ESc101: Introduction to Computers and Programming Languages

Instructor: Krithika Venkataramani
Semester 2, 2011-2012

The content of these slides is taken from previous course lectures of Prof. R. K. Ghosh, Prof. Arnab Bhattacharya, Prof. Manindra Agrawal and Prof. Dheeraj Sanghi

What are the components of a Computer?

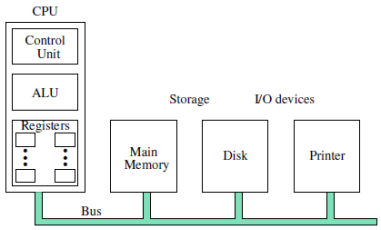
Desktop

Laptop

- Output devices
  - Monitor, printer
- Input devices
  - Keyboard
  - Mouse
  - CD, DVD, Pen-drive
- Storage
  - Hard disk, Memory
- Central Processing unit (CPU)
  - Arithmetic and Logic Unit
  - Control Unit

What are the operations of a Computer?

- Basic Operations
  - Input
  - Output
  - Storage
  - Processing
  - Control



Source: Lecture slides from Prof. R. K. Ghosh

What can a Computer do?

- A computer can carry out any computational task
- Properties of a computer
  - Precise: Will do exactly what you ask it to do { no more, no less}
  - Error-free: Will not commit errors
  - Dumb: Has no intelligence to work on its own { requires detailed instructions}
  - Faster than humans
  - Cannot perform everything that humans can

How to use a Computer?

- To execute any operation in a computer, precise instructions are required
- Algorithms are set of instructions
- Algorithms are written as programs in a computer
- So, a program is nothing but a finite set of instructions
- An algorithm can be used by a computer to generate a solution to a specific problem

What is an Algorithm?

- Algorithm: a step-wise solution to a specific problem
- Properties of an algorithm
  - Has input/output
  - Precise
  - Finite number of steps
  - Steps must be ordered
  - Must terminate (either successfully or otherwise)
  - Must work for all inputs within a specified domain
- There are many non-trivial problems for which no algorithm can be designed.
- There are many interesting problems for which there are no practical algorithms.
- There are many problems for which good algorithms exist

Algorithm: Find sum of first N natural numbers

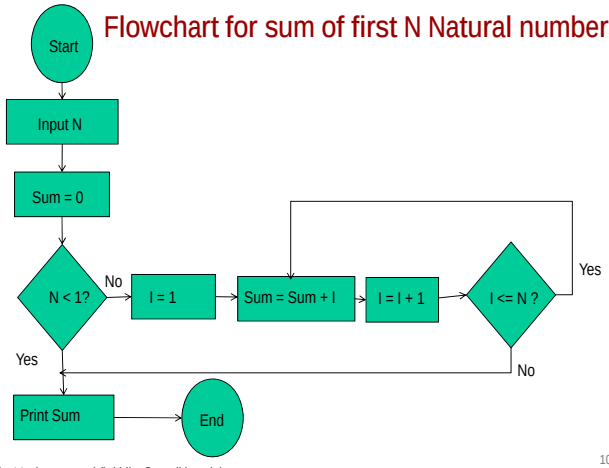
1. Input N
2. Sum = 0
3. If N < 1, go to step 8
4. I = 1
5. Sum = Sum + I
6. I = I + 1
7. If I <= N, go back to step 5
8. Print Sum

Flow Chart is used for pictorial representation of an Algorithm

- Many times, an algorithm is easy to design/understand, if it is represented pictorially.
- Different steps are shown as boxes.
- There are boxes for checking conditions, assigning values, reading, writing, etc.
- There are arrows which determine the direction of flow, or the order in which steps should be executed.

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Flowchart for sum of first N Natural numbers



Krithika Venkataramani (krithika@cse.iitk.ac.in)

More examples of algorithms

- Find the factorial of a number N
- Find the Highest Common Factor (HCF) of two numbers
- Find the Least Common Multiple (LCM) of two numbers
- Find prime factors of a number

Krithika Venkataramani (krithika@cse.iitk.ac.in)

What do Computers read and process?

- Computers can use only binary format: only 1's and 0's
- Any number in decimal system can be represented in Binary number system
- Number “6” in decimal system is equivalent to “110” in binary system
- 110 is interpreted as $0 * 2^0 + 1 * 2^1 + 1 * 2^2 = 6$

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Decimal number to Binary

- Convert decimal representation to binary

1. Divide the number by 2
2. Store quotient and remainder
3. Use quotient as the new number
4. Go back to Step 1 till quotient is 0
5. Read remainders from bottom to top to get binary representation

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Decimal number to Binary

- Convert decimal number 39 to binary
- 39/2 = 19 +1
- 19/2 = 9 +1
- 9/2 = 4 +1
- 4/2 = 2 +0
- 2/2 = 1 +0
- 1/2 = 0 +1
- Binary representation: 100111

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Binary Number to Decimal

- Convert binary representation to decimal
- Start with the right-most digit
- Multiply it by 2^0
- Take the next digit and multiply by 2^1
- Continue till the last digit (left-most digit)
- Add all these numbers to get the decimal number

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Binary Number to Decimal

- Convert binary number 100111 to decimal
- $1 * 2^0 = 1$
- $1 * 2^1 = 2$
- $1 * 2^2 = 4$
- $0 * 2^3 = 0$
- $0 * 2^4 = 0$
- $1 * 2^5 = 32$
- Decimal representation: 1+2+4+0+0+32 = 39

Krithika Venkataramani (krithika@cse.iitk.ac.in)

The Octal Basis

- Uses digits 0-7.
- Number 8 is written as 10,
  - 9 as 11,
  - 10 as 12, …
  - 15 as 17,
  - 16 as 20, ..
- Three bits of a binary number make one digit of an octal number.
- Number 11001101 is same as 315 in octal basis.

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Decimal number to Octal

- Convert decimal representation to Octal

1. Divide the number by 8
2. Store quotient and remainder
3. Use quotient as the new number
4. Go back to Step 1 till quotient is 0
5. Read remainders from bottom to top to get Octal representation

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Conversion of Decimal number to Octal

- Convert decimal number 39 to Octal
- 39/8 = 4 +7
- 4/8 = 0 +4
- Octal representation: 47

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Hexadecimal System

- Uses digits 0-9 and letters a-f .
- Number 10 is written as a,
- 11 as b,
- 12 as c, …
- 15 as f ,
- 16 as 10, ..
- Four bits of a binary number make one digit of hexadecimal number.
- Number 11001101 is same as cd in hexadecimal basis.

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Machine Language

- Instructions are in binary format as well
- Example: adding two numbers 2 and 3
  - 0011000
  - 0011010
  - 0011011
  - 0010100
- This is in machine language
- Above programs are hard for humans to write and understand
- But, computers understand only these

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Machine Language

- Program : sequence of instructions in machine language.
- The execution happens as follows:
  - Control unit fetches an instruction
  - If operands are in memory locations, fetch them as well.
  - Ask ALU to execute the instruction on the operands
  - The result needs to be stored in the memory location.
  - Next instruction is fetched, unless the previous instruction asked the Control Unit to fetch a difference instruction.

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Programming Languages

- Machine Language is hard for humans to use
- Programming Languages like C are better for humans to understand
- The instructions from programming languages are converted to machine language using a compiler
- The converted (compiled) machine language instructions are then processed by the computer to perform the task (execute)

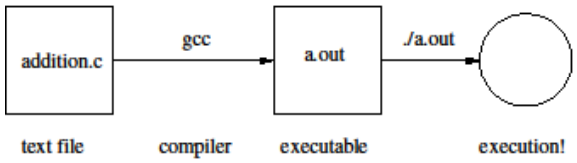
Krithika Venkataramani (krithika@cse.iitk.ac.in)

Writing C programs in Linux OS

- Create files with .c suffix (example: addition.c)
- Linux is an operating system that is (still) mostly command-based
- Use gcc to compile (gcc addition.c)
- Will create a.out (execute by typing ./a.out)
- Some useful commands
  - cd: change directory
  - mv: move or rename
  - rm: delete
  - mkdir: create directory
  - vim filename.c: opens le filename.c for editing
  - ls: lists contents in a directory
  - pwd: shows the present working directory

Krithika Venkataramani (krithika@cse.iitk.ac.in)

Executing programs in C



Example Program to Add two numbers

```
/* Program to add two numbers */
#include <stdio.h>        // Include
headers
void main() // Main function
{
int a, b, c;   // Declare variables
scanf("%d", &a);  // Read 'a' from
keyboard
scanf("%d", &b);  // Read 'b' from
keyboard
c = a + b;
printf("%d\n", c);  // Write 'c' to screen
}
```


Crash course in C

- Comments in a program are not translated into machine language by compiler
  - gcc simply ignores them

  /* Program . . . */

  #include . . . // I n c l u d e he ade r
- In general, certain standard files are needed

  #include<stdio.h>
- Actual execution of the program begins at main
  - Every instruction inside main is executed step by step

  void main ( )

  {

  . . .

  }

Crash course in C (cont.)

- Variables help define the components of a program
  - a, b, c represent the different numbers
- Types of each variable must be defined
  - a, b, c are integers

  int a, b, c ;
- Reading (or input of) a number is done using scanf

  scanf(“%d”, &a) ;
- Arithmetic (or logical) operations are written using operators
  - Addition is denoted by +

  c = a + b ;
- Writing (or output of) a number is done using printf

  printf(“%d\n”, c) ;