Software

Purpose - In order for the CPU to control the physical states of the digital circuitry, it must receive signals from a computer program and convert these signals into action (switching circuits on and off).

Software

- Programming Language
- Operating System
- General purpose Application Software
- Application specific Software

Program Language

- a set of symbols and rules used to write code that can be interpreted by a computer.
- Syntax: the rules of the language.

Programs written for the ENIAC

- Determine trajectory of artillery shells
- Scientists had to set thousands of switches
EDVAC – Electronic Discrete Variable Automatic Computer

- Designed by Maulchy and Eckerd
- Used binary numbers (changed programming)
- Could internally store programs
- Made computers more versatile
- Lines of code could be modified and/or borrowed

Alan Turing

- Manchester Mark I Computer
- Father of Computer Science
- Created a code that would represent a series of 0's and 1's
- Letters typed into teleprinter, which would produce a series of holes representing binary code on paper tape.
- Problem – different scientists, different codes

Problems with Turin code

- Not easily understood
- Few people had access to his notes
- Impossible to determine which two letter code related to which strings of binary code.
- In addition, different two letter codes were different binary strings for different machines.

Need:

- A formal programming language
- Across computers
- To simplify
- And to advance programming

Grace Hopper

- Developed ideas for the first High Level language
- A-0 system was the first compiler ever created and led to the development of FLOW-MATIC which led to the development of COBOL

Compiler

- She noticed that programmers were copying parts of programs into other programs to transform source code (high level language) into object code (eg. binary language)
- The compiler “compiles” subroutines into binary language
Program Language Levels

• 1st generation - Machine Language:

Program Language Levels (cont.)

• 2nd generation - Assembly Language:

Program Language Levels (cont.)

• 3rd generation - High-level:

Program Language Levels (cont.)

• 4th generation - Natural Language:

Evolution of Programming Languages

Object-Oriented Languages

• Combine data elements and the procedures that will be performed upon them into OBJECTS
Object-Oriented Languages

Reusable: can use an object from one application in another application

Savings Account Object

Web Languages

- HTML
- XML
- Java

Role of System Software

- to control the hardware and act as an interface between application software and hardware

Operating Systems

- Scientists had to schedule time on computers to run programs (flipping switches, using punch cards, or punch tape)
- Later, computers were built with libraries of software and scientist could link programs to those libraries.
- Later systems software allowed spooling (placing data in a temporary working area for another program to process) and multitasking

Example of Application Software, OS, and Hardware

Application Software

- What is the price of Gizmo$?

Operating System Software

- Fetch Gizmo$ from hard disk, track 1, sector 7
Operating Systems for PCs

• PC-DOS and MS-DOS.
• OS/2 - Operating System 2.
• Windows
• Mac OS X
• Unix
• Linux

Virtual Memory

• An operating system can divide memory into a number of sections and swap parts of programs and data between primary memory and secondary memory in order to process a larger program or a greater amount of data than primary memory would ordinarily allow.

Application Software

• Programs that direct computers to perform information processing for end users

General Purpose Application Software

• perform common information processing jobs for end users

Groupware

Application-Specific Programs

• Application-Specific Programs - support specific applications of end users in business, science and other areas
End of Software Chapter