COMP1200-001

Introduction to Computing for Engineers and Scientists

C Programming

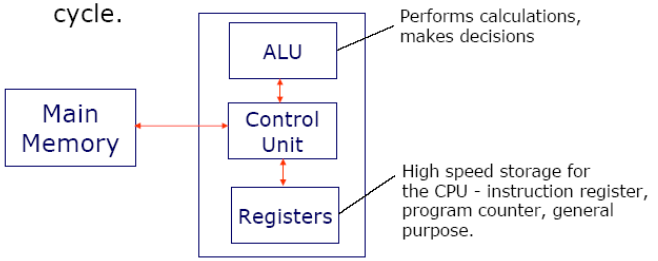
Course Notes Chapter 1:
Overview of Computers and Software

Overview of Computers and Software

- Computer Components (Hardware)
  - Major components of a computer system
  - How component work together to solve problems and manipulate data
- Computer Software
  - Major categories of software
  - Kinds of languages in which they are implemented
  - Writing, compiling, and executing high-level language programs
- Computing for Engineers and Scientists
- The Software Development Method

Computer Components

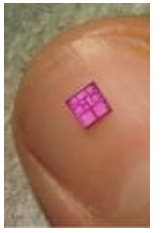
- The CPU interprets and executes instructions in a continuous "**fetch-decode-execute**" cycle.

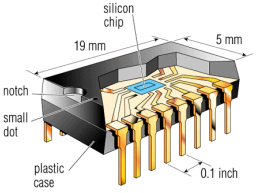


- All the activities of the CPU are synchronized and regulated by the ***system clock***.

Central Processing Unit (CPU)

- Control unit (CU)
  - Processor, or microprocessor
- Arithmetic/Logic unit (ALU)
- Housed in an Integrated circuit (IC), or chip



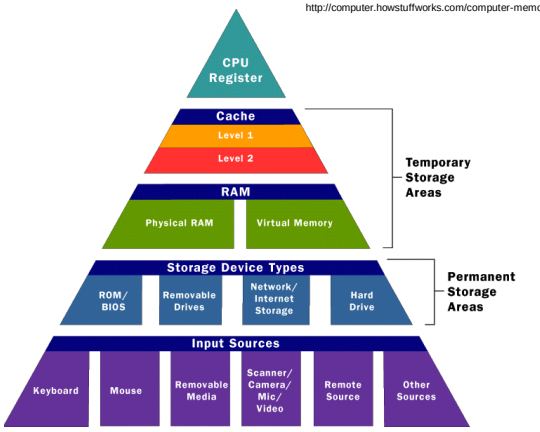


Central Processing Unit (CPU)

- Control unit (CU)
  - Control activities of CPU
  - Copies data and instructions from memory to registers
- Arithmetic-logic unit (ALU)
  - Contains circuitry for data manipulation
  - Arithmetic operations: + - * /
  - Logical operations: > < = not= AND OR SHIFT ROTATE
- Registers
  - Memory cells in CPU
  - Allows rapid access of information by CU and ALU

Memory

http://computer.howstuffworks.com/computer-memory1.htm



Main Memory

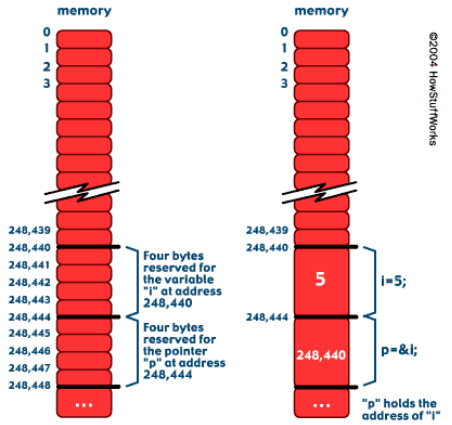
Where information that is to be processed and instructions used to process the information are stored

- Information is stored in bits (binary digits)
  - Two states - 1 or 0...ON or OFF
- Bits are grouped into bytes (8 bits) and words (4 bytes)
- Byte is smallest piece of information that can be addressed
  - All letters, special characters, and instructions are represented by a type code and stored as a bit pattern
- Size
  - megabytes (MB) - mega ~ 1 million ~ 2^20 = 1,048,576
  - gigabytes (GB) - giga ~ 1 billion ~ 2^30 = 1,073,741,824

Main Memory cont.

- Random access memory (RAM)
  - Allows access to the bytes in no particular order
- Read-only memory (ROM)
  - Information is permanent in memory
  - Cannot be modified by user
  - Set during manufacturing process
  - Contains instructions and information fundamental to the computer's performance

Memory



Characters and Symbols

- Digital representation by 1s and 0s
- ASCII code –
  - each character and symbol represented by 8 bits, or 1 byte

Char	Binary	Hex	Decimal
A	0100 0001	41	65
B	0100 0010	42	66
a	0110 0001	61	97
b	0110 0010	62	98
?	0011 1111	3F	63
[space]	0010 0000	20	32
NUL	0000 0000	00	00
9	0011 1001	39	57
0	0011 0000	30	48

Hardware Communicates w/ CPU

Peripheral devices
- Mass storage (secondary storage)
- Input-output (I/O)

- Mass storage (Secondary storage)
  - Store information
    - Tapes, hard disks, floppy disks, CDs
    - Drives: tape, hard disk, floppy disk, CD-ROM
  - Disk or tape - "on-off" markings are magnetized or not magnetized
  - CDs - "on-off" markings are pit or smooth

Hardware Communicates w/ CPU

Mass Storage cont.
- Difference between mass storage and main memory
  - Mass storage is
    - Slower
      - Moved into main memory to be processed
      - Required mechanical motion
    - Portable
    - Greater capacity
    - Remains even when power is off…Non-volatile!

access speed and cost increases ↑ registers, main memory, mass storage ↓ space increases

Hardware Communicates w/ CPU

Input-output (I/O) devices

- Input
  - Keyboard, mouse
  - Microwave oven temperature probe
  - Bar code scanner, card swipe, touch pad, microphone
- Output
  - Monitor, printer, speaker

- **Peripheral device controller**
  - Miniature computer
  - Coordinates the actions of peripheral device with activates of computer

  ex.
  - CPU send information to the SLOWer printer
  - Controller takes over relieving the very FAST CPU
  - Very FAST CPU can continue processing
- **Device driver**
  - Software program compatible with the OS

Connecting Computers

- Wide are networks (WANs)
  - Large geographical area

- Local area networks (LANs)
  - University or company
  - Your home
  - ex. Multiple computers sharing one printer

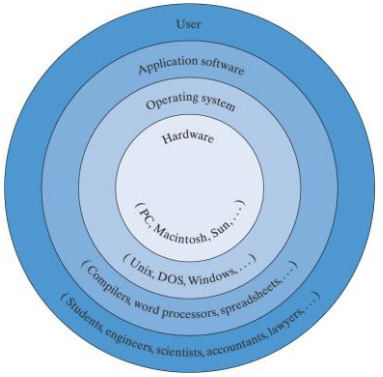
Software

- Hardware is the tangible part of a computer
- Software is a set of instructions
  - Two types
    - System software
    - Application software

System Software

- Operating system (OS)
  - Written into memory at the startup of the computer
  - Handles communication between all equipment
  - "Look and feel" of the computer
  - Layer between the user and the details in dealing with the devices
  - ex.
    - UNIX
    - Microsoft windows
    - Macintosh operating system
  - MS-DOS
- Utility programs - handle files

Software Interface to the Computer



Language translators

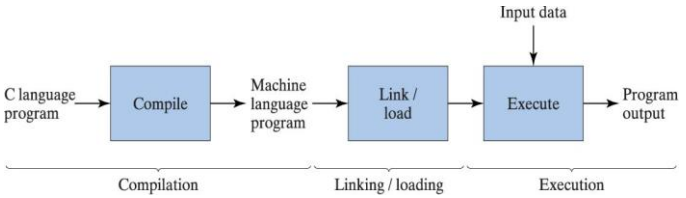
- Create machine language instructions
- 3 types: assemblers, compliers, interpreters

Assembler
- Convert assembly to object code
- Less complicated than complier and interpreter

Compiler
- Converts high level language to machine language
- Errors are detected
- Looks for violations of language or syntax rules
- Cannot detect logic errors
- Links modules that are needed

Interpreter
- Used in high level languages
- Converts an instruction to machine language and execute it; then goes to next instruction

Program Compilation/Linking/Execution



Application Software

- Programs with which you are familiar
- Word processing, games, spreadsheets, internet browser, etc.

Programming Languages

- Machine language
  - 1s and 0s
  - Only language a computer understands
- Assembly language
  - One level above machine language
  - Included instructions for moving information to and from registers
- High level languages
  - Created to simplify the commands written by a human
  - Syntax rules must be followed
- C is a procedural language
  - Layout the procedure, or steps, to solve a problem
  - Write a program to carry out the steps
- 4th generation language
  - programming environment designed with a specific purpose
  - MATLAB

C:

```
for (i=10; i > 0; i--)
{
   //loop1 body
}
```

Assembly:

```
      SET r1, 10          ;r1 will take the place of i
      SET r2, 1           ;r2 will hold the value to
                          ;subtract each time
LOOP1TOP:
      ...                 ;loop1 body
      SUB r1, r1, r2      ;subtract one from r1
      CMP r1, r0          ;compare r1 with 0
      JMP NEQ, LOOP1TOP   ;keep going until r1=0
```


Comparison of Software Statements

Software	Example Statement
C++	`area = 3.141593*(diameter/2)*(diameter/2);`
C	`area = 3.141593*(diameter/2)*(diameter/2);`
MATLAB	`area = pi*((diameter/2)^2);`
Fortran	`area = 3.141593*(diameter/2.0)**2`
Ada	`area := 3.141593*(diameter/2)**2;`
Pascal	`area := 3.141593*(diameter/2)*(diameter/2)`
Basic	`let a = 3.141593*(d/2)*(d/2)`
COBOL	`compute area = 3.141593*(diameter/2)* (diameter/2).`

SoftwareEngineering Life-Cycle Phases

Life Cycle	Percent of Effort
Definition	3
Specification	15
Coding and modular testing	14
Integrated testing	8
Maintenance	60

Software Engineering

Describes the process of software development

- Define the function of the program
- Sketch out a design
  - Pseudo code
- Discuss with all parties
- Modify
- Repeat
- After the design is agreed upon,
  - Write the real program
  - Test
  - Modify
  - Repeat

The Software Development Method

1. Specify the problem requirements
2. Analyze the problem
3. Design the algorithm to solve the problem
4. Implement the algorithm
5. Test and verify the completed program
6. Maintain and update the program

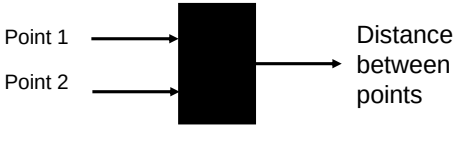
Applying the Software Development Method

State the problem

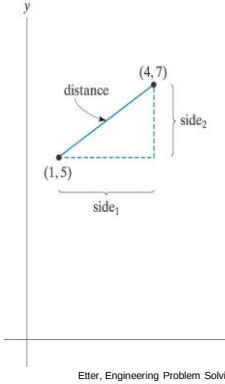
- Compute the straight-line between two points in a plane.

Analysis

- Input: point_1 and point_2
- Output: distance between points



Straight-line Distance Between Two Points



Hand example:

P1 = (1,5)
P2 = (4,7)

Distance = sqrt($(side1)^2 + (side2)^2$)
= sqrt($(4 - 1)^2 + (7- 5)^2$)
= sqrt(13)
= 3.61

Design

The Algorithm

Step by step outline of the problem solution

Decompose the problem into simpler steps

Develop the Algorithm

1. Give values of two points
2. Compute the lengths of the two side of the right triangle generated by the two points
3. Compute distance between points, the hypotenuse
4. Print distance between two points

Implementation - C

```
// p1_1.c : Defines the entry point for the console application.
/* Program chap1_1 - Computes distance between 2 pts. */
#include <stdio.h>
#include <math.h>
int main(void)
{
   double x1=1, y1=5, x2=4, y2=7
          side1, side2, distance;
   side_1 = x2 - x1;
   side_2 = y2 - y1;

   distance = sqrt(side_1*side_1 + side_2*side_2);
   printf("The distance between the two
          points is %5.2f \n",distance);
   return 0;
}
```


Implementation - MATLAB

```
% calDist.m : Defines the entry point for the console application.
% Computes distance between 2 pts.
x1=1;
y1=5;
x2=4;
y2=7;

side_1 = x2 - x1;
side_2 = y2 - y1;

distance = sqrt(side_1^2 + side_2^2);
fprintf('The distance between the two
         points is %5.2f \n',distance);
```


The Program

Testing

- Use the numbers from your hand example
- Do you get the same result?

```
The distance between the two points is  3.61.
```

Maintenance

- Enhancements
- Fixing errors
- Adapting to new software and hardware