

University of AL-Mustansiriyah
College Of Pharmacy



Chapter 4

Computer Software

Section II

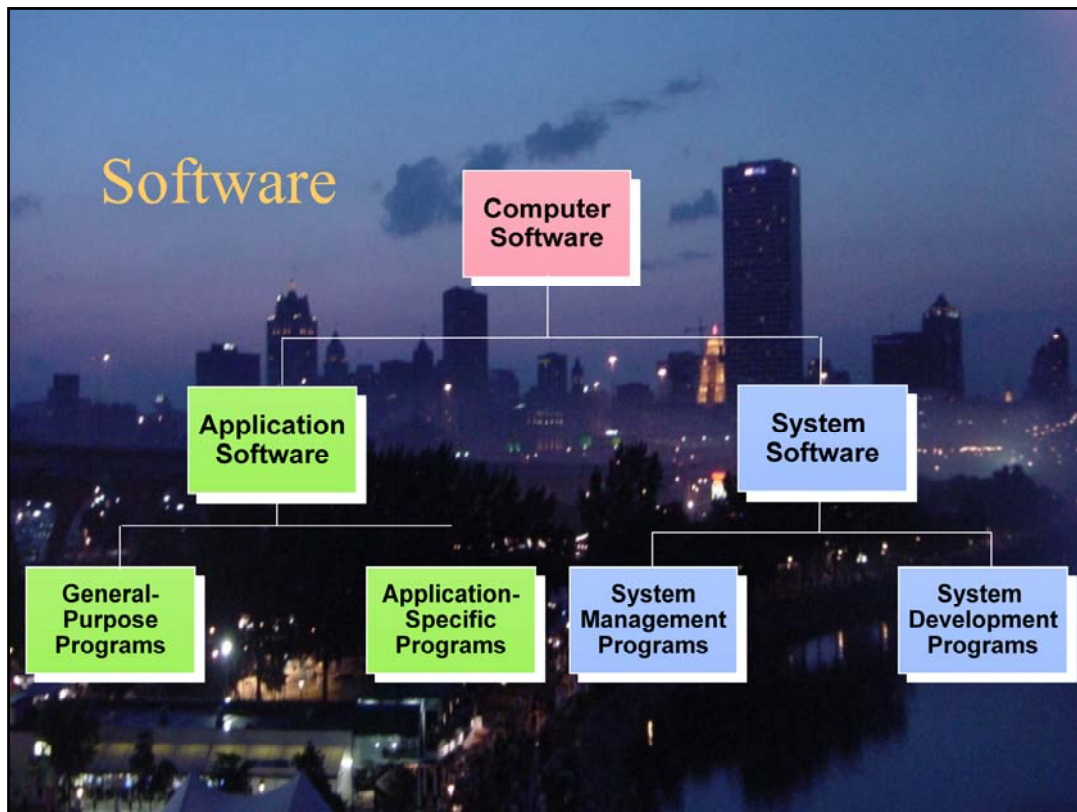
System Software: Computer System Management

Present by

L.A. Mr. Hussein Salim Qasim

Section II

- System Software: Computer System Management

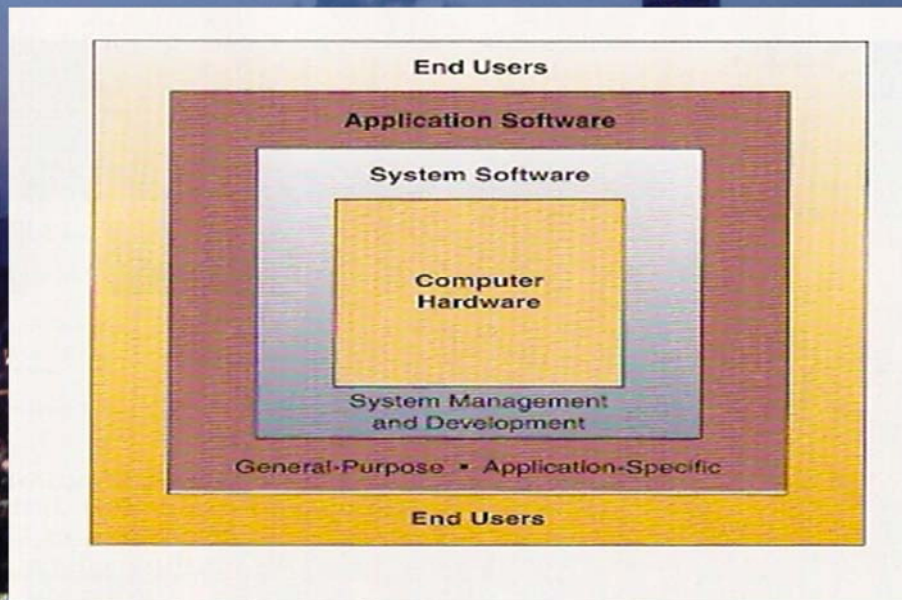


Application software & system software

System Software Overview

- Programs that manage and support a computer system and its information processing activities
- Serves as the software interface between computer networks and hardware and the application programs of end users

Operating System Overview



A nighttime photograph of a city skyline, likely Chicago, with a prominent skyscraper (the Willis Tower) and other buildings reflected in a body of water. The sky is dark blue, and the city lights are visible.

System Software categories

- **Two major categories**

- 1. System management programs**

Programs that manage hardware, Software, network and data resources of computer systems during the execution.

System software Examples:

- Operating systems
- Network management programs
- Database management systems
- System utilities



System Software categories

2. System development programs

Programs that help users to develop system programs

Examples:

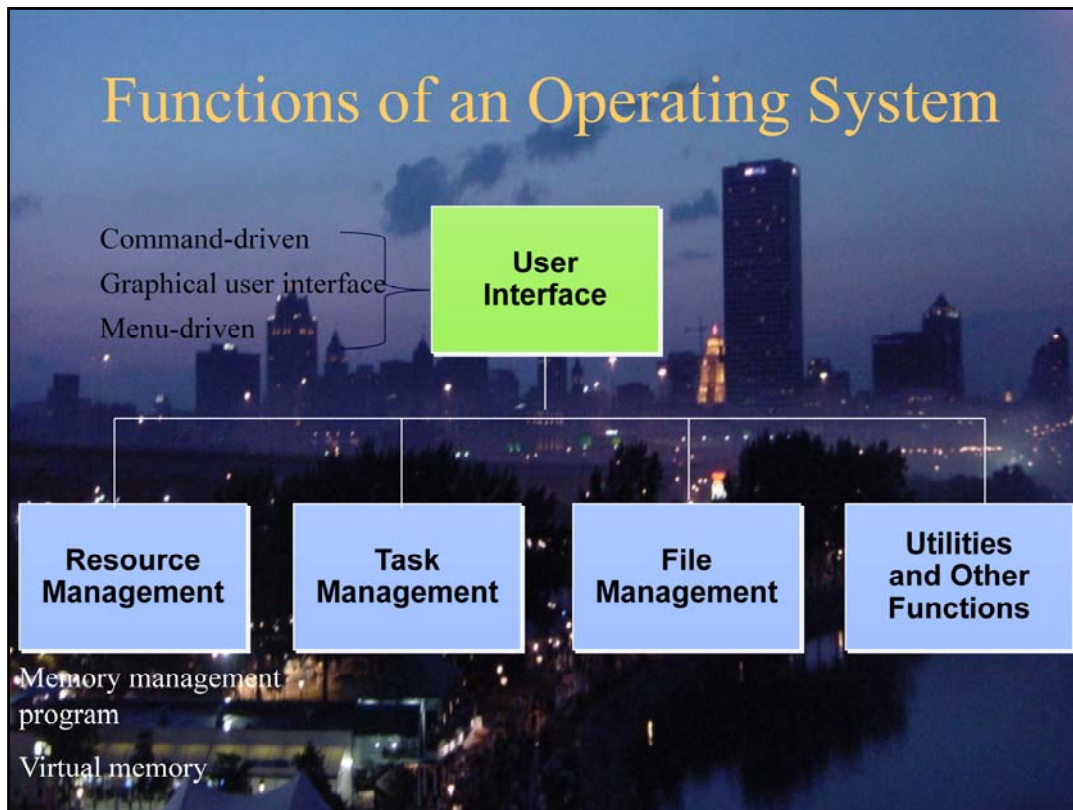
- Programming language translators & editors and other programming tools.

Operating Systems

- Most important system software package for any computer
- Is an integrated system of programs that (**OS Activity**):
 - Manages the operations of the CPU
 - Controls the input/output and storage resources and activities of the computer system
 - Provides various support services as the computer executes application programs

Operating System Objective:

- To maximize the productivity of a computer system by operating it in the most efficient manner.
- To help application programs perform operation such as accessing a network , enter data, saving and retrieving data.



The operating system of a computer manages the operations of the CPU, controls the input/output and storage resources and activities of the computer system, and provides various support services as the computer executes the application programs of users.

- A. The user interface is the part of the operating system that allows users to communicate with the computer. Interfaces types:
1. command-driven (the user types in command instructions through the keyboard, MS-DOS, NC)
 2. menu-driven (the user selects commands presented on-screen with either a mouse or keyboard. Early windows)
 3. Graphical user interface (the user selects commands that appear as icons, buttons, bars, and other images with a mouse or other pointing device, windows , Linux , Mac). Through the interface, the end user has access to the following resources:

Resource Management. These programs manage the hardware of a computer system, including its CPU, memory, secondary storage devices, and input/output peripherals.

File Management. Operating systems also contain file management programs that control the creation, deletion, and access to files of data and programs. File management involves keeping track of the physical location of files on magnetic disks and other secondary storage devices.

Task Management. Task management programs manage the computing tasks of end users. They give slices of the CPU's processing time to tasks and determine the logistics of which tasks take priority over others when more processing time is required.

Instructor's Note: The following is extra-textual material on this topic:

Utilities and Other Functions. Operating systems may contain additional programs (or support for them to be added later) called utilities that help maintain the integrity of the system and its interface with the system hardware. Common utilities are used to defrag a system's hard drive, compress the space programs and files need for storage, or other functions.

Discussion topic: Discuss the advantages and disadvantages of popular operating systems, e.g. Windows 2000, UNIX, LINUX, Netware, etc.

Functions of an Operating System

- **Performs five basic functions**

1. User Interface

Allows humans to communicate with the computer:

- Command-driven
- Menu-driven
- Graphical user interface

Functions of an Operating System

2. Resource management

- Manages the hardware and networking resources of the system
- Virtual memory capability

3. File management

- Controls the creation, deletion, and access of files of data and programs
- Keeps track of the physical location of files

Functions of an Operating System

4. Task management

- Manages the accomplishment of the computing tasks of end users
- Multitasking
 - Multiprogramming
 - Timesharing

Functions of an Operating System

5. Utility Programs

- Perform miscellaneous housekeeping and file conversion functions
 - Data backup
 - Data recovery
 - Virus protection
 - Data compression
 - Data defragmentation
- Performance monitors and security monitors



Operating Systems

- Popular Operating Systems
 - Windows (Microsoft)
 - 95, 98, ME
 - NT
 - 2000
 - XP
 - UNIX
 - Linux
 - Mac OS X

Categories of Programming Languages

<u>Machine Languages</u> Use binary coded instructions 1001 1001 1100 1101	<u>High Level Languages</u> Use brief statements Compute $X = Y + Z$	<u>Markup Languages</u> Use embedded control codes <H1>First heading</H> <!ELEMENT Product (#Item manuf)>
<u>Assembler Languages</u> Use symbolic coded instructions LOD Y ADD Z	<u>Fourth Generation Languages</u> Use natural statements SUM THE FOLLOWING NUMBERS	<u>Object-Oriented Languages</u> Define objects that contain data and actions Document.write ("Hi There")

A programming language allows a programmer or end user to develop the set of instructions that constitute a computer program. Each programming language has its own unique vocabulary, grammar, and uses. The major categories of programming languages are identified on the slide and described below:

Machine Languages. Machine languages (or first-generation languages) are the most basic level of programming languages. These use binary codes unique to the computer, requiring programmers to have a detailed knowledge of the specific CPU they wish to write software for.

Assembler Languages. Assembler languages (or second-generation languages) reduce the difficulties in writing language code by using translator programs (assemblers) that convert the symbolic language of the code into machine language.

High-Level Languages. High-level languages (or third-generation languages) use instructions, called statements that closely resemble human language or the standard notation of mathematics. They are translated into machine language by compilers or interpreters.

Fourth-Generation Languages. Fourth-generation languages describe a variety of programming languages that are more nonprocedural and conversational than previous languages. Nonprocedural languages have programmers specify the results they want while the program works with the computer to determine the sequence of instructions that will accomplish those results.

Object-Oriented Languages. Object-oriented programming (OOP) ties data and instructions together into objects that can be combined in many different ways with other objects to create programs. Unlike procedural languages, OO systems objects tell other objects to perform actions on themselves. Thus, objects are more efficient and can be reused to create new programs. *Java* is an example of an OOP, which is specifically designed for real-time, interactive, web-based network applications. What makes Java so special is that it is computing platform independent. This means that any computer and any operating system anywhere in a network can execute Java programs.

HTML. Is a page description language that creates hypertext or hypermedia web documents. HTML embeds control codes, or tags, in the ASCII text of a document. These tags are used to designate titles, headings, graphics, multimedia components, as well as hyperlinks within the document.

XML. Unlike HTML, XML describes the content of web pages by applying identifying tags or contextual labels to the data in web documents. By classifying data in this way, an XML website's information is more searchable, sortable, and easier to analyze.

A nighttime photograph of a city skyline, likely Pittsburgh, with a river in the foreground. The city lights are reflected in the water, and the sky is dark blue. The title 'Programming Languages' is overlaid in a yellow, serif font.

Programming Languages

- Assembly/machine language
- FORTRAN, COBOL, Pascal, Basic
- C/C++ (the code behind Unix & Linux)
- Visual Basic
- SQL (for database development)
- HTML, XML (web pages)
- Java Script (adds programming power to web pages)
- 4GLs – A.I. & non-procedural/natural languages

A nighttime photograph of a city skyline, likely Chicago, with the Willis Tower prominently visible. The city lights are reflected in a body of water in the foreground. The title 'Programming Languages' is overlaid in blue text at the top.

Programming Languages

- Allows a programmer to develop the sets of instructions that constitute a computer program
- Machine Language
 - First generation language
 - Written using binary codes unique to each computer

Programming Languages (continued)

- Assembler Language
 - Second generation
 - Requires language translator programs called assemblers
 - Allows a computer to convert the instructions into machine instructions
 - Frequently called symbolic language

A nighttime photograph of a city skyline, likely New York City, with a river in the foreground. The city lights are reflected in the water. The title 'Programming Languages (continued)' is overlaid in blue text at the top.

Programming Languages (continued)

- High-level Languages
 - Third generation
 - Uses instructions, called statements, that use brief statements or arithmetic expressions
 - Uses translator programs called compilers or interpreters
 - Syntax and semantics

A nighttime photograph of a city skyline, likely New York City, with the Empire State Building prominently visible. The city lights are reflected in a body of water in the foreground.

Programming Languages (continued)

- Fourth-generation Languages (4GLs)
 - More nonprocedural and conversational than prior languages
 - Natural languages
 - Ease of use gained at the expense of some loss in flexibility

A nighttime photograph of a city skyline, likely Chicago, with the Willis Tower prominently visible. The city lights are reflected in a body of water in the foreground.

Programming Languages (continued)

- Object-Oriented Languages (OOP)
 - Ties data elements to the procedures or actions that will be performed on them into “objects”
 - Easier to use and more efficient for programming GUIs

A nighttime photograph of a city skyline, likely New York City, with a river in the foreground. The city lights are reflected in the water. The title 'Programming Languages (continued)' is overlaid in blue text at the top.

Programming Languages (continued)

- Markup language
 - HTML, XML, and Java
 - Important for building multimedia Web pages, websites, and Web-based applications
 - HTML (Hypertext Markup Language)
 - A page description language that creates hypertext or hypermedia documents

A nighttime photograph of a city skyline, likely Chicago, with the Willis Tower prominently visible. The city lights are reflected in a body of water in the foreground. The text is overlaid on the upper half of the image.

Programming Languages (continued)

- XML (extensible Markup Language)
 - Describes the contents of web pages by applying identifying tags or contextual labels to the data
 - Makes the web site more searchable, sort able, and easier to analyze
- Java
 - Designed for real-time, interactive, Web-based network applications
 - Applets

Programming Software

- Helps programmers develop computer programs
- Two basic categories
 - Programming language translators
 - Programming tools



Programming Software (continued)

- Language Translator Programs
 - Assembler
 - Translates symbolic instruction codes into machine language instructions
 - Compiler
 - Translates high-level language statements
 - Interpreter
 - Translates and executes each statement in a program one at a time

A nighttime photograph of a city skyline, likely Pittsburgh, with a river in the foreground. The city lights are reflected in the water. The title 'Programming Software (continued)' is overlaid in blue text at the top.

Programming Software (continued)

- Programming Tools
 - Programming editors and debuggers
 - Provides a computer-aided programming environment or workbench
 - Diagramming packages
 - Code generators
 - Libraries of reusable objects & code
 - Prototyping tools
 - CASE



Chapter Summary

- Computer software consists of two major types of programs: application software that directs the performance of a particular end user task, and system software that controls and supports the operations of a computer system.
- Application software includes a variety of programs that can be segregated into general-purpose and application-specific categories.

Chapter Summary (cont)

- System software can be subdivided into system management programs and system development programs. The former is used to manage hardware, software, networks, and data resources. The latter is used by IS specialists to develop computer programs.
- An operating system is an integrated system of programs that supervises the operations of the CPU.



Chapter Summary (cont)

- There are 5 major levels of programming languages. Language translator programs convert programming language instructions into machine language instructions.