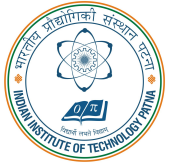
Data Structures and Programming Language



Arijit Mondal
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
arijit@iitp.ac.in

Introduction

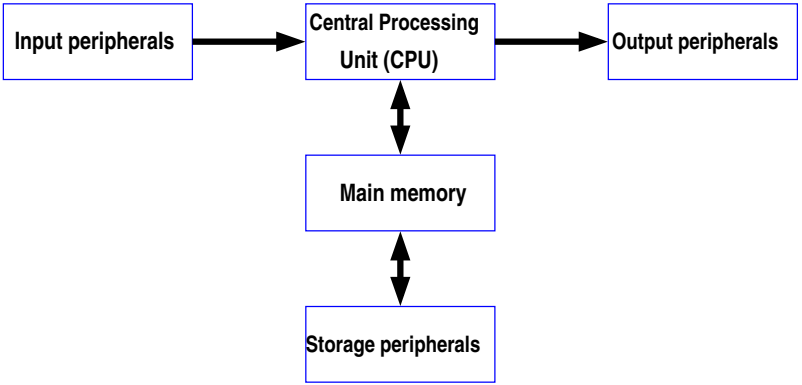
Course syllabus

- Introduction to C
- Variables, data type
- Statement, Conditional statement
- Loop construct
- Array, structure, union
- Function, Recursion
- Pointers
- Stack, queue, tree
- Searching, Sorting
- File handling

Books

- **Programming with C** by Byron Gottfried, Third Edition, Schaums Outlines Series,
- **The C Programming Language** by Brian W Kernighan, Dennis M Ritchie
- **Data structures** by S. Lipschutz, Schaums Outline Series

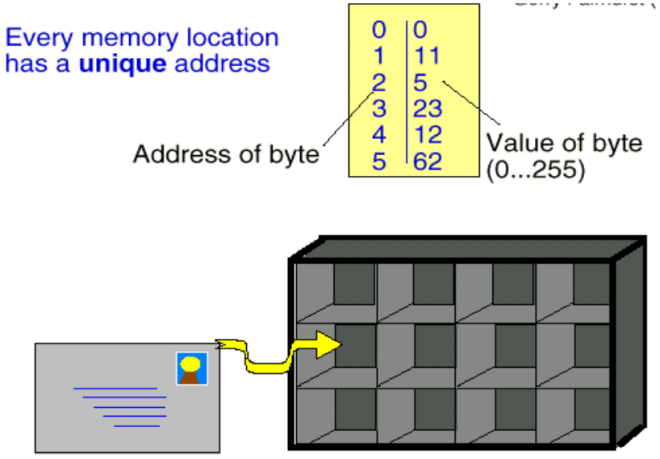
Simple view of computer



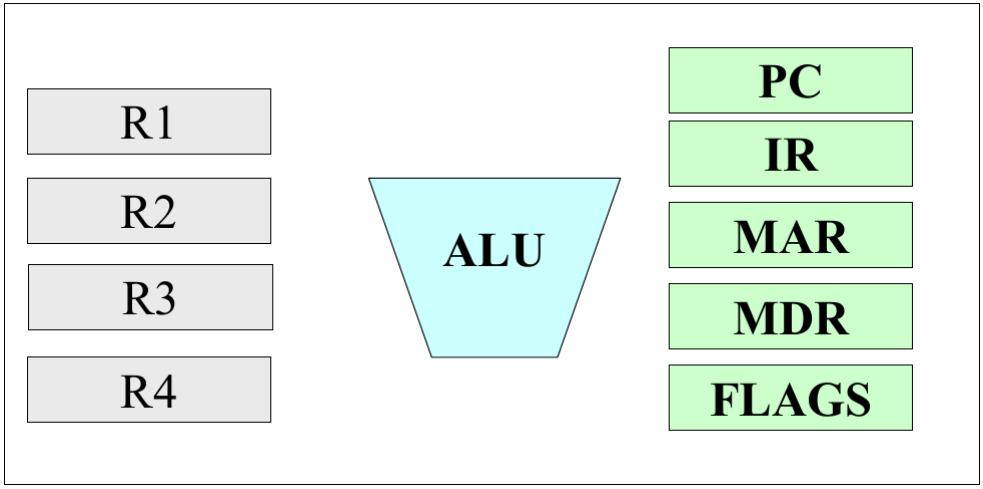
Peripherals

- Input devices
  - Keyboard, mouse, webcam
- Output devices
  - Monitor, printer, speaker
- Storage peripherals
  - Magnetic disks - hard disk
  - Optical disks - CDROM
  - Flash memory - pen drives

View of memory



A simple view of CPU



What can a computer do

- Check prime number
- Palindrome recognizer
- Find shortest path between two points
- Telephone pole placement
- Spaceship control
- Finger-print recognition
- Play chess
- Speech recognition
- Language recognition and many more!

Program and Software

- Computer needs to be **programmed** to do such tasks
- **Programming** is the process of writing instructions in a **language** that can be understood by the computer so that a desired task can be performed by it
- **Program:** sequence of instructions to do a task, computer processes the instructions sequentially one after the other
- **Software:** programs for doing tasks on computers
- CPU understand **machine language**
  - Different strings of 0's and 1's only
  - Hard to remember and use
- **Instruction set** of CPU
  - Mnemonic names for this strings

Instruction set & Program

Instruction set

start
read m
write m
load data,m
copy m1,m2
add m1,m2,m3
compare m1,m2,m3
jump l
jz m,l
halt

Program

1. start
2. read 10
3. read 11
4. add 10,11,12
5. write 12
6. halt

Programming issue with instruction set

- Instruction sets of different types of CPUs different
  - Need to write different programs for computers with different types of CPUs even to do the same thing
- Still hard to remember
- Solution: High level languages (C, C++, Java,...)
  - CPU neutral, one program for many
  - **Compiler** to convert from high-level program to low level program that CPU understands

High vs Low level program

```
variable x,y
begin
read x
read y
if(x>y) then write x
else write y
end
```

1. start
2. read 10
3. read 11
4. compare 10,11,12
5. jz 12,7
6. write 10
7. jump 9
8. write 11
9. halt

Three steps in writing programs

- **Step 1:** Write the program in a high-level language (in your case, C)
- **Step 2:** Compile the program using a C compiler
- **Step 3:** Run the program (as the computer to execute it)

Fundamentals of C

First C program

```
#include <stdio.h>
void main()
{
  printf("Hello, World!\n");
}
```

More print

```
#include <stdio.h>
void main()
{
  printf("Hello, World!\n");
  printf("Hello,\n World!\n");
}
```

More print

```
#include <stdio.h>
void main()
{
  printf("Hello, World!\n");
  printf("Hello,\n World!\n");
  printf("Hello,\t World!\n");
}
```

Reading values from keyboard

```
#include <stdio.h>
void main()
{
  int number;
  scanf("%d",&number);
  printf("Number of students in this class is %d\n",number);
}
```

Centigrade to Fahrenheit

```
#include <stdio.h>
void main()
{
  float cent,fahr;
  scanf("%d",&cent);
  fahr=cent*(9.0/5.0)+32;
  printf("%f C equals to %f\n",cent,fahr);
}
```

Maximum of two numbers

```
#include <stdio.h>
void main()
{
  int x,y;
  scanf("%d%d",&x,&y);
  if(x>y) printf("Largest is %d\n",x);
  else printf("Largest is %d\n",y);
}
```

What will be the output?

```
#include <stdio.h>
void main()
{
  int x,y;
  scanf("%d%d",&x,&y);
  if(x>y) printf("Largest is %d\n",x);
  printf("Largest is %d\n",y);
}
```

The C character set

- C language alphabet
  - Uppercase letters 'A' to 'Z'
  - Lowercase letters 'a' to 'z'
  - Digits '0' to '9'
  - Special characters: ! # % ^& * - _ + = ~[ ] \| ; : ' " { } , . < > / ? blank
- **A C program should not contain anything else**

Structure of a C program

- A collection of functions
- Exactly one special function named **main** must be present.
  - Program always starts from there
- Each function has statements for declaration, assignment, condition check, looping, etc.
- Statements are executed one by one

Variables

- Very important concept for programming
- An entity that has a value and is known to the program by a name
- Can store any temporary result while executing a program
- Can have only one value assigned to it at any given time during the execution of the program
- The value of a variable can be changed during the execution of the program
- Variables stored in memory
- Remember that memory is a list of storage locations, each having a unique address
- A variable is like a bin
  - The contents of the bin is the value of the variable
  - The variable name is used to refer to the value of the variable
  - A variable is mapped to a location of the memory, called its address

Example

```
#include <stdio.h>
void main()
{
  int x;
  int y;
  x=1;
  y=3;
  printf("x=%d, y=%d\n",x,y);
}
```

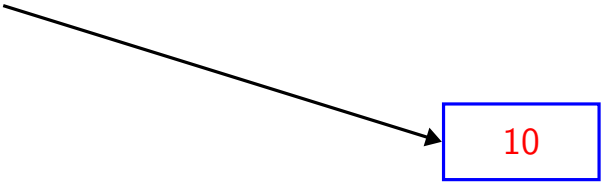
Variables in memory

$x = 10$

$x = 20$

$x = x + 1$

$x = x * 5$



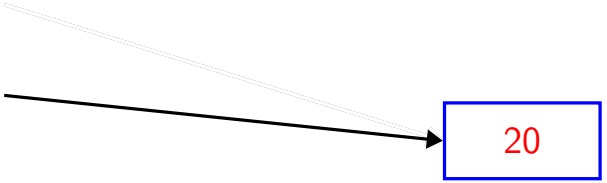
Variables in memory

$x = 10$

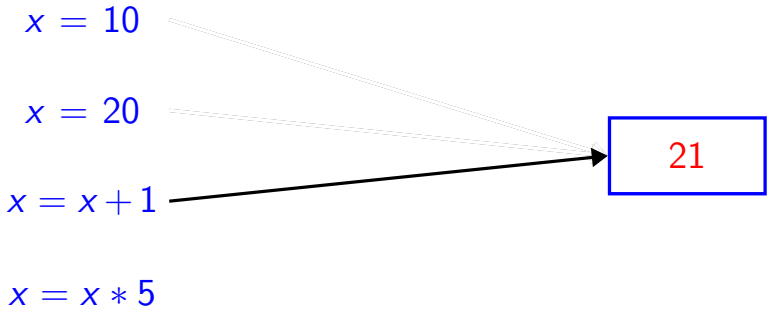
$x = 20$

$x = x + 1$

$x = x * 5$



Variables in memory



Variables in memory