Escola de Engenharia
Universidade do Minho

Programação – Aula Teórica 1

Introdução à Computação e à Programação em C/C++

Luís Paulo Reis

lpreis@dsi.uminho.pt

Professor Associado do Departamento de Sistemas de Informação, Escola de Engenharia,
Universidade do Minho, Portugal

(Slides Baseados em Deitel e Deitel 2010 e L.P.Reis et al., 2006)

Escola de Engenharia
Universidade do Minho

Introduction to Computers and C++ Programming

Outline

1.1 Introduction
1.2 What Is a Computer?
1.3 Computer Organization
1.4 Evolution of Operating Systems
1.5 Personal Computing, Distributed Computing and Client/Server Computing
1.6 Machine Languages, Assembly Languages and High-level Languages
1.7 The History of C
1.8 The C Standard Library
1.9 The Key Software Trend: Object Technology

Introduction to Computers and C++ Programming

Outline

1.10 C and C++ Programming
1.11 Java and other High-level Languages
1.12 Structured Programming
1.13 The Basics of a typical C Program Development Environment
1.14 Hardware Trends
1.15 History of the Internet
1.16 History of the World Wide Web
1.17 General Notes About C

Objectives

- In this lesson, you will learn:
  - To understand basic computer concepts
  - To become familiar with different types of programming languages
  - To become familiar with the history of the C programming language
  - To become aware of the C standard library
  - To understand the elements of a typical C program development environment
  - To appreciate why it is important to learn C in a first programming course
  - To appreciate why C provides a foundation for further study of programming languages in general and of C++ and Java in particular

1.1 Introduction

- **Course content**
  - Basics of algorithms and data structures
  - The C/C++ programming language
  - Structured programming and proper programming techniques

1.2 What is a Computer?

- **Computer**
  - Device capable of performing computations and making logical decisions
  - Computers process data under the control of sets of instructions called computer programs
- **Hardware**
  - Various devices comprising a computer
  - Keyboard, screen, mouse, disks, memory, CD-ROM, and processing units
- **Software**
  - Programs that run on a computer

1.3 Computer Organization

Six logical units in every computer:

1. Input unit
   - Obtains information from input devices (keyboard, mouse)
2. Output unit
   - Outputs information (to screen, to printer, to control other devices)
3. Memory unit
   - Rapid access, low capacity, stores input information
4. Arithmetic and logic unit (ALU)
   - Performs arithmetic calculations and logic decisions
5. Central processing unit (CPU)
   - Supervises and coordinates the other sections of the computer
6. Secondary storage unit
   - Cheap, long-term, high-capacity storage
   - Stores inactive programs

1.4 Evolution of Operating Systems

- **Batch processing**
  - Do only one job or task at a time
- **Operating systems**
  - Manage transitions between jobs
  - Increased throughput:
    - Amount of work computers process
- **Multiprogramming**
  - Computer resources are shared by many jobs or tasks
- **Timesharing**
  - Computer runs a small portion of one user’s job then moves on to service the next user

1.5 Personal Computing, Distributed Computing, and Client/Server Computing

- **Personal computers**
  - Economical enough for individual
- **Distributed computing**
  - Computing distributed over networks
- **Client/server computing**
  - Sharing of information across computer networks between file servers and clients (personal computers)

1.6 Machine Languages, Assembly Languages, and High-level Languages

Three types of programming languages

1. Machine languages
   - Strings of numbers giving machine specific instructions
   - Example: `+1300042774      +1400593419      +1200274027`
2. Assembly languages
   - English-like abbreviations representing elementary computer operations (translated via assemblers)
   - Example:

```
LOAD     BASEPAY
ADD      OVERPAY
STORE    GROSSPAY
```

3. High-level languages
   - Codes similar to everyday English
   - Use mathematical notations (translated via compilers)
   - Example: `grossPay = basePay + overTimePay`

1.7 History of C

- **C Language**
  - Evolved by Ritchie from two previous programming languages, BCPL and B
  - Used to develop UNIX operating system
  - Used to write modern operating systems
  - Hardware independent (portable)
  - By late 1970's C had evolved to "Traditional C"
- **Standardization**
  - Many slight variations of C existed, and were incompatible
  - Committee formed to create a "unambiguous, machine-independent" definition
  - Standard created in 1989, updated in 1999

1.8 The C Standard Library

- **C programs consist of pieces/modules called functions**
  - A programmer can create his own functions
    - Advantage: the programmer knows exactly how it works
    - Disadvantage: time consuming
  - Programmers will often use the C library functions
    - Use these as building blocks
  - Avoid re-inventing the wheel
    - If a premade function exists, generally best to use it rather than write your own
    - Library functions carefully written, efficient, and portable

1.9 The Key Software Trend: Object Technology

- **Objects**
  - Reusable software components that model items in the real world
  - Meaningful software units
    - Date objects, time objects, paycheck objects, invoice objects, audio objects, video objects, file objects, record objects, etc.
    - Any noun can be represented as an object
  - Very reusable
  - More understandable, better organized, and easier to maintain than procedural programming
  - Favor modularity

1.10 C and C++ Programming

- **C++**
  - Superset of C developed by Bjarne Stroustrup at Bell Labs
  - "Spruces up" C, and provides object-oriented capabilities
  - Object-oriented design very powerful
    - 10 to 100 fold increase in productivity
  - Dominant language in industry and academia
- **Learning C++**
  - Because C++ includes C, some feel it is best to master C, then learn C++
  - However whenever useful we will introduce some C++ concepts

1.11 Java and other High-level Languages

- **Java is used to**
  - Create Web pages with dynamic and interactive content
  - Develop large-scale enterprise applications
  - Enhance the functionality of Web servers
  - Provide applications for consumer devices (such as cell phones, pagers and personal digital assistants)

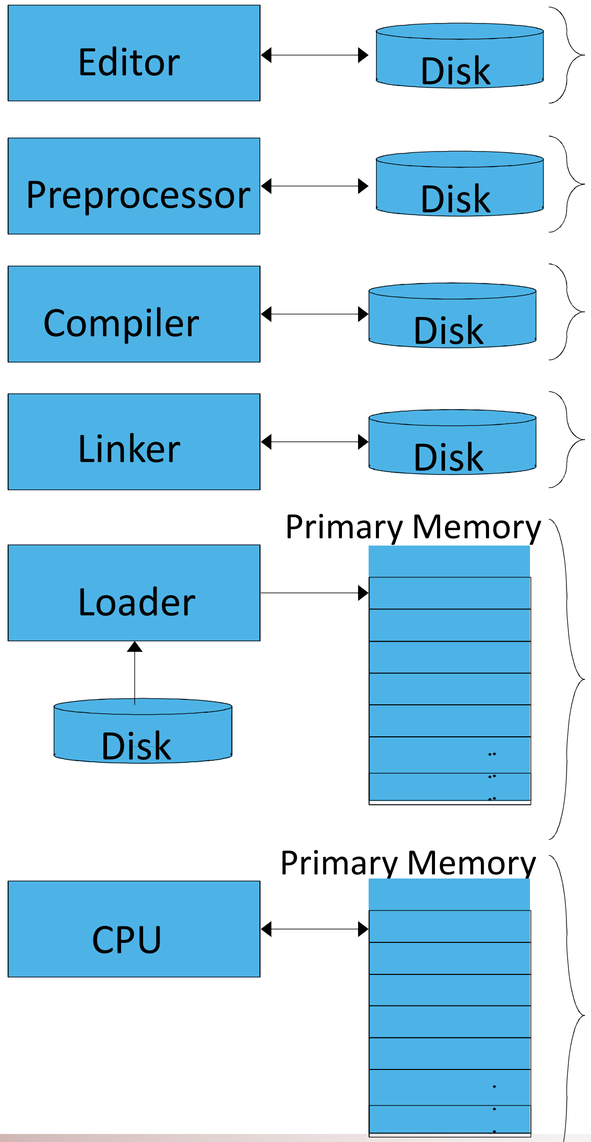
- **Other high-level languages**
  - FORTRAN
    - Used for scientific and engineering applications
  - COBOL
    - Used to manipulate large amounts of data
  - Pascal
    - Intended for academic use (although Delphi is very useful)
  - Prolog
    - High-level logic programming

1.12 Structured Programming

- **Structured programming**
  - Disciplined approach to writing programs
  - Clear, easy to test and debug and easy to modify
- **Multitasking**
  - Specifying that many activities run in parallel

1.13 Basics of a Typical C Program Development Environment

- Phases of C/C++ Programs:
  1. *Edit*
  2. *Preprocess*
  3. *Compile*
  4. *Link*
  5. *Load*
  6. *Execute*



Program is created in the editor and stored on disk.

Preprocessor program processes the code.

Compiler creates object code and stores it on disk.

Linker links the object code with the libraries

Loader puts program in memory.

CPU takes each instruction and executes it, possibly storing new data values as the program executes.

1.14 Hardware Trends

- **Every year or two the following approximately double:**
  - Amount of memory in which to execute programs
  - Amount of secondary storage (such as disk storage)
    - Used to hold programs and data over the longer term
  - Processor speeds
    - The speeds at which computers execute their programs

1.15 History of the Internet

- **The Internet enables**
  - Quick and easy communication via e-mail
  - International networking of computers
- **Packet switching**
  - The transfer of digital data via small packets
  - Allows multiple users to send and receive data simultaneously
- **No centralized control**
  - If one part of the Internet fails, other parts can still operate
- **TCP/IP**
- **Bandwidth**
  - Information carrying capacity of communications lines

1.16 History of the World Wide Web

- **World Wide Web**
  - Locate and view multimedia-based documents on almost any subject
  - Makes information instantly and conveniently accessible worldwide
  - Possible for individuals and small businesses to get worldwide exposure
  - Changing the way business is done

1.17 General Notes About C

- **Program clarity**
  - Programs that are convoluted are difficult to read, understand, and modify
- **C is a portable language**
  - Programs can run on many different computers
  - However, portability is an elusive goal
- **Careful walkthrough of C**
  - Some details and subtleties are not covered
  - If you need additional technical details
    - Read the C standard document
    - Read the book by Kernigan and Ritchie
    - Read other course documentation

Escola de Engenharia
Universidade do Minho

Questões?

Programação – Aula Teórica 1

Introdução à Computação e à Programação em C/C++

Luís Paulo Reis

lpreis@dsi.uminho.pt

**Professor Associado do Departamento de Sistemas de Informação, Escola de Engenharia,
Universidade do Minho, Portugal**

(Slides Baseados em Deitel e Deitel 2010 e L.P.Reis et al., 2006)