General Announcements

CS11001: Programming & Data Structures

Pallab Dasgupta
Professor, Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur

Course Materials

- Slides will be made available at http://www.facweb.iitkgp.ernet.in/~pallab/course.html

Books:

1. Programming with C (Second Edition)

   Byron Gottfried, Third Edition, Schaum's Outlines Series, Tata McGraw-Hill, 2011

2. Data structures

   S. Lipschutz, Schaum's Outline Series, Tata McGraw-Hill, 2010

3. The C Programming Language

   Brian W Kernighan, Dennis M Ritchie, Prentice Hall India

Many other books are available and may serve the same purpose

About the Course

- Venue [Sec 4,5]: F-116
- Class Timings: W(11:30-12:25), Th(10:30-11:25), F(8:30-9:25)
- Tutorial classes (one hour per week) will be conducted on a “per section” basis before Lab hours.
- Evaluation in the theory course:

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

Attendance REALLY matters

- Any student with less than 80% attendance may be deregistered from the course
- Leave due to medical reasons must be certified by the B.C. Roy Technology Hospital

Important Dates

- Class Test-1: Feb 2, 2011 (18:30 – 19:30)
- Class Test-2: March 30, 2011 (18:30 – 19:30)

- Mid-semester: Feb 18-25, 2011
- End-semester: April 22-29, 2011

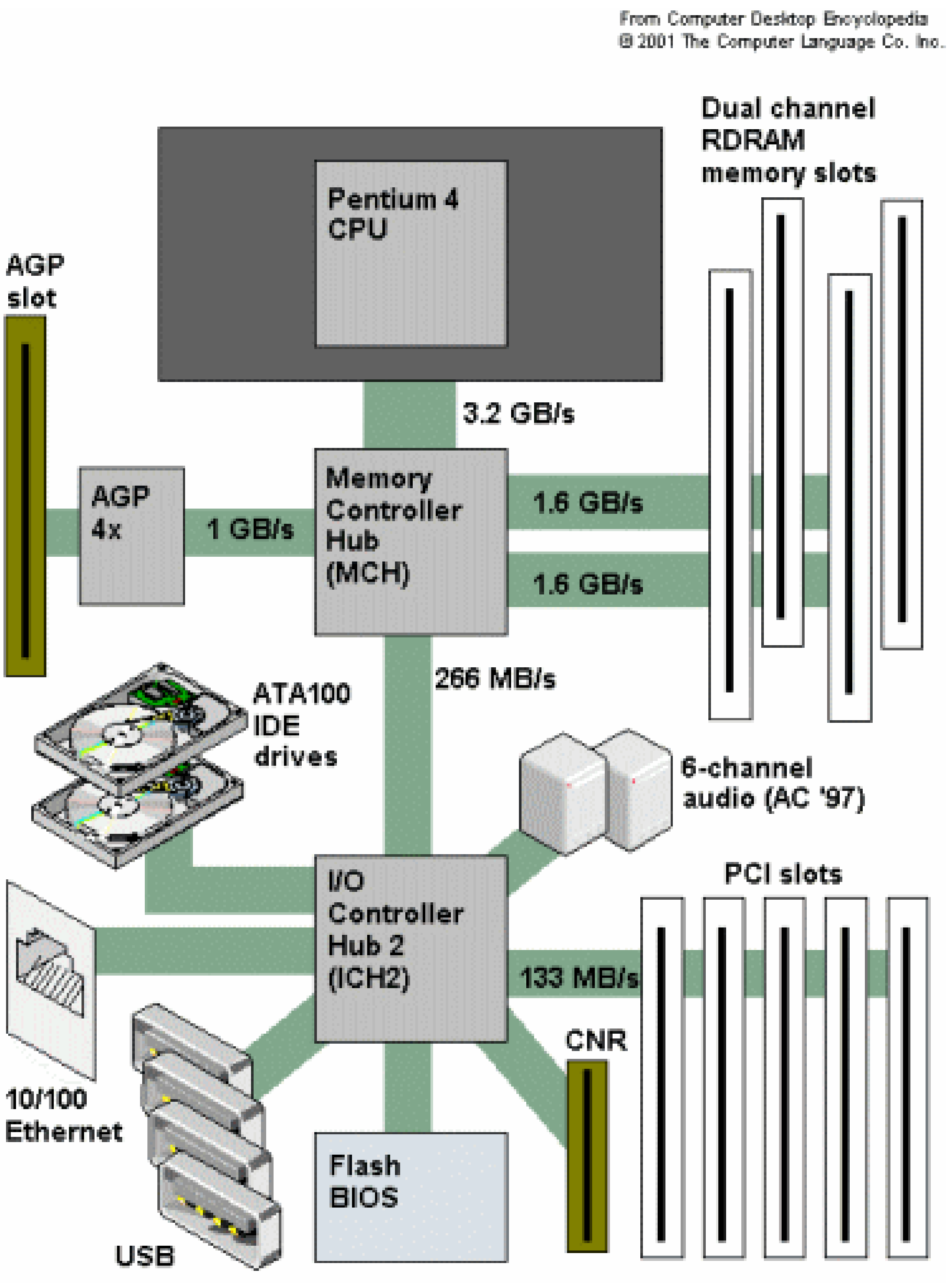
Introduction

CS11001: Programming & Data Structures

Pallab Dasgupta
Professor, Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur

Architecture

Typical system architecture for a desktop PC


From Computer Desktop Encyclopedia
© 2001 The Computer Language Co. Inc.
Dual channel RDRAM memory slots
Pentium 4 CPU
AGP slot
3.2 GB/s
AGP 4x
1 GB/s
Memory Controller Hub (MCH)
1.6 GB/s
1.6 GB/s
266 MB/s
ATA100 IDE drives
6-channel audio (AC '97)
PCI slots
I/O Controller Hub 2 (ICH2)
133 MB/s
CNR
10/100 Ethernet
Flash BIOS
USB

CPU (Central Processing Unit)

  - 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 functional units (circuitry) to carry out arithmetic and logic operations
  - It retrieves instructions from the memory, interprets (decodes) them, and performs the requested operation
  - Fetch → Decode → Execute cycle
- CPU is also referred to as the processor
- Computers may have multiple processors
- Modern processors are multi-core (multiple processors in one chip)

Main Memory

- Uses semiconductor technology
  - Allows direct access

- Memory sizes in the range of 256 MegaBytes to 8 GigaBytes are typical today.

- Some measures to be remembered
  - 1 K = 2^{10} (= 1024)
  - 1 M = 2^{20} (= one million approx.)
  - 1 G = 2^{30} (= one billion approx.)

I/O and Peripherals

- Input Device
  - Keyboard, Mouse, Scanner, Digital Camera
- Output Device
  - Monitor, Printer
- Storage Peripherals
  - Magnetic Disks: hard disk, floppy disk (obsolete)
    - Allows direct (semi-random) access
  - Optical Disks: CDROM, CD-RW, DVD
    - Allows direct (semi-random) access
  - Flash Memory: pen drives
    - Allows direct access
  - Magnetic Tape: DAT (obsolete)
    - Only sequential access

A Sample Configuration of a PC

• Processor:	Intel® Core™ i3-530 Processor (2.93GHz 1333MHz 4MB)
• Total memory:	2 GB DDR3 1333MHz
• Display type:	23.0 " With integrated camera 0.3M 1920x1080
• Hard drive device:	320GB
• Optical device:	DVD Recordable (Dual Layer)
• Input Device:	Keyboard, Mouse
• Ports:	USB, Infrared
• Chipset …	
• Graphics …	

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.

What is the agenda?

- To learn the logic of programs
  - C is only the vehicle
- To learn the basics of how a computer works
- To learn elementary data structures

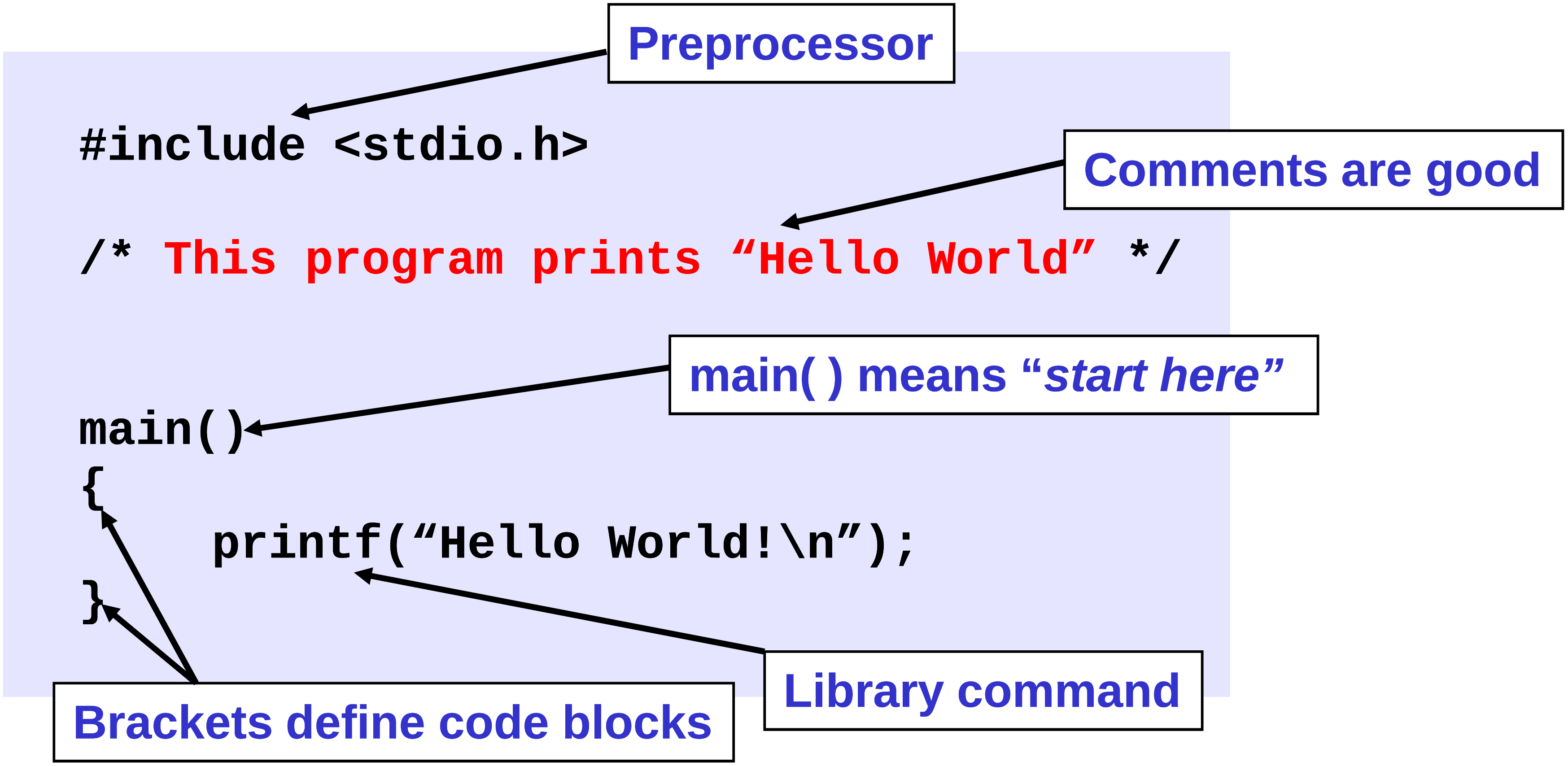
Why teach C?

- C is *small* (only 32 keywords).
- C is *common* (lots of C code about).
- C is *stable* (the language doesn’t change much).
- C is *quick running*.
- C is the *basis for many other languages* (Java, C++, awk, Perl).
- It may not feel like it but C is one of the easiest languages to learn.

Some programmer jargon

- Some words that will be used a lot:
  - Source code: The stuff you type into the computer. The program you are writing.
  - Compile (build): Taking source code and making a program that the computer can understand.
  - Executable: The compiled program that the computer can run.
  - Language: The core part of C central to writing C code.
  - Library: Added functions for C programming which are bolted on to do certain tasks.
  - Header file: Files ending in .h which are included at the start of source code.

Our First C Program: *Hello World*



```
#include <stdio.h>

/* This program prints “Hello World” */


main()
{
     printf(“Hello World!\n”);
}
```

Preprocessor

Comments are good

main( ) means “*start here*”

Library command

Brackets define code blocks

C doesn’t care much about spaces

```
#include <stdio.h> /* This program prints “Hello World” */
int main( ) {printf(“Hello World!\n”);}
```

```
#include <stdio.h>
/* This program
prints “Hello
World”
*/
int
main( )
{
printf(“Hello
World!
\n”)
;
}
```

Both of these programs are exactly the same as the original as far as your compiler is concerned.

Keywords of C

- Flow control (6) – `if, else, return, switch, case, default`
- Loops (5) – `for, do, while, break, continue`
- Common *types* (5) – `int, float, double, char, void`
- *structures* (3) – `struct, typedef, union`
- Counting and sizing things (2) – `enum, sizeof`
- Rare but still useful *types* (7) – `extern, signed, unsigned, long, short, static, const`
- Evil keywords which we avoid (1) – `goto`
- Wierdies (3) – `auto, register, volatile`

The C Character Set

- **The C language alphabet:**
  - **Uppercase letters ‘A’ to ‘Z’**
  - **Lowercase letters ‘a’ to ‘z’**
  - **Digits ‘0’ to ‘9’**
  - **Certain special characters:**

!	#	%	^	&	*	(	)
-	_	+	=	~	[	]	\
|	;	:	‘	“	{	}	,
.	<	>	/	?	blank		

Some simple operations for variables

- In addition to +, -, * and / we can also use
  +=, -=, *=, /=, -- and % (modulo)

`n++`	*increment n*
`n--`	*decrement n*

`a+=5`	*is equivalent to*	`a = a+5;`
`a-=5`	*is equivalent to*	`a = a-5;`
`a*=5`	*is equivalent to*	`a = a*5;`
`a/=5`	*is equivalent to*	`a = a/5;`

`(x % y)` gives the remainder when x is divided by y

Classification of Software

- Two categories:

1. **Application Software**
    - Used to solve a particular problem.
    - Editor, financial accounting, weather forecasting, etc.

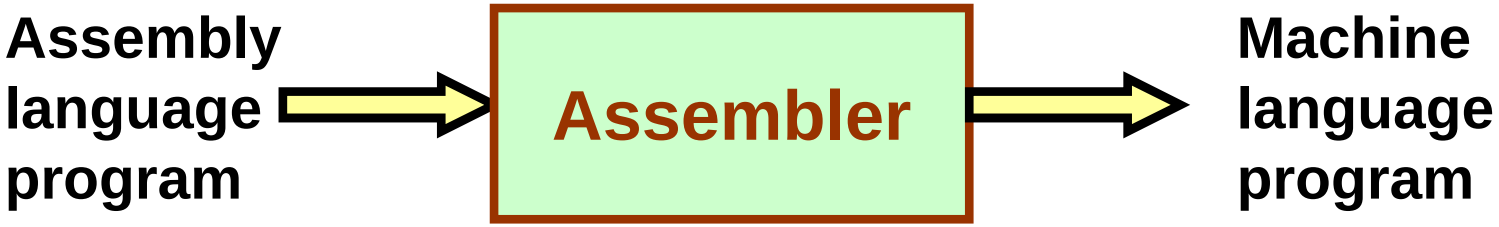
2. **System Software**
    - Helps in running other programs.
    - Compiler, operating system, etc.

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,

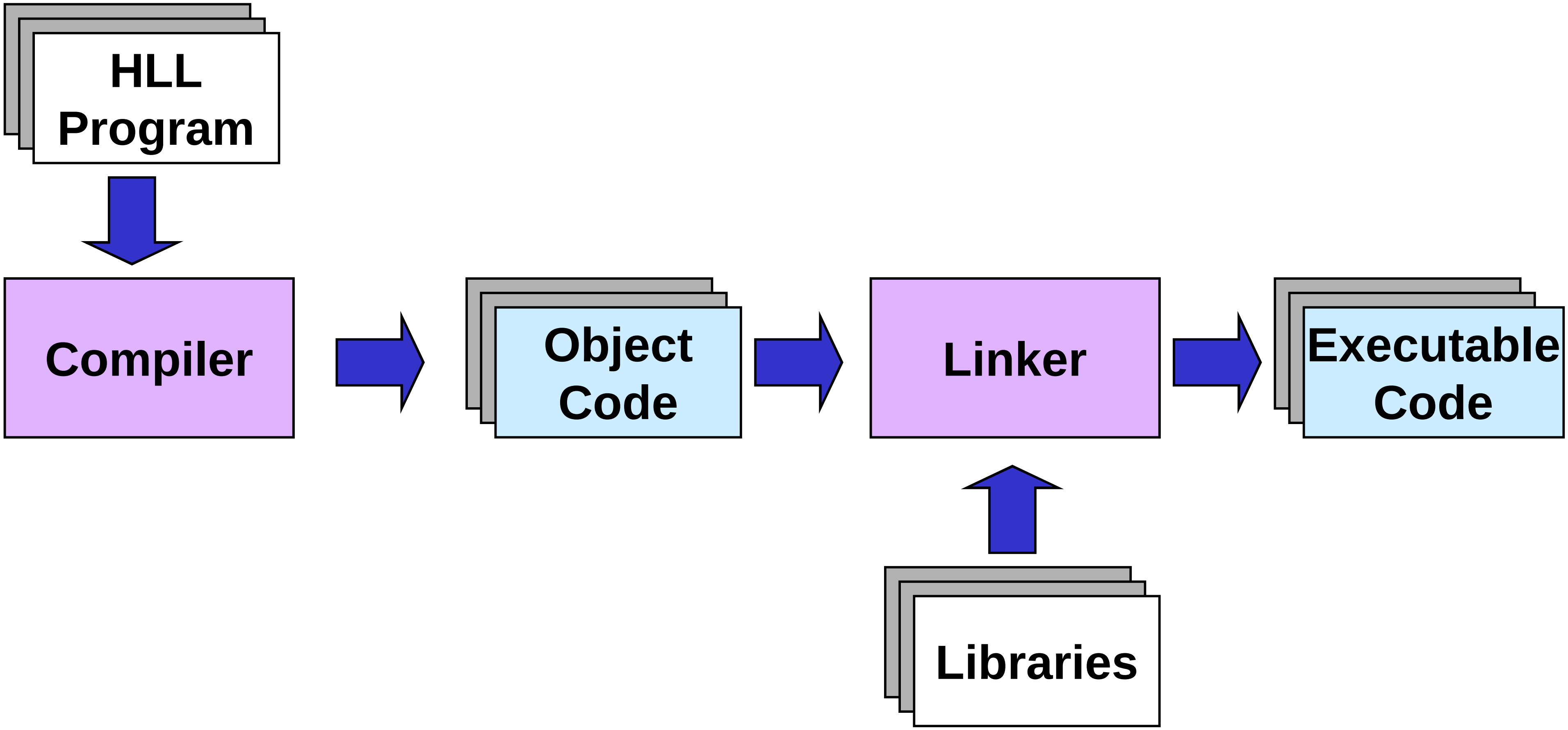
```
RES = (X + Y + Z) / 3
```

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.**

From HLL to executable



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:
  - 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.