analysis—is the analysis process used if the system implementation phase will use a procedural language.
The specification in this case may use several modeling tools, but we discuss only a few of them here.

b.Object-oriented analysis

-Object-oriented analysis is the analysis process used if the implementation uses an

-Object-oriented language. The specification document in this case may use several tools, but we discuss only a few of them here.

3 - Design phase

-The design phase defines how the system will accomplish what was defined in the analysis phase. In the design phase, all components of the system are defined

a.Procedure-oriented design

-In procedure-oriented design we have both procedures and data to design. We discuss a category of design methods that concentrate on procedures. In procedure-oriented design, the whole system is divided into a set of procedure or modules.

b. Object-oriented design

In object-oriented design, the design phase continues by elaborating the details of classes.

4- Implementation phase

-In the waterfall model, after the design phase is completed, the implementation phase can start.
-In this phase the programmers write the code for the modules in procedure-oriented design, or write the program units to implement classes in object-oriented design. There are several issues to mention in each case.

a. Choice of language

-In a procedure-oriented development, the project team needs to choose a language or a set of languages from among the procedural languages.

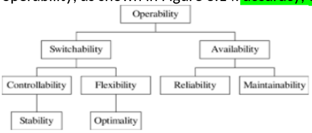
-Although some languages like C++ are considered to be both a procedure—and an object-oriented language—normally an implementation uses a purely procedural language such as C. In object-oriented cases, both C++ and Java, C# are common.

b. Software quality

-The quality of software created at the implementation phase is a very important issue. A software system of high quality is one that satisfies the user's requirements, meets the operating standards of the organization, and runs efficiently on the hardware for which it was developed. However, if we want to achieve a software system of high quality, we must be able to define some attributes of quality.

c.Operability

Operability refers to the basic operation of a system. Several measures can be mentioned for operability, as shown in Figure 8.14: accuracy, efficiency, reliability, security, timeliness, and usability.



d.Maintainability

-Maintainability refers to the ease with which a system can be kept up to date and running correctly. -Many systems require regular changes, not because they were poorly implemented, but because of changes in external factors.

e.Transferability

-Transferability refers to the ability to move data and/or a system from one platform to another and to reuse code. In many situations this is not an important factor. On the other hand, if we are writing generalized software, it can be critical.

5 - Testing phase

The goal of the testing phase is to find errors, which means that a good testing strategy is the one that finds most errors. There are two types of testing: white-box and black-box

a. Glass-box testing (or white-box testing)

-Glass-box testing (or white-box testing) is based on knowing the internal structure of the software. The testing goal is to check to determine whether all components of the software do what they are designed to do.

-Glass-box testing assumes that the tester knows everything about the software. In this case, the software is like a glass box in which everything inside the box is visible.

-Glass-box testing is done by the software engineer or a dedicated team.

b.Black testing

-Black-box testing gets its name from the concept of testing software without knowing what is inside it and without knowing how it works. In other words, the software is like a black box into which the tester cannot see.

-Black-box testing tests the functionality of the software in terms of what the software is supposed to accomplish, such as its inputs and outputs. Several methods are used in black-box testing, discussed below.

c. Exhaustive testing

-The best black-box test method is to test the software for all possible values in the input domain. However, in complex software the input domain is so huge that it is often impractical to do so.

d. Random testing

In random testing, a subset of values in the input domain is selected for testing. It is very important that the subset be chosen in such a way that the values are distributed over the domain input. The use of random number generators can be very helpful in this case.

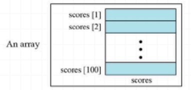
e. Boundary-value testing

-Errors often happen when boundary values are encountered. For example, if a module defines that one of its inputs must be greater than or equal to 100, it is very important that module be tested for the boundary value 100. If the module fails at this boundary value, it is possible that some condition in the module's code such as x . 100 is written as x . 100.

CHƯƠNG 11: DATA STRUCTURE

1 - ARRAYS

-An array is a sequenced collection of elements, of the same data type. We can refer to the elements in the array as the first element, the second element, and so forth until we get to the last element.
If we were to put our 100 scores into an array, we could designate the elements as scores[1], scores[2], and so on.
-The index indicates the ordinal number of the element, counting from the beginning of the array. The elements of the array are individually addressed through their subscripts (Figure 11.3).
-The array as a whole has a name, scores, but each score can be accessed individually using its subscript.



Loops:We can use loops to read and write the elements in an array. We can also use loops to process elements. Now it does not matter if there are 100, 1000, or 10,000 elements to be processed—loops make it easy to handle them all.We can use an integer variable to control the loop, and remain in the loop as long as the value of this variable is less than the total number of elements in the array

Multi-dimensional arrays: The arrays discussed so far are known as one-dimensional arrays because the data is organized linearly in only one direction.Many applications require that data be stored in more than one dimension.

Memory layout: The indexes in a one-dimensional array directly define the relative positions of the element in actual memory. A two-dimensional array, however, represents rows and columns. How each element is stored in memory depends on the computer. Most computers use row-major storage, in which an entire row of an array is stored in memory before the next row. However, a computer may store the array using column-major storage, in which the entire column is stored before the next column.

Operations on array: Although we can apply conventional operations defined for each element of an array there are some operations that we can define on an array as a data structure.
The common operations on arrays as structures are searching, insertion, deletion, retrieval, and traversal

2- RECORDS

-A record is a collection of related elements, possibly of different types, having a single name. Each element in a record is called a field.
-A field is the smallest element of named data that has meaning. A field has a type, and exists in memory.Fields can be assigned values, which in turn can be accessed for selection or manipulation. A field differs from a variable primarily in that it is part of a record.

Record name versus field name: Just like in an array, we have two types of identifier in a record: the name of the record and the name of each individual field inside the record. The name of the record is the name of the whole structure, while the name of each field allows us to refer to that field.

Comparison of records and arrays: We can conceptually compare an array with a record. This helps us to understand when we should use an array and when a record. An array defines a combination of elements, while a record defines the identifiable parts of an element.

Array of records : If we need to define a combination of element and at the same time some attributes of each element, we can use an array of records.

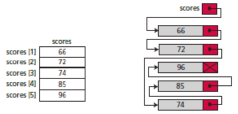
3 - LINKED LIST

-A linked list is a collection of data in which each element contains the location of the next element—that is, each element contains two parts: data and link. The data part holds the value information: the data to be processed.
-The link is used to chain the data together, and contains a pointer (an address) that identifies the next element in the list. In addition, a pointer variable identifies the first element in the list. The name of the list is the same as the name of this pointer variable.

Arrays versus linked lists

-Both an array and a linked list are representations of a list of items in memory. The only difference is the way in which the items are linked together. In an array of records, the linking tool is the index.

The element scores[3] is linked to the element scores[4] because the integer 4 comes after the integer 3. In a linked list, the linking tool is the link that points to the next element—the pointer or the address of the next element.



a. Array representation b. Linked list representation

Linked list names versus nodes names

As for arrays and records, we need to distinguish between the name of the linked list and the names of the nodes, the elements of a linked list. A linked list must have a name.

Operations on linked lists (inserting)

-Before insertion into a linked list need searching algorithm.
-If the flag returned from the searching algorithm is false, will allow insertion,
-otherwise abort the insertion algorithm Four cases can arise:
1.Inserting into an empty list.
2.Insertion at the beginning of the list.
3.Insertion at the end of the list.
4.Insertion in the middle of the list.

Operations on linked lists (searching)

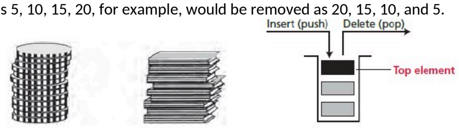
-The search algorithm for a linked list can only be sequential because the nodes in a linked list have no specific names (unlike the elements in an array) that can be found using a binary search.

-However, since nodes in a linked list have no names, we use two pointers, pre (for previous) and cur (for current).

4- Introduction: Stack, Queue, Tree, graph

STACKS

-A stack is a restricted linear list in which all additions and deletions are made at one end, the top. If we insert a series of data into a stack and then remove it, the order of the data is reversed. Data input as 5, 10, 15, 20, for example, would be removed as 20, 15, 10, and 5.



Stack of coins Stack of books Computer stack

Operations on stacks

Although we can define many operations for a stack, there are four basic operations, stack, push, pop, and empty, which we define in this chapter.

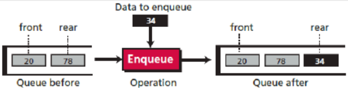
QUEUES

-A queue is a linear list in which data can only be inserted at one end, called the rear, and deleted from the other end, called the front. These restrictions ensure that the data are processed through the queue in the order in which it is received. In other words, a queue is a first in, first out (FIFO) structure.

The queue operation

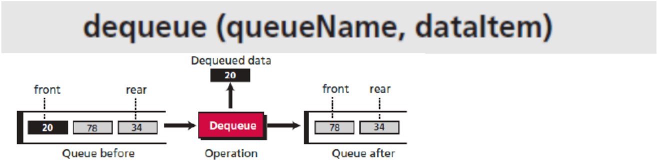
The queue operation creates an empty queue. The following shows the format:

```
queue (queueName)
```



The dequeue operation

The dequeue operation deletes the item at the front of the queue. The following shows the format:



dequeue (queueName, dataItem)

-<u>A record</u> is a collection of related elements, possibly of different types, having a single name. Each element in a record is called a field.

-<u>A field</u> is the smallest element of named data that has meaning. A field has a type, and exists in memory.Fields can be assigned values, which in turn can be accessed for selection or manipulation. A field differs from a variable primarily in that it is part of a record.

<u>Record name versus field name:</u> Just like in an array, we have two types of identifier in a record: the name of the record and the name of each individual field inside the record. The name of the record is the name of the whole structure, while the name of each field allows us to refer to that field.

<u>Comparison of records and arrays:</u>We can conceptually compare an array with a record. This helps us to understand when we should use an array and when a record. An array defines a combination of elements, while a record defines the identifiable parts of an element.

<u>Array of records :</u>If we need to define a combination of element and at the same time some attributes of each element, we can use an array of records.

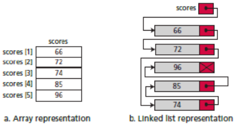
3 - LINKED LIST

-A linked list is a collection of data in which each element contains the location of the next element—that is, each element contains two parts: data and link. The data part holds the value information: the data to be processed.

-The link is used to chain the data together, and contains a pointer (an address) that identifies the next element in the list. In addition, a pointer variable identifies the first element in the list. The name of the list is the same as the name of this pointer variable.

<u>Arrays versus linked lists</u>

-Both an array and a linked list are representations of a list of items in memory. The only difference is the way in which the items are linked together. In an array of records, the linking tool is the index. The element scores[3] is linked to the element scores[4] because the integer 4 comes after the integer 3. In a linked list, the linking tool is the link that points to the next element—the pointer or the address of the next element.



a. Array representation

b. Linked list representation

<u>Linked list names versus nodes names</u>

As for arrays and records, we need to distinguish between the name of the linked list and the names of the nodes, the elements of a linked list. A linked list must have a name.

<u>Operations on linked lists (inserting)</u>

-Before insertion into a linked list need searching algorithm.

-If the flag returned from the searching algorithm is false, will allow insertion,

-otherwise abort the insertion algorithm Four cases can arise:

1.Inserting into an empty list.

2.Insertion at the beginning of the list.

3.Insertion at the end of the list.

4.Insertion in the middle of the list.

<u>Operations on linked lists (searching)</u>

-The search algorithm for a linked list can only be sequential because the nodes in a linked list have no specific names (unlike the elements in an array) that can be found using a binary search.

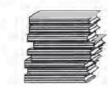
-However, since nodes in a linked list have no names, we use two pointers, pre (for previous) and cur (for current).

4- Introduction: Stack, Queue, Tree, graph

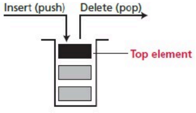
STACKS

-A stack is a restricted linear list in which all additions and deletions are made at one end, the top. If we insert a series of data into a stack and then remove it, the order of the data is reversed. Data input as 5, 10, 15, 20, for example, would be removed as 20, 15, 10, and 5.

Stack of coins



Stack of books



Computer stack

Operations on stacks

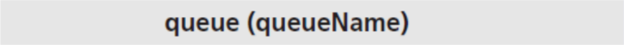
Although we can define many operations for a stack, there are four basic operations, stack, push, pop, and empty, which we define in this chapter.

QUEUES

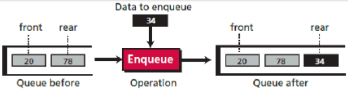
-A queue is a linear list in which data can only be inserted at one end, called the rear, and deleted from the other end, called the front. These restrictions ensure that the data are processed through the queue in the order in which it is received. In other words, a queue is a first in, first out (FIFO) structure.

The queue operation

The queue operation creates an empty queue. The following shows the format:

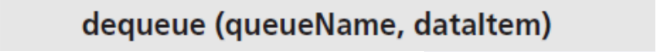


queue (queueName)

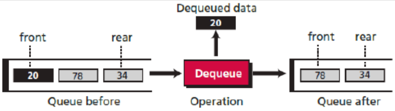


The dequeue operation

The dequeue operation deletes the item at the front of the queue. The following shows the format:



dequeue (queueName, dataItem)



CHƯƠNG 13 : FILE STRUCTURE

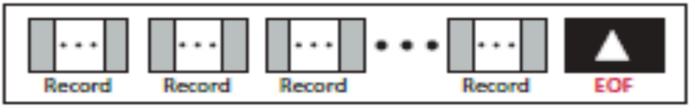
1 - Access methods

Files are stored on auxiliary or secondary storage devices. The two most common forms of secondary storage are disk and tape. Files in secondary storage can be both read from and written to. Files can also exist in forms that the computer can write to but not read

2.Sequential files

A sequential file is one in which records can only be accessed one after another from beginning to end. Figure 13.2 shows the layout of a sequential file. Records are stored one after another in auxiliary storage, such as tape or disk, and there is an EOF (end-of-file) marker after the last record.

s



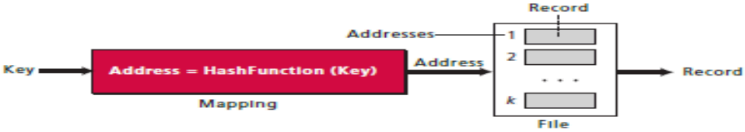
Sequential file

3.Indexed files

To access a record in a file randomly, we need to know the address of the record.

3. Hashed files

In an indexed file, the index maps the key to the address. A hashed file uses a mathematical function to accomplish this mapping. The user gives the key, the function maps the key to the address and passes it to the operating system, and the record is retrieved (Figure 10.5).



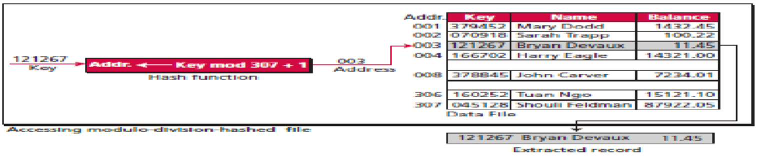
Hashing methods

-Direct hashing

-In direct hashing, the key is the data file address without any algorithmic manipulation. The file must therefore contain a record for every possible key. Although situations suitable for direct hashing are limited, it can be very powerful because it guarantees that there are no synonyms or collisions.

Modulo division hashing

-Also known as division remainder hashing, the modulo division method divides the key by the file size and uses the remainder plus 1 for the address. This gives the simple hashing algorithm that follows, where list_size is the number of elements in the file. The reason for adding a 1 to the mod operation result is that our list starts with 1 instead of 0:



Addr.	Key	Name	Balance
001	379452	Mary Dodd	1432.45
002	070918	Sarah Trapp	100.22
003	121267	Bryan Devaux	11.45
004	166702	Harry Eagle	14321.00
008	378845	John Carver	7234.01
306	160252	Tuan Ngo	15121.10
307	045128	Shouli Feldman	87922.05

Accessing modulo-division-hashed file

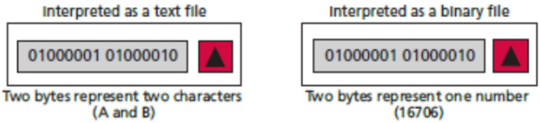
5. Collision resolution

-When we hash a new key to an address, we may create a collision.

-There are several methods for handling collisions, each of them independent of the hashing algorithm. That is, any hashing method can be used with any collision resolution method

6.Text versus Binary

Two terms used to categorize files: text files and binary files. A file stored on a storage device is a sequence of bits that can be interpreted by an application program as a text file or a binary file, as shown in Figure 13.15.


Interpreted as a text file
01000001 01000010
Two bytes represent two characters (A and B)
Interpreted as a binary file
01000001 01000010
Two bytes represent one number (16706)

a. Text File

A text file is a file of characters. It cannot contain integers, floating-point numbers, or any other data structures in their internal memory format. To store these data types, they must be converted to their character equivalent formats.

b. Binary files

A binary file is a collection of data stored in the internal format of the computer. In this definition, data can be an integer (including other data types represented as unsigned integers, such as image, audio, or video), a floating-point number, or any other structured data (except a file).

c.Directorior

-Directories are provided by most operating systems for organizing files. A directory performs the same function as a folder in a filing cabinet. However, a directory in most operating systems is represented as a special type of file that holds information about other files.

-A directory not only serves as a kind of index that tells the operating system where files are located on an auxiliary storage device, but can also contain other information about the files it contains, such as who has access to each file, or the date when each file was created, accessed, or modified.

CHƯƠNG 14 : DATABASES (xong)

Introduction

Data storage traditionally used individual, unrelated files, sometimes called flat-files.

Definition: A database is a collection of related, logically coherent, data used by the application programs in an organization.

Advantages of databases

-**Less redundancy** In a flat-file system there is a lot of redundancy. For example, in the flat-file system for a university, the names of professors and students are stored in more than one file.

-Inconsistency avoidance If the same piece of information is stored in more than one place, then any changes in the data need to occur in all places that data is stored. For example, if a female student marries and accepts the last name of her husband, the last name of the student needs to be changed in all files that hold information about the student. Lack of care may create inconsistency in the data.

-Efficiency A database is usually more efficient that a flat-file system, because a piece of information is stored in fewer locations.

-Data integrity In a database system it is easier to maintain data integrity (see Chapter 16) because a piece of data is stored in fewer locations.

-Confidentiality It is easier to maintain the confidentiality of the information if the storage of data is Centralized in one location.

Database management systems (hệ thống quản lý dữ liệu DBMS)

A database management system (DBMS) defines, creates, and maintains a database. The DBMS also allows controlled access to data in the database. A DBMS is a combination of five components: hardware, software, data, users, and procedures

Hardware is the physical computer system that allows access to data.

Software is the actual program that allows users to access, maintain, and update data

Data in a database is stored physically on the storage devices. In a database, data is a separate entity from the software that accesses it.

Users are the people who control and manage the databases and perform different types of operations on the databases in the database management system.

Procedures refer to general rules and instructions that help to design the database and to use a database management system.

1- Database architecture (Kiến trúc cơ sở dữ liệu)

The American National Standards Institute/Standards Planning and Requirements Committee (ANSI/SPARC) has established a three-level architecture for a DBMS: internal, conceptual, and external

a. **internal level** determines where data is actually stored on the storage devices. This level deals with low-level access methods and how bytes are transferred to and from storage devices.

-The internal level has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

+Physical representation of the DB on the computer.

+How the data is stored in the database.

+Physical implementation of the DB to achieve optimal run, time performance and storage space utilization, Storage space allocation for data and indexes, Record description for storage, Record placement, Data compression, encryption.

b. Conceptual level defines the logical view of the data. The data model is defined on this level, and the main functions of the DBMS, such as queries, are also on this level.

-The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.

+Usually, a representational data model is used to describe the conceptual schema when a database system is implemented.

+What data is stored in the database.

+The logical structure of the entire database as seen by DBA.

+The relationships among the data.

+Complete view of the data requirements of the organization, independent of any storage consideration.

+Represents: - Entities, attributes, relations , constraints on data , semantic information on data and security integrity information

c. External level interacts directly with the user (end users or application programs). It changes the data coming from the conceptual level to a format and view that is familiar to the users.

-The external or view level includes a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. The external level interacts directly with the user (end users or application programs). It changes the data coming from the conceptual level to a format and view that is familiar to the users.

+For example, one user may view dates in theform (day, month, year), while another mayview dates as (year, month, day).

+Consists of a number of different external views of the DB.

+The user’s view of the database

+Describes part of the DB for particular group of users.

+Provides a powerful and flexible security mechanism by hiding parts of the DB from certain users. The user is not aware of the existence of any attributes that are missing from the view.

+It permits users to access data in a way that is customized to their needs, so that the same data can be seen by different users in different ways, at the same time.

2.Database models (mô hình cơ sở dữ liệu) 3 mô hình

-A database model defines the logical design of data. The model also describes the relationships between different parts of the data. In the history of database design, three models have been in use:

a.the hierarchical model (mô hình phân cấp)

-In the hierarchical model, data is organized as an inverted tree

-There is one entity, which is called the root

b. the network model (mô hình mạng)

-In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths

c.the relational model (mô hình quan hệ) thinh hành nhất ngày nay

In the relational model, data is organized in two-dimensional tables called relations. There is no hierarchical or network structure imposed on the data. The tables or relations are, however, related to each other

3-The relationship database model (mô hình cơ sở dữ liệu quan hệ)

a. Relation

Relation, in appearance, is a two-dimensional table. The RDBMS organizes the data so that its external view is a set of relations or tables. This does not mean that data are stored as tables: the physical storage of the data is independent of the way in which the data is logically organized. A relation in an RDBMS has the following features:

+Name. Each relation in a relational database should have a name that is unique among other relations.

+*Attributes*. Each column in a relation is called an attribute. The attributes are the column headings in the table in Figure 14.6. Each attribute gives meaning to the data

+*Tuples*. Each row in a relation is called a tuple. A tuple defines a collection of attribute values. The total number of rows in a relation is called the cardinality of the relation.

b. Operations on relations

-In a relational database we can define several operations to create new relations based on existing ones. We define nine operations in this section: **insert, delete, update, select, project, join, union, intersection, and difference**. Instead of discussing these operations in the abstract, we describe each operation as defined in the database query language SQL

-Structured Query Language (SQL) is the language standardized by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) for use on relational databases

c. Insert

The insert operation is a unary operation—that is, it is applied to a single relation. The operation inserts a new tuple into the relation.

d.Delete

A unary operation. The operation deletes a tuple defined by a criterion from the relation.

e. Update

A unary operation that is applied to a single relation. The operation changes the value of some attributes of a tuple

f. Select

A unary operation. The tuples (rows) in the resulting relation are a subset of the tuples in the original relation.

g. Join (7)

A binary operation that combines two relations on common attributes.

h. Union (8)

The union operation takes two relations with the same set of attributes.

3 - Database design

-The first step normally involves a lot of interviewing of potential users of the database, for example in a university, to collect the information needed to be stored and the access requirements of each department.

-The second step is to build an entity–relation model (ERM) that defines the entities for which some information must be maintained, the attributes of these entities, and the relationship between these entities.

a. Entity–relation model (ERM)

In this step, the database designer creates an entity–relationship (E-R) diagram to show the entities for which information needs to be stored and the relationship between those entities. E-R diagrams uses several geometric shapes,

-**Rectangles** represent entity sets

-**Ellipses** represent attributes

-**Diamonds** represent relationship sets

-Lines link attributes to entity sets and link entity sets to relationship sets

b. From E-R diagrams to relations

Relations for entity sets

-For each entity set in the E-R diagram, we create a relation (table) in which there are n columns related to the n attributes defined for that set.

Relations for relationship sets

-For each relationship set in the E-R diagram, we create a relation (table). This relation has one column for the key of each entity set involved in this relationship and also one column for each attribute of the relationship itself if the relationship has attributes (not in our case).

c. Normalization

-Normalization is the process by which a given set of relations are transformed to a new set of relations with a more solid structure.

-Normalization is needed to allow any relation in the database to be represented, to allow a languages like SQL to use powerful retrieval operations composed of atomic operations, to remove anomalies in insertion, deletion, and updating, and reduce the need for restructuring the database as new data type are added.
-The normalization process defines a set of hierarchical normal forms (NFs). Several normal forms have been proposed, including 1NF, 2NF, 3NF, BCNF (Boyce–Codd Normal Form), 4NF, PJNF (Projection/Joint Normal Form), 5NF, and so on.

d. First normal form (1NF)

When we transform entities or relationships into tabular relations, there may be some relations in which there are more values in the intersection of a row or column.

e. Second normal form (2NF)

In each relation we need to have a key (called a primary key) on which all other attributes (column values) needs to depend.

CHƯƠNG 16 :SECURITY (BẢO MẬT) xong

1.Security goals

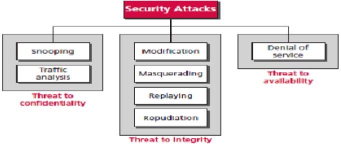
-**Confidentiality (bảo mật)** is probably the most common aspect of information security.An organization needs to guard against those malicious actions that endanger the confidentiality of its information.

-Integrity (tính toàn vẹn) Information needs to be changed constantly. In a bank, when a customer deposits or withdraws money, the balance of their account needs to be changed. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms.
(có nghĩa là dữ liệu phải đến người nhận chính xác như khi chúng được gửi đi)

-Availability (tính khả dụng) The third component of information security is availability. The information created and stored by an organization needs to be available to authorized entities. Information is useless if it is not available. Information needs to be constantly changed, which means it must be accessible to authorized entities.

2. Attacks

Our three goals of security—confidentiality, integrity, and availability—can be threatened by security attacks. Although the literature uses different approaches to categorizing the attacks, we divide them into three groups related to the security goals.



|

3. Services and techniques (dịch vụ và kỹ thuật)

-ITU-T defines some security services to achieve security goals and prevent attacks. Each of these services is designed to prevent one or more attacks while maintaining security goals. Two techniques are below(có 2 kỹ thuật)

Cryptography (general)

Although in the past cryptography referred only to the encryption and decryption of messages using secret keys.(Mặc dù trước đây mật mã chỉ đề cập đến việc mã hóa và giải mã tin nhắn bằng khóa bí mật.)

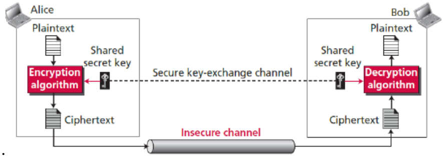
Steganography (specific)

The word steganography, with origins in Greek, means 'covered writing', in contrast to cryptography, which means 'secret writing'..
(Từ steganography, có nguồn gốc từ tiếng Hy Lạp, có nghĩa là 'văn bản được che đậy', trái ngược với mật mã, có nghĩa là 'văn bản bí mật'..)

1- CONFIDENTIALITY (bảo mật)

a. Symmetric-key ciphers (mật mã khóa đối xứng)

A symmetric-key cipher uses the same key for both encryption and decryption (mã hóa và giải mã), and the key can be used for bidirectional communication.

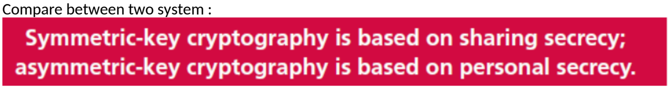


.

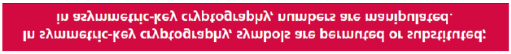
(Mật mã khóa đối xứng sử dụng cùng một khóa cho cả mã hóa và giải mã và khóa này có thể được sử dụng để liên lạc hai chiều)

b.Asymmetric-key ciphers (mật mã khóa bất đối xứng)

Symmetric- and asymmetric-key ciphers will exist in parallel and continue to serve the community.
In symmetric-key cryptography, the secret must be shared between two persons. In asymmetric-key cryptography, the secret is personal (unshared); each person creates and keeps his or her own secret.
Compare between two system :

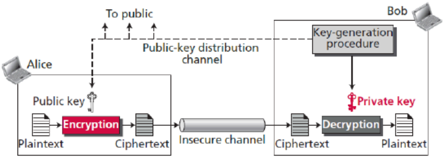

Symmetric-key cryptography is based on sharing secrecy;
asymmetric-key cryptography is based on personal secrecy.

And


In symmetric-key cryptography, symbols are permuted or substituted;
in asymmetric-key cryptography, numbers are manipulated.

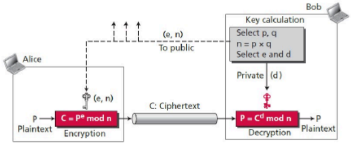
3. General idea

Figure 12.3 shows the general idea of **asymmetric-key cryptography** as used for **encipherment.**



R4. SA cryptosystem

-Although there are several asymmetric-key cryptosystems, one of the common public-key algorithms is the ***RSA*** cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).
-RSA uses two exponents, e and d, where e is public and d is private

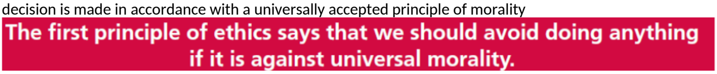


2 - ETHICAL PRINCIPLES

Có 3 nguyên tắc chính :

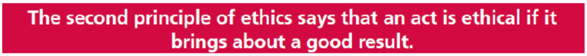
a.Moral rule

The first ethical principle states that when we make an ethical decision, we need to consider if the decision is made in accordance with a universally accepted principle of morality


The first principle of ethics says that we should avoid doing anything if it is against universal morality.

b.Utilization

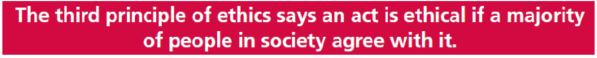
The second theory of ethics is related to the consequences of the act. An act is ethical if it results in consequences which are useful for society.



The second principle of ethics says that an act is ethical if it brings about a good result.

c. Social contract

The social contract theory says that an act is ethical when a majority of people in society agrees with it. If someone breaks into somebody else’s house and commits a robbery, does this act receive the approval of a majority of society? Since the answer is negative, this act is not ethical.

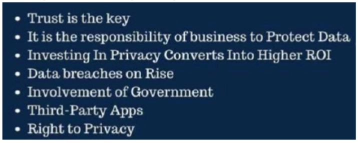


The third principle of ethics says an act is ethical if a majority of people in society agree with it.

3.PRIVACY

Codes of ethics related to the use of computers to collect data, as shown below:
1. Collect only data that are needed.
2. Be sure that the collected data are accurate.
3. Allow individuals to know what data have been collected.
4. Allow individuals to correct the collected data if necessary.
5. Be sure that collected data are used only for the original purpose.
6. Use encryption techniques

a. Why Data Privacy is important?4. Non-Disclosure Agreement



- Trust is the key
- It is the responsibility of business to Protect Data
- Investing In Privacy Converts Into Higher ROI
- Data breaches on Rise
- Involvement of Government
- Third-Party Apps
- Right to Privacy

b. Non-Disclosure Agreement

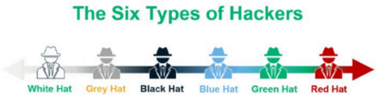
-A NDA can also be known by other names such as a confidentiality, non-use or trade secret agreement.

-Essentially, a non-disclosure (NDA) agreement is a legally binding contract between parties that requires them to keep certain information confidential.

4 - HACKERS

-The word hacker today has a different meaning than when it was used in the past. Previously, a hacker was a person with a lot of knowledge who could improve a system and increase its capability.

-Today, a hacker is someone who gains unauthorized access to a computer belonging to someone else in order to copy secret information.



-**Black Hat Hacker** Basically, these are the “bad guys”. They are the types of hackers who break into computer networks with purely negative motives such as monetary gain or reputation.

-White Hat Hacker As opposed to the black hat, these are the “good guys”. They are ethical hackers who create algorithms to break existing internet networks so as to solve the loopholes in them.

-Grey Hat Hacker Basically, these are hackers who exploit the internet systems only to make public, certain vast datasets of information that would be of benefit to everyone.

-Blue Hat Hacker In one word, this is the amateur. Usually, their techniques are deployed out of ill motives such as revenge attacks.

Red Hat Hacker The objective of a red hat hacker is to find black hat hackers, intercept and destroy their schemes.

Green Hat Hacker This is the set of individuals who simply want to observe and learn about the world of hacking. It comprises those who join learning communities to watch videos and tutorials about hacking.

Common types of hacking

-**Hacking for financial gain** Lone black hat hackers as well as hacking collectives are typically thieves. Their cybercrimes are targeted at either directly stealing money, enabling later theft via data hijacking, or selling the acquired data to other cybercriminals.
-Corporate espionage With so many industries as cutthroat as they are, it’s unsurprising that companies are often willing to get dirty to triumph over the competition. Corporate (or industrial) espionage is the commercial application of hacking, malware, phishing, and other unsavory spying techniques to obtain privileged insider information from a business competitor — aka information hacking.
-State-sponsored hacking The potential rewards from security hacking can be so great, even governments want to get in on the party. Countries all across the world are constantly playing games of cat-and-mouse cyber warfare with one another. Everyone knows that everyone else is doing it, and everyone acts surprised and offended when they get caught.