

第1章 程序设计和C语言概述

1.1 什么是计算机程序

1.2 什么是计算机语言

1.3 C语言的发展及其特点

1.4 最简单的C语言程序

1.5 运行C程序的步骤与方法

1.6 程序设计的任务

1.1 什么是计算机程序

- **程序**：一组计算机能识别和执行的**指令**
- 只要让计算机执行这个程序，计算机就会**自动地、有条不紊地**进行工作
- 计算机的一切操作都是由**程序**控制的，离开程序，计算机将一事无成

1.2 什么是计算机语言

➤ **计算机语言**：人和计算机交流信息的、计算机和人都能识别的语言

1.2 什么是计算机语言

低级语言

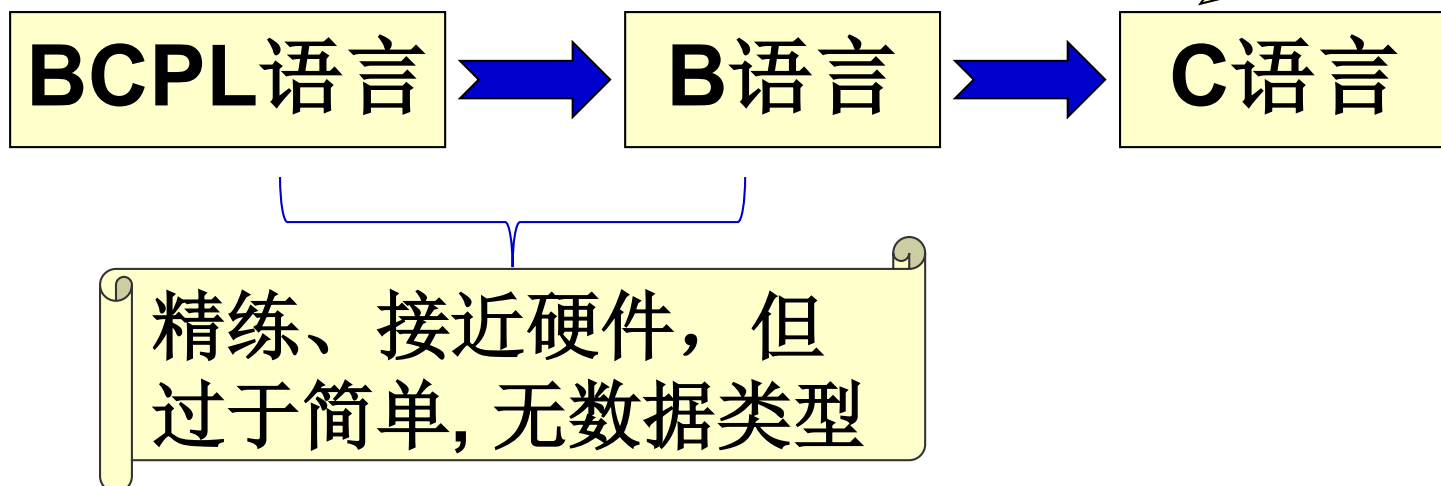
➤ 计算机语言发展阶段：

- ◆ 机器语言（由**0**和**1**组成的指令）
- ◆ 符号语言（用英文字母和数字表示指令）
- ◆ 高级语言（接近于人的自然语言和数学语言）
 - 面向**过程**的语言
（非结构化的语言、结构化语言）
 - 面向**对象**的语言

1.3 C语言的发展及其特点

➤ **C语言**是国际上广泛流行的计算机高级语言。

➤ **C语言的发展**:



1.3 C语言的发展及其特点

- **1963年英国剑桥大学推出了CPL（Combined Programming Language）语言。CPL语言在ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。**
- **1967年英国剑桥大学的 Martin Richards对CPL语言做了简化，推出了 BCPL（Base Combined Programming Language）语言。**
- **1970年美国贝尔实验室的 Ken Thompson 以BCPL 语言为基础，又作了进一步的简化，设计出了很简单的而且很接近硬件的 B 语言（取BCPL的第一个字母），并用B语言写出了第一个 UNIX 操作系统。但 B 语言过于简单，功能有限。**

1.3 C语言的发展及其特点

- **1972年至1973年间**，贝尔实验室的 **D.M.Ritchie** 在 **B** 语言的基础上设计出了 **C** 语言（取**BCPL**的第二个字母），用于描述和实现**UNIX**操作系统。**C**语言既保持**BCPL**语言和 **B** 语言的优点（精练、接近硬件），又克服了他们的缺点（过于简单，数据无类型等）。
- **1975年**用**C**语言改写的**UNIX**第六版发布
- **1977年**出现了《可移植**C**语言编译程序》推动了**UNIX**在各种机器上实现
- **1978年**影响深远的名著《**The C Programming Language**》由 **Brian W. Kernighan**和**Dennis M. Ritchie** 合著,被称为**K&R C**。**C**语言开始流行,成为最广泛使用的几种计算机语言之一。

1.3 C语言的发展及其特点

- **1983**年，美国国家标准协会(**ANSI**)成立了一个委员会，根据**C**语言问世以来各种版本对**C**语言的发展和扩充，制定了第一个**C**语言标准草案(**'83 ANSI C**)。

1.3 C语言的发展及其特点

- **1989年，ANSI公布了一个完整的C语言标准—ANSI X3.159-1989(常称ANSI C，或C89)。**

1.3 C语言的发展及其特点

- **1990年**，国际标准化组织
ISO(International Standard Organization) 接受**C89**作为国际标准
ISO/IEC 9899:1990，它和
ANSI的C89基本上是相同的。

1.3 C语言的发展及其特点

- **1995年，ISO对C90作了一些修订，1999年，ISO又对C语言标准进行修订，在基本保留原来的C语言特征的基础上，针对应用的需要，增加了一些功能，尤其是C++中的一些功能，命名为ISO/IEC 9899:1999。**

1.3 C语言的发展及其特点

- **2001、2004**年先后进行了两次技术修正（**TC1**和**TC2**）。
ISO/IEC 9899:1999(及其技术修正)被称为 **C99**。
 - ◆ **C99**是**C89**(及**1995**基准增补**1**)的扩充。
- **2011**年，**ISO**更新了**C**语言标准为**ISO/IEC:9899:2011**,
此标准在**2011**年底被**ANSI**采纳。
- **2018**年，**ISO**更新了**C**语言标准为**ISO/IEC:9899:2018**
- 未来版标准：**ISO/IEC:9899:202x**

1.3 C语言的发展及其特点

- 课堂示例均以**C99/C11**标准为依据。
- 目前不同软件公司提供的各**C**语言编译系统大多未完全实现**C99/C11**建议的功能
- 本课程的示例程序基本上都可以在目前所用的编译系统(如**MSVC, GCC**)上编译和运行(课堂示例采用版本: **GCC8.3.1**)。

1.3 C语言的发展及其特点

- **C语言**是一种用途广泛、功能强大、使用灵活的过程性(**procedural**)编程语言，既可用于编写应用软件，又能用于编写系统软件。因此**C语言**问世以后便得到迅速推广。

1.3 C语言的发展及其特点

➤ C语言主要特点:

- ◆语言简洁、紧凑，使用方便、灵活。

- 只有**44**个关键字、**9**种控制语句

- 程序书写形式自由，源程序短

C语言中的关键字（C11标准）

Auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
inline	int	long	register
restrict	return	short	signed
sizeof	static	struct	switch
typedef	union	unsigned	void
volatile	while	_Alignas	_Alignof
_Atomic	_Bool	_Complex	_Generic
_Imaginary	_Noreturn	_Static_assert	_Thread_local

1.3 C语言的发展及其特点

➤ C语言主要特点:

◆运算符丰富。

- 有**37**种运算符
- 把括号、赋值、强制类型转换等都作为运算符处理
- 表达式类型多样化

1.3 C语言的发展及其特点

➤ C语言主要特点:

◆数据类型丰富。

- 包括:整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型
- C99又扩充了复数浮点类型 (**_Complex**, **_Imaginary**)、超长整型(**long long**)、布尔类型(**_Bool**)
- 指针类型数据, 能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。

1.3 C语言的发展及其特点

➤ C语言主要特点:

◆ 具有结构化的控制语句

- 如**if...else**语句、**while**语句、**do...while**语句、**switch**语句、**for**语句
- 用函数作为程序的模块单位，便于实现程序的模块化
- C语言是完全模块化和结构化的语言

1.3 C语言的发展及其特点

➤ C语言主要特点:

- ◆语法限制不太严格，程序设计自由度大。

- 对数组下标越界不做检查

- 对变量的类型使用比较灵活，例如，整型量与字符型数据可以通用

- C语言允许程序编写者有较大的自由度，因此放宽了语法检查

1.3 C语言的发展及其特点

➤ C语言主要特点:

- ◆允许直接访问物理地址，能进行位操作，可以直接对硬件进行操作

- C语言具有高级语言的功能和低级语言的许多功能，可用来编写系统软件

- 这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言

1.3 C语言的发展及其特点

➤ C语言主要特点:

- ◆用C语言编写的程序可移植性好。

- C的编译系统简洁，很容易移植到新系统

- 在新系统上运行时，可直接编译“标准链接库”中的大部分功能，不需要修改源代码

- 几乎所有计算机系统都可以使用C语言

- ◆生成目标代码质量高，程序执行效率高。

1.4最简单的C语言程序

1.4.1 最简单的C语言程序举例

1.4.2 C语言程序的结构

1.4.1 最简单的C语言程序举例

例**1.1** 要求在屏幕上输出以下一行信息。

This is my first C program.

➤ 解题思路：

在主函数中用**printf**函数原样输出以上文字。

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

C程序必须有一个 main 函数

```
{
```

函数的名字，表示主函数

```
printf ("This is my first C program.\n");
```

```
return 0;
```

```
}
```

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

主函数类型

```
    printf ("This is my first C program.\n");
```

```
    return 0;
```

```
}
```

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    printf ("This is my first C program.\n");
```

```
    return 0;
```

```
}
```



函数体

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

输出函数

```
printf ("This is my first C program.\n");
```

```
return 0;
```

```
}
```

输出语句

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{           This is my first C program!
```

```
printf ("This is my first C program.\n");
```

```
return 0;
```

```
}
```

输出语句

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{           This is my first C program!
```

```
    printf ("This is my first C program.\n");
```

```
    return 0;
```

```
}
```

换行符

1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    printf ("This is my first C program.\n");
```

```
    return 0;
```

```
}
```

当main函数执行结束前
将整数0作为函数值

1.4.1 最简单的C语言程序举例

#include <stdio.h>

int main()

{

用到函数库中的输入输出函数时

printf ("This is my first C program.\n");

return 0;

}

表示语句结束

1.4.1 最简单的C语言程序举例

C语言允许用两种注释方式:

➤ **//**: 单行注释或

- ◆可单独占一行

- ◆可出现在一行中其他内容的右侧

➤ **/*.....*/**: 块式注释

- ◆可包含多行

- ◆也可以是行中的一部分

1.4.2 C语言程序的结构

C语言程序的结构特点：

1. 一个程序由一个或多个文件组成

◆ 文件类型：头文件(.h)和源文件(.c)

◆ 小程序往往只包括一个源文件

1.4.2 C语言程序的结构

C语言程序的结构特点：

➤ 一个源文件一般包括三个部分：

- ◆ 预处理指令 `#include <stdio.h>`等，可以缺省
- ◆ 全局声明 在函数之外进行的声明，包括数据声明和函数声明等，可以缺省
- ◆ 函数定义 每个函数用来实现一定的功能，可以缺省

1.4.2 C语言程序的结构

C语言程序的结构特点：

2. 函数是C程序的主要组成部分

- ◆ 一个C程序是由一个或多个函数组成的
- ◆ 必须包含一个**main**函数（只能有一个）
- ◆ 每个函数都用来实现一个特定功能
- ◆ 被调用的函数可以是库函数(**printf**)，也可以是自己设计的函数(第七章讲)

1.4.2 C语言程序的结构

C语言程序的结构特点：

3. 程序的运行总是从main函数开始执行，在main函数结束

4. C程序对计算机的操作由C语句完成

◆C程序书写格式是比较自由的

- 一行内可以写多个语句

- 一个语句可以分写在多行上

◆为清晰起见，习惯上每行只写一个语句

1.4.2 C语言程序的结构

C语言程序的结构特点：

- 3. 程序总是从**main**函数开始执行
- 4. **C**程序对计算机的操作由**C**语句完成
- 5. **C**程序语句最后必须有分号
- 6. **C**语言本身不提供输入输出语句
- 7. 程序应当包含注释，增加可读性
- 8. 程序块应使用缩进，缩进的空格数为**4**个

1.5 运行C程序的步骤与方法

1. 上机输入和编辑源程序（**.c**文件）
2. 对源程序进行编译（**.o/.obj**文件，随编译器不同）
3. 进行连接处理（生成可执行文件（**.exe**文件，随编译器操作系统不同））
4. 运行可执行程序，得到运行结果

1.6 程序设计的任务

1. 问题分析

- 对于接手的任务要进行认真的分析
- 研究所给定的条件
- 分析最后应达到的目标
- 找出解决问题的规律
- 选择解题的方法

1.6 程序设计的任务

1.问题分析

2.设计算法

➤设计出解题的方法和具体步骤

1.6 程序设计的任务

1.问题分析

2.设计算法

3.编写程序

4.对源程序进行编辑、编译和连接

5.运行程序，分析结果

◆结果错了，程序肯定错

◆结果对了，程序未必对

1.6 程序设计的任务

- 1. 问题分析**
- 2. 设计算法**
- 3. 编写程序**
- 4. 对源程序进行编辑、编译和连接**
- 5. 运行程序，分析结果**
- 6. 编写程序文档**