Programming and Data Structure

Sujoy Ghose

Sudeshna Sarkar

Jayanta Mukhopadhyay

Dept. of Computer Science & Engineering.

Indian Institute of Technology

Kharagpur

Some General Announcements

About the Course

- L-T-P rating of 3-1-0.
- There is a separate laboratory of 0-0-3.
  - Grading will be separate.
- Tutorial classes (one hour per week) will be conducted on a “per section” basis.
- Evaluation in the theory course:

– Mid-semester	30%
– End-semester	50%
– Two class tests and attendance	20%

Course Materials

- Some general resources for the course are available on the web.

  http://cse.iitkgp.ac.in/~pds

- All important announcements will be put up on the web page.
- Specific resources for this class will be made available from http://cse.iitkgp.ac.in/~pds/current/SS

ATTENDANCE IN THE CLASSES IS MANDATORY

Students must have at least 75% attendance

Text/Reference Books

1. Programming with C

   B.S. Gottfried, Schaum's Outline Series, Tata McGraw-Hill, 2006.

2. The C Programming language. Kernighan and Richie, Pearson Education
3. R. G. Dromey, How to Solve It by Computer, Pearson Education

Tentative Schedule

1	Introduction, data types, variables and constants, operators, expression, assignment, basic I/O, flowcharts, programming examples	5 hrs
2	Control statements, programming examples and exercises	3 hrs
Class Test 1, Feb 1, 2012		
3	Functions, recursion, programming examples and exercises	4 hours
4	Number systems	2 hrs
5	Arrays, using arrays with functions, programming examples and exercises	2 hrs
6	Pointers, relationship with arrays	1 hr
Midterm, Feb 17 to 24		

Tentative Schedule

<table>
<tr><td>7</td><td>Strings, 2d arrays, using arrays with functions, programming examples and exercises</td><td>3 hrs</td></tr>
<tr><td>8</td><td>Structures, array of structures, examples</td><td>3 hrs</td></tr>
<tr><td>9</td><td>Pointers, relationship with arrays, parameter passing, dynamic memory allocation</td><td>4 hrs</td></tr>
<tr><td colspan="3"><b>Class test 2, March 21, 2012</b></td></tr>
<tr><td>10</td><td>Files, command-line parameters</td><td>1 hr</td></tr>
<tr><td>11</td><td>Linked list, abstract data types, implementation of stacks and queues</td><td>4 hrs</td></tr>
<tr><td>12</td><td>Sorting algorithms: selection, bubble and quick, time and space complexity. Searching: linear, binary.</td><td>4 hrs</td></tr>
<tr><td>13</td><td>Algorithm analysis, formal notations.</td><td>2 hrs</td></tr>
<tr><td colspan="3"><b>Endterm, April 20-27, 2012</b></td></tr>
</table>

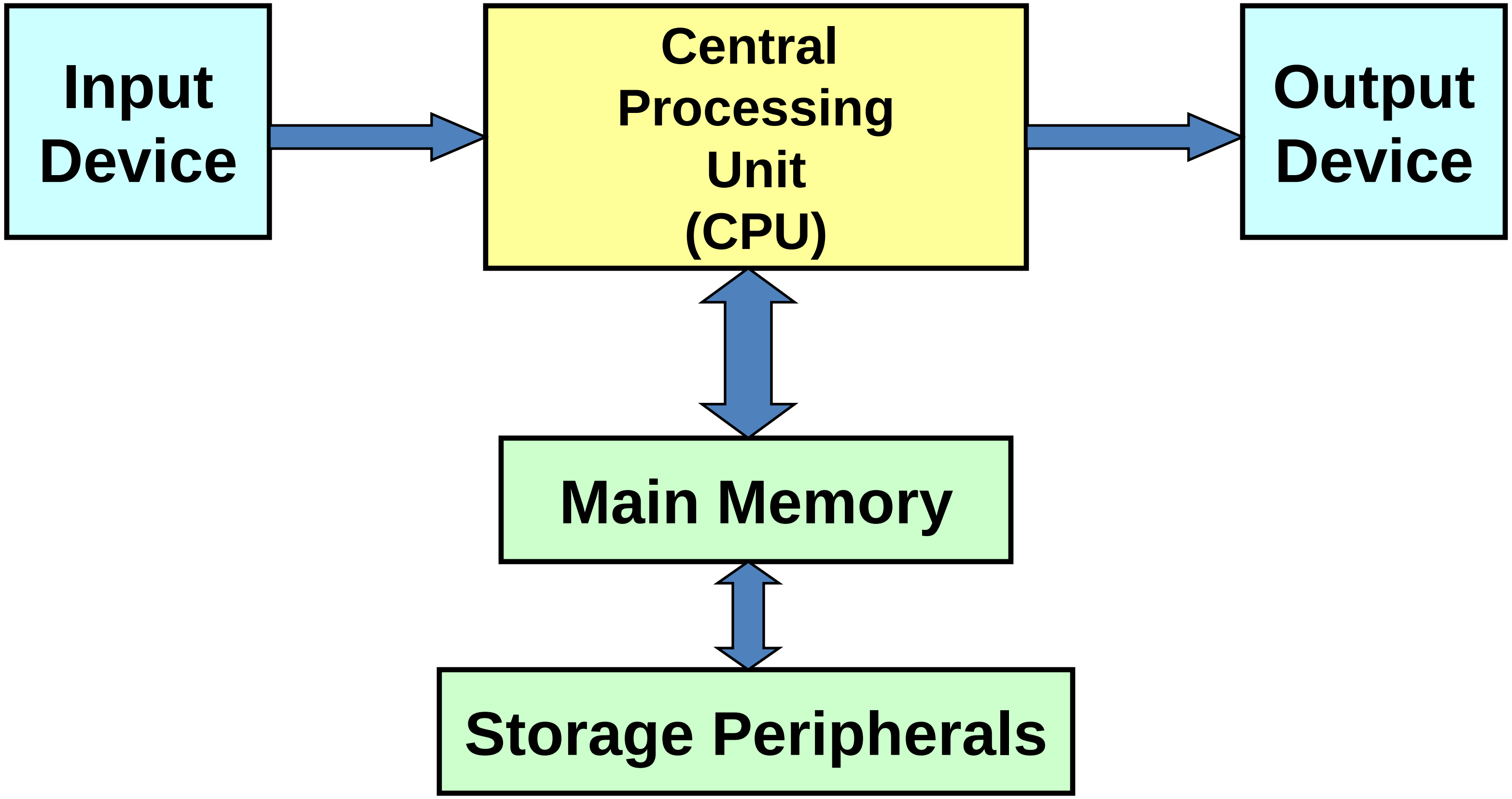
Introduction

What is a computer

- Machine which can receive data from the external world, store it, process it, and send result to the external world.
- Data: numbers, images, text, sounds, ...
- Process:
  - Compute the average of a sequence of numbers,
  - Invert a matrix
  - Identify a face
  - Speech recognition

What is a Computer?

It is a machine which can accept data, process them, and output results.



Parts of a computer

- MEMORY: cells which store data. Each cell stores 1 number.
- CPU
  - ARITHMETIC UNIT: Takes two numbers as input and produces their sum/... as an output.
  - CONTROL UNIT: Decides what the other units are supposed to do.

Parts of a computer (contd)

- INPUT DEVICE: reads information from the external world into memory. e.g. keyboard
- OUTPUT DEVICE: sends data from memory to the external world, e.g. the computer screen
- Wiring between parts. Data stored in memory can be sent to Arithmetic Unit and back.

CPU

- All computations take place here in order for the computer to perform a designated task.
- It has a large number of registers which temporarily store data and programs (instructions).
- It has circuitry to carry out arithmetic and logic operations, take decisions, etc.
- It retrieves instructions from the memory, interprets (decodes) them, and perform the requested operation.

How does a computer work?

- Stored program concept.
  - Main difference from a calculator.
- What is a program?
  - Set of instructions for carrying out a specific task.
- Where are programs stored?
  - In secondary memory, when first created.
  - Brought into main memory, during execution.

Instructions

- “Add content of cell 35 and cell 45 and put the result in cell 57 of data memory”
- “Read one digit from the keyboard and put in cell 63 of data memory.”

- Instructions are normally fetched from program memory in order, i.e from cell 1, then cell 2, ...
- “Jump” instructions:

  “If data memory cell 35 has value > 0, then fetch next instruction from cell 379 of program memory.”
- Program = sequence of numbers. Add = 1, Multiply = 2, Jump = 3, ...

Computer Programming

- Formulate the problem we want to solve as a problem of processing binary numbers (which can be represented by electrical signals)
- State what operations must be performed by the circuits. = Program
- A program is also a sequence of binary numbers, and is really an input to the circuit.

3 Incarnations of a Program

- Algorithm: “Add the product of the acceleration and time to the velocity.”
- program: “V = U + AT;”
- Program in computer memory:
  - Put product of locations 34, 35 into location 37.
  - Put sum of locations 37, 38 into location 39.

Contd.

- A digital computer is built out of tiny electronic switches.
  - From the viewpoint of ease of manufacturing and reliability, such switches can be in one of two states, ON and OFF.
  - A switch can represent a digit in the so-called *binary number system*, 0 and 1.
- A computer works based on the binary number system.

Concept of Bits and Bytes

- Bit
  - A single binary digit (0 or 1).
- Nibble
  - A collection of four bits (say, 0110).
- Byte
  - A collection of eight bits (say, 01000111).
- Word
  - Depends on the computer.
  - Typically 4 or 8 bytes (that is, 32 or 64 bits).

Contd.

- A k-bit decimal number
  - Can express unsigned integers in the range

    0 to $10^k - 1$

    - For k=3, from 0 to 999.

- A k-bit binary number
  - Can express unsigned integers in the range

    0 to $2^k - 1$

    - For k=8, from 0 to 255.
    - For k=10, from 0 to 1023.

Classification of Software

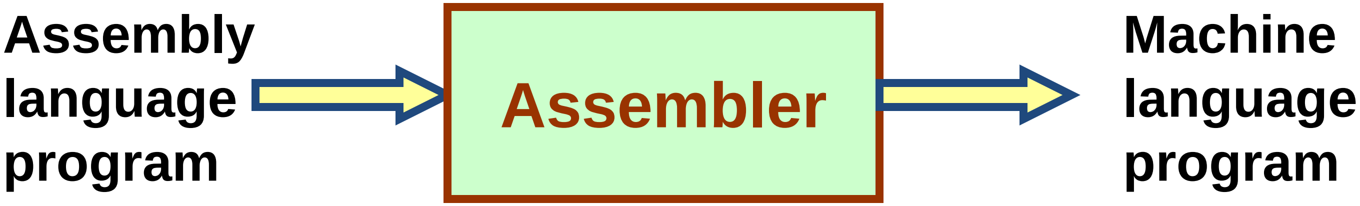
- Two categories:
  1. Application Software
     - Used to solve a particular problem.
     - Editor, financial accounting, weather forecasting, etc.
  2. System Software
     - Helps in running other programs.
     - Compiler, operating system, etc.

Computer Languages

- Machine Language
  - Expressed in binary.
  - Directly understood by the computer.
  - Not portable; varies from one machine type to another.
    - Program written for one type of machine will not run on another type of machine.
  - Difficult to use in writing programs.

Contd.

- Assembly Language
  – Mnemonic form of machine language.
  – Easier to use as compared to machine language.
    - For example, use “ADD” instead of “10110100”.
  – Not portable (like machine language).
  – Requires a translator program called *assembler*.



Contd.

- Assembly language is also difficult to use in writing programs.
  - Requires many instructions to solve a problem.
- Example: Find the average of three numbers.

```
MOV   A,X     ; A = X
ADD   A,Y     ; A = A + Y
ADD   A,Z     ; A = A + Z
DIV   A,3     ; A = A / 3
MOV   RES,A   ; RES = A
```

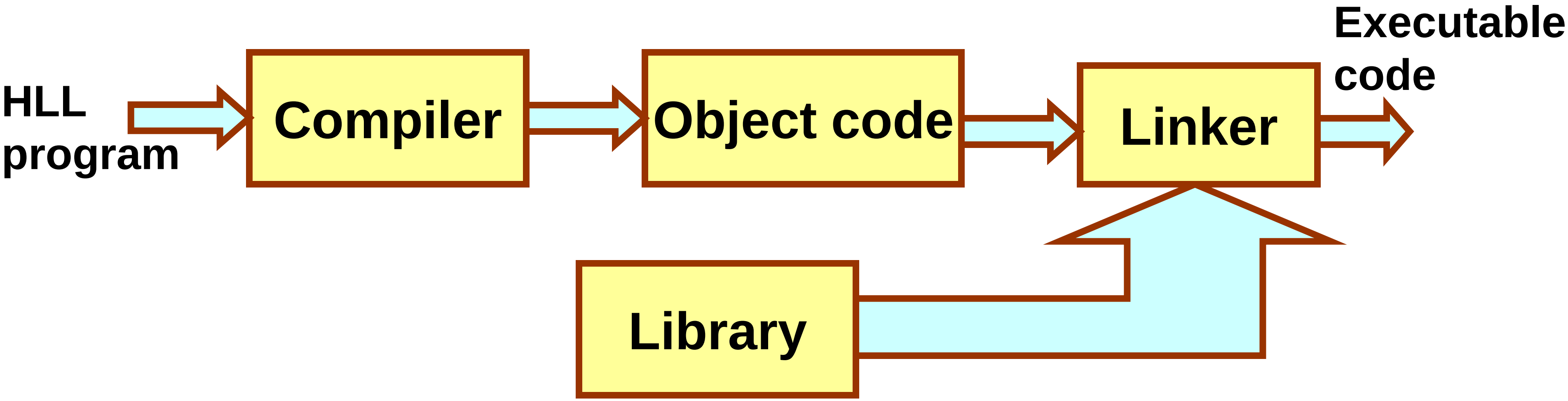
In C,

RES = (X + Y + Z) / 3

High-Level Language

- Machine language and assembly language are called low-level languages.
  - They are closer to the machine.
  - Difficult to use.
- High-level languages are easier to use.
  - They are closer to the programmer.
  - Examples:
    - Fortran, Cobol, C, C++, Java.
  - Requires an elaborate process of translation.
    - Using a software called *compiler*.
  - They are portable across platforms.

Contd.


HLL program

Compiler

Object code

Linker

Executable code

Library

To Summarize

- Assembler
  - Translates a program written in assembly language to machine language.
- Compiler
  - Translates a program written in high-level language to machine language.

Operating Systems

- Makes the computer easy to use.
    - Basically the computer is very difficult to use.
    - Understands only machine language.
- Operating systems make computers easy to use.
- Categories of operating systems:
    - Single user
    - Multi user
        - Time sharing
        - Multitasking
        - Real time

Contd.

- Popular operating systems:

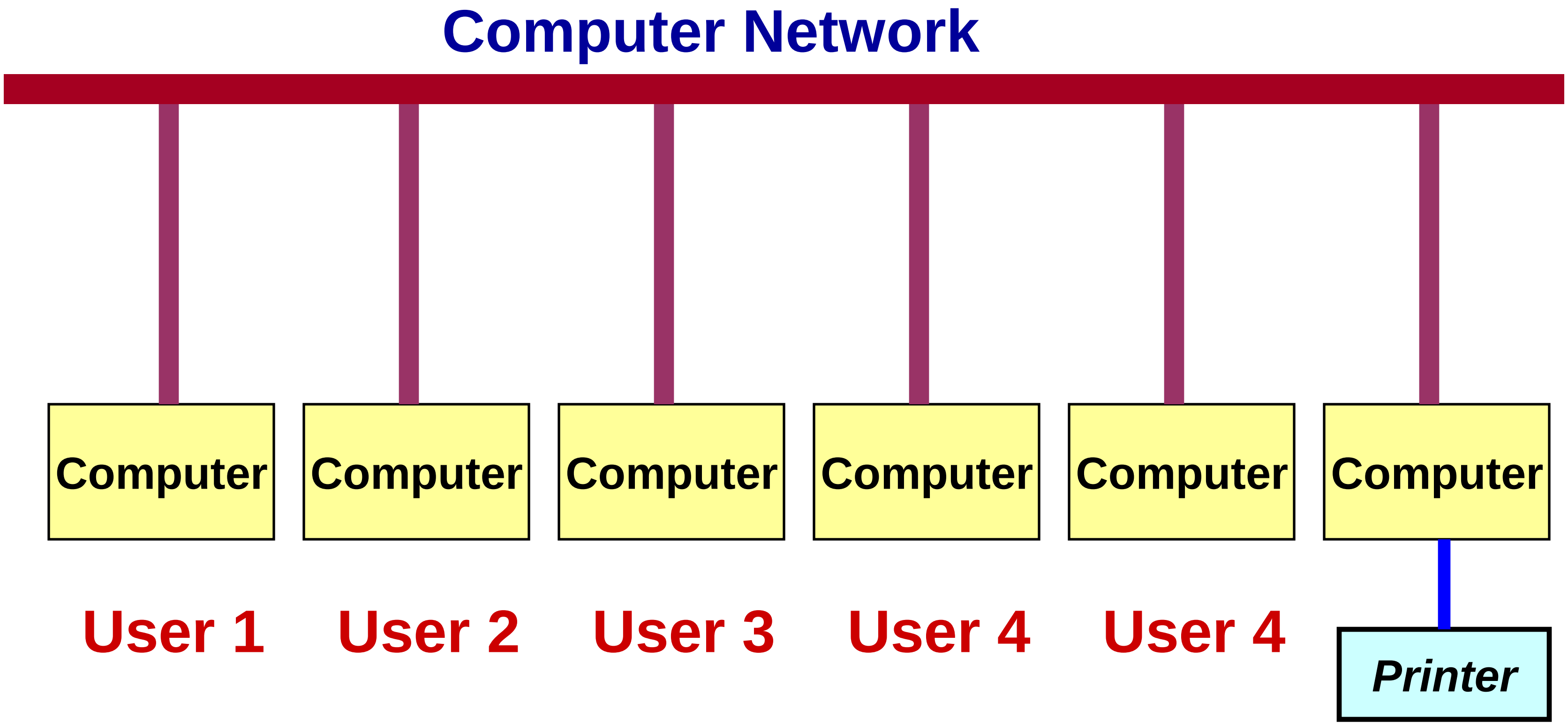
– DOS:	single-user
– Windows 2000/XP:	single-user multitasking
– Unix:	multi-user
– Linux:	a free version of Unix

- The laboratory class will be based on Linux.
- Question:
  - How multiple users can work on the same computer?

Contd.

- Computers connected in a network.
- Many users may work on a computer.
  - Over the network.
  - At the same time.
  - CPU and other resources are shared among the different programs.
    - Called time sharing.
    - One program executes at a time.

Multiuser Environment


Computer Network
Computer
Computer
Computer
Computer
Computer
Computer
User 1
User 2
User 3
User 4
User 4
Printer