

18-19学年秋季学期计算概论课后辅导讲义Lecture 1

陈婉雯

2018年9月28日

1 本讲主题

- 为什么学习C语言：C语言的历史、特点
- 如何编程：编程的步骤与IDE
- 新手常见问题及注意事项

2 C语言简介

C语言是一种通用的、面向过程式的计算机程序设计语言。1972年，为了移植与开发UNIX操作系统，丹尼斯·里奇在贝尔电话实验室设计开发了C语言。C语言是一种广泛使用的计算机语言，在现代软件程序员之间都得到广泛使用，也是许多其他编程语言的基础。

为什么学习C语言：

- C语言是一种简洁但强大的语言：C语言的核心内容很少，只有32个关键字，9种控制语句，但有44种运算符和丰富的数据类型。
- C语言是一种面向过程的语言：分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用。相比与其他面向对象的语言（比如C++），C更容易上手，也能让学习者学到基础编程思维，是一种很好的入门语言。
- C语言是一种具有低级语言特点的高级语言：C语言具有部分与汇编语言相近的功能和描述方法，如地址运算、二进制数位运算、操作硬件端口等。
- C语言是一种结构化语言：C语言以函数为基本单位，通过参数传递数据，实现模块化设计。同时，操作系统提供了大量库函数供用户使用。
- 可移植性好：程序可以从一个环境不经改动或稍加改动就可移植到另一个完全不同的环境中运行。这是因为系统库函数和预处理程序将可能出现的与机器有关的因素与源程序隔开，这就容易在不同的C编译系统之间重新定义有关内容。
- 运行效率高。
- 语法限制相对不严：没有数组下标越界检查、变量类型兼容性检查、允许大量的数据类型转换、可以自定义新类型等。当然，这也会导致新手容易犯一些不易发现的错误。

3 编程七步曲

编程通常可以分为七步：

- 定义程序目标：在动手写之前一定要在脑中有清晰的思路：程序要实现什么功能，需要什么信息，要报告什么信息，要进行什么计算和控制。
- 设计程序：考虑如何用程序完成任务。程序如何组织？如何处理数据？用户界面设计成什么样？
- 编写代码：在IDE的编辑器里写代码，把你设计的程序翻译成C语言。
- 编译：把高级语言变成机器语言与生成可执行程序。在这一过程中，编译器也会检查程序是否有误并给予反馈。如果编译器发现错误，就不生成可执行文件并报错。要学会看编译器的错误报告和警告信息。
- 运行：计算机运行生成的可执行程序。
- 测试和调试：用不同的测试数据检验程序是否按照你的思路运行。查找并修复程序错误（bug）的过程就叫做调试（debug）。主要方法是对错误进行定位，确定错误产生的原因，对程序错误予以改正，重新测试。
- 维护和修改：根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改。

其中，第一和第二步主要是思考程序算法（可以通过伪代码、算法流程图等方式辅助实现），其后的步骤需要在IDE 中完成。本课程基本不涉及维护和修改程序。本课程的主要难点在于设计程序与测试、调试程序。

3.1 设计程序

很多人在设计程序过程中遇到的难点在于培养一种编程的思维：**将问题划分为子问题，并按步骤解决问题**。这个问题会在以后的课程中继续讲解。在设计程序的过程中，我们主要考虑三个问题：如何表示数据、如何存储数据与如何处理数据。

- 表示数据：我需要什么数据？
- 存储数据：这些数据该存储为什么类型，该用多大的存储空间¹？
- 处理数据：我要对这些数据进行什么操作？

在设计程序的过程中，我们可以用算法流程图、伪代码等方式来帮助我们理清思路。广义地说，为解决一个问题而采取的方法和步骤，称为“算法”。当你能够把你解决问题的方法写成按步骤执行的过程，你离成功就不远了。

- 算法流程图：用一些图框来表示各种类型的操作，在框内写出各个步骤，然后用带箭头的线把它们连接起来，以表示执行的先后顺序。用图形表示算法，直观形象，易于理解。当你的程序变得更加复杂的时候，可以通过流程图表现不同模块的功能与参数传递关系，更容易理清你的思路。

¹大约在你们学了指针之后开始考虑存储空间大小的问题。目前在定义变量的时候C就会自动给你开辟好合适的存储空间，只需要注意你定义的数据类型和你的数据相符就可以。

- 伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章一样，自上而下地写下来。每一行（或几行）表示一个基本操作。它不用图形符号，因此书写方便、格式紧凑，易懂，也便于向计算机语言算法（即程序）过渡。你可以用英文、汉字、中英文混合表示算法，并无固定的、严格的语法规则，只要把意思表达清楚，书写的格式清晰易读即可。

3.2 编写程序

想必同学们在努力完成作业的过程中已经体会到了编写程序的感觉。接下来简要介绍一下我们的编程环境。

3.2.1 IDE

IDE：集成开发环境（Integrated Development Environment）是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具。集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。

常用的IDE：Visual Studio、Visual C++ Express、Dev C++

在IDE内我们主要使用”解决方案管理器“来管理我们的文件。文件主要是头文件和源文件。
头文件：

- 扩展名为.h 的文件，通常分为程序员编写的头文件和编译器自带的头文件。头文件可以定义所用的函数列表，方便查阅你可以调用的函数；头文件可以定义很多宏定义，就是一些全局静态变量的定义，在这样的情况下，只要修改头文件的内容，程序就可以做相应的修改，不用亲自跑到繁琐的代码内去搜索。调用头文件相当于复制头文件的内容，赋予了调用某些函数的权限。
- 头文件只是声明，不占内存空间，要知道其执行过程，要看你头文件所申明的函数是在哪个.c文件里定义的，
- 建议把所有的常量、系统全局变量和函数原型写在头文件中，在需要的时候随时引用这些头文件。
- 引用语法：

`#include <file>` 引用编译器自带头文件：输入输出stdio.h, 内存管理stdlib.h, 复杂数学运算math.h, 字符串操作string.h, ……

`#include "file"` 引用自己编写的头文件。

源文件：

- 扩展名为.c（C语言）或.cpp（C++）的文件。
- 实际编写函数过程。我们一般采用函数声明和定义分离的方式：把所有的声明都放在头文件中，把所有的定义放在源文件中。

3.2.2 编程机制

C编程的过程是用程序把源代码文件转换成可执行文件（包含可直接运行的机器语言代码）。典型的C实现通过编译和链接两个步骤来完成。编译器把源代码转换成中间代码，链接器把中间代码和其他代码合并，生成可执行文件。

在计算机底层硬件中，所有的数据都是以1和0两个高低电平来表示，计算机只能识别这两个电平。编程语言类似于人类语言，我们很容易就能理解它的意思，编写代码的效率非常高。编译器能够将源代码（人类语言）翻译成目标代码（0&1语言），便于人类和计算机的沟通。我们叫存储着目标代码的文件为目标代码文件（或简称为目标文件）。

但这个文件还不能直接运行。目标代码文件缺失启动代码（程序和操作系统之间的接口）和库函数。链接器会把你目标代码、系统的标准启动代码和库代码合并成一个可执行文件。

3.2.3 程序结构

C程序主要包括：

- 预处理指令：以#开头，写在文件开头。比较常用的预处理指令：

 #include 包含一个源代码文件

 #define, #undef 定义，取消定义宏²

 #if 如果给定条件为真，则编译以下代码

- 函数：程序的执行指令，C语言的核心。整段程序从主函数（main）开始执行，遇到函数时会通过参数传递的方式执行函数，获得函数返回值。
- 变量：程序可操作的存储区的名称。C 中每个变量都有特定的类型，类型决定了变量存储的大小和布局，该范围内的值都可以存储在内存中，运算符可应用于变量上。变量定义就是告诉编译器在何处创建变量的存储，以及如何创建变量的存储。
- 语句、表达式：

 语句指的是当程序运行时执行某个动作的语法结构。它改变变量的值，产生输出，或处理输入。语句可分为：表达式语句(x=y+z)、函数调用语句(printf("Hello, world!"))、控制语句（条件判断语句、循环执行语句、转向语句）、复合语句（把多个语句用括号括起来组成的一个语句）、空语句（只有分号“;”组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句可用来作空循环体）。

 表达式一种有值的语法结构，它由运算符将变量、常量、函数调用返回值结合而成。

- 注释：不执行的内容，用于提供有关代码行的信息。单行注释由“//”表示，多行注释由“/*...*/”表示。

同学们可以参考PPT中的程序示例来学习这几部分。

²宏是一种预处理指令。它提供了一种机制，可以用来替换源代码中的字符串。例如：#define VERSION_STAMP 1 中所定义的这种形式的宏通常被称为标识符。在编译预处理时，源代码中的每个VERSION_STAMP标识符都将被数值1替换掉。宏在计算概论中用的不多，在此仅供了解。

3.2.4 养成良好编程习惯

- 命名规范：要体现变量、函数的含义，直观可读；不要出现仅靠大小写区分的标识符；变量用形容词+ 名词；函数用动词+ 名词；常量全部大写，变量小写，单字符一般用在局部变量/循环。

下划线命名法：单词间用下划线分开，如`print_employee_paychecks()`

驼峰命名法：开头小写，单词间通过大小写分开，如`printEmployeePaychecks()`

帕斯卡命名法：开头大写，单词间通过大小写分开，如`PrintEmployeePaychecks()`

目前比较主流的方式是变量用驼峰命名法，函数用帕斯卡命名法。

- 换行与缩进：要保证能够看清循环、选择等逻辑结构；缩进用两个空格或者Tab（编译器会自动缩进，保证缩进在同一个模块内对齐即可）；括号要对齐。

- 注释：

在开头写程序简介、作者信息。

每个函数要标注函数的作用、输入和输出的含义。

在重要的变量和步骤旁做一下标注、参考来源（如有）。

在编写复杂程序时，可以用文本文档写技术文档，详细写自己的设计思路、参考文献、模块作用等。

最好养成用英文写注释的习惯。

3.3 运行、调试程序

一般IDE会提供两种运行方式：调试程序与仅运行（不调试）。以下说法以Visual Studio为例。

- 启动调试：可以查看程序断点处的变量值，方便寻找程序过程中的错误。当程序运行到断点的时候，它会暂时停止运行后面的语句，供用户观察程序的运行情况（比如变量当前的值、指针地址等），并等待用户发出指令。我们可以让语句一句句地执行。如果执行到某一句语句时发生了运行时错误，那么这个错误一般就是由这个语句引起的。我们还可以观察每一句语句执行的顺序和执行后变量变化的情况。如果发现程序无法实现既定的功能，我们可以将期望的结果和实际的结果作比对，并分析可能引起这些不同的原因。这样一来，大大加快了我們找到问题和解决问题的速度。学会用断点调试来debug是非常重要的技能，希望大家可以多尝试。
- 开始执行（不调试）：直接运行程序。
- 我个人的习惯是先调试，调试没问题之后再执行。执行无法通过之后再进行断点调试。

通常来说调试过程会遇到两种情况：

- 编译未通过：无法生成可执行文件，显示出错误报告。

按顺序查看错误说明（会显示在哪一行出现什么错误）并修改，**从上往下看错误说明，每改一条就再编译一次。**

遇到看不懂的说明就去百度/谷歌，总有人会和你错的一样的。

- 编译通过但结果不对：程序可以运行，但不能输出想要的结果。

用断点调试（推荐）或在控制台中直接输出（`printf`）变量的值（可用，但不推荐）等方法检查程序运行过程中变量的值是否有误。也可以逐行运行代码检查代码运行顺序和预想的是否不同。

4 新手常见错误

- 为什么程序读不了我输入的数？

检查你输入格式和你预设的格式是否完全相同，尤其注意中英文标点符号、换行和空格。

- 为什么总是编译不成功？

检查括号数量是不是对上了；

检查分号有没有打对（不要打成中文符号，不要打成冒号）；

检查你前后变量名有没有打错。

- 其他常见问题：

不要弄混赋值（`=`）和判断等于（`==`）；

循环过程逻辑容易乱，可以通过手动运算或者断点调试等方法查看循环里面有没有问题；

在选择过程中没有考虑全所有可能性，可以通过更改测试数据找到问题。

5 可供参考的学习网站

Leetcode：可以刷题。

stackoverflow：是一个程序设计领域的问答网站，回答质量较高，使用英语。

CSDN：中文博客网站，可以找到很多总结、经验等，但质量参差不齐。

github：开源程序平台，可以在上面找到很多开源代码。目前用处不大，但你们可以试着读一下别人怎么写代码，学习一下代码规范（要看那些收藏比较多的）。以后在进一步学习、科研时能在上面找到很多现成的工具箱。