教育部大学计算机课程改革项目规划教材

C语言教程

主编 陈良银 游洪跃 李旭伟

高等教育出版社

教育部大学计算机课程改革项目规划教材

C语言教程

主编　陈良银　游洪跃　李旭伟

高等教育出版社·北京

内容提要

本书作者从事C语言教学已有十余年，但从未觉得C语言教学是一件轻松的事，在总结多年教学经验基础上萌生了写一本适用于初学者的最易教、最易学的C语言教材的想法。

为了更适合教学，作者首先在结构上做了新的调整，让每一章的结构变得简单，且每章仅包含三部分内容，即：1. 程序示例，2.相关语法，3.研究部分。其次，本书在内容上也做了大量简化，重点关注那20%左右的编程常用语言要素。本书的内容主要包括：基础知识、表达式求值、C程序控制结构、函数、数组和指针、结构类型、预处理命令和文件。最后本书还以附录的形式增加了高级主题部分。

本书可作为初学者的自学教材，或大一新生学习C语言编程的核心教材或参考资料。也可供自学人员参考。

图书在版编目（CIP）数据

C语言教程/陈良银，游洪跃，李旭伟主编. --北京：高等教育出版社，2018.9

ISBN 978-7-04-050392-0

Ⅰ. ①C… Ⅱ. ①陈… ②游… ③李… Ⅲ. ①C语言-程序设计-高等学校-教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字（2018）第178012号

C Yuyan Jiaocheng

策划编辑 刘 娟　　责任编辑 刘 娟　　封面设计 李卫青　　版式设计 马 云

插图绘制 于 博　　责任校对 张 薇　　责任印制 刘思涵

出版发行	高等教育出版社	网　址	http://www.hep.edu.cn
社　址	北京市西城区德外大街4号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印　刷	河北鹏盛贤印刷有限公司		http://www.hepmall.com
开　本	850mm×1168mm　1/16		http://www.hepmall.cn
印　张	17.25		
字　数	410千字	版　次	2018年9月第1版
购书热线	010-58581118	印　次	2018年9月第1次印刷
咨询电话	400-810-0598	定　价	38.00元

物 料 号　50392-00

C语言教程

陈良银　游洪跃

李旭伟

1　计算机访问http://abook.hep.com.cn/1865450或手机扫描二维码、下载并安装Abook应用。

2　注册并登录，进入“我的课程”。

3　输入封底数字课程账号（20位密码，刮开涂层可见），或通过Abook应用扫描封底数字课程账号二维码，完成课程绑定。

4　点击“进入学习”，开始本数字课程的学习。

课程绑定后一年为数字课程使用有效期。受硬件限制，部分内容无法在手机端显示，请按提示通过计算机访问学习。

如有账号问题，请发邮件至：abook@hep.com.cn。

http://abook.hep.com.cn/1865450

前　言

本书作者在从事C语言教学工作十多年以来，从未觉得C语言教学是一件轻松的事情。每一届总会有不少学生感叹“C语言太难学了!”。因此，我们萌生了写一本适用于初学者的最易教、最易学的C语言教材的想法。

为了避免让学习者过早陷入学习C语言烦琐的语法细节、及轻视上机训练，本书采用了全新的组织结构，即本书的主要章节都包含3个部分：程序示例、相关语法、研究部分。下面对各部分加以介绍。

第1部分是程序示例，其注释和讲解都非常详细、透彻，作者的目标是——让完全没有基础的人也能轻松看懂程序。这部分程序示例是学习的重点，只要学透其语法要点、并会模仿编程，那样读者就会慢慢建立起对自己编程的自信，教学的最大目的应该是培养、增强学习者的自信心。有信心就可能有一切；而没有信心就会放弃努力，结果可想而知。

第2部分是对相关语法的详细讲解，当学生在上机编程实践时遇到C语言问题时可在此部分查阅相关语法，通过编程实践学习C语言及C语言的相关语法是最高效的学习方法。

第3部分作为研究部分，包括程序陷阱和实例研究两部分内容，程序陷阱部分包含了在实际编程时容易出现的问题，也包括正文内容的深入讨论，这部分内容不管对初学者还是对那些长期编程的人都很有用。实例研究部分主要提供给那些精力充沛的学生深入学习与研究，这些实例包括读者深入学习时可能会遇到的问题（例如2.3.2节中的“用贪心算法高效求解纸币支付问题”，5.3.2节中的“编程的艺术——求解兑换钱问题”），也包括应用所学知识解决实际问题（例如8.3.2节中的“人事管理系统”），还包括容易引起读者兴趣的问题（例如3.3.2节中的“暴力破解数字谜语”，7.3.2节中的“碰运气游戏”），通过读者对实例研究的学习，可提高实际应用C语言程序设计的能力，当然有一定的难度，但应比读者的想象更易学习与掌握。

因此，作者给教学者的建议是精讲第1部分，选讲第2部分，不讲第3部分。作者给学习者的建议是通过反复上机训练吃透第1部分，之后再自学第2部分，并适当研究第3部分。

注：本书没加有星号“*”及双星号“**”的部分是基本内容，适合所有读者学习；加有星号“*”的部分适合计算机专业的读者根据情况研读；加有双星号“**”的部分适合于感兴趣的同学研究，尤其适合于那些有兴趣参加ACM竞赛的读者加以深入研究。

本书提供大量配套的资源，包括电子教案、源代码、部分习题参考答案等，老师可在Abook

网站上下载获得。

全书由陈良银教授、游洪跃副教授、李旭伟副教授执笔统稿，由陈良银、游洪跃、李旭伟任主编，张磊、袁平、卓碧华、殷锋、陈彦如、郭敏任参编。最后对有关专家和老师提供的帮助和支持表示感谢！由于作者的水平有限，同时本书也是一种新的探索，书中难免存在不妥之处，恳请广大读者不吝赐教。

编　者

2018年6月

目　录

第 1 章　基础知识

电子教案 1-1：基础知识

本章试图说明白 C 程序的构建原理。作者认为首先要理解 C 程序的宏观结构——C 程序就是由多个功能模块（像积木块一样，称之为“特制砖头”）通过特定的组合搭建而成的。学习 C 程序的视角首先要放在类似“通过搭积木构建童话中的城堡”这种宏观思路上，其次才放在每一个积木块的具体制作过程上。一个积木块就可以理解为一个 C 函数。可见“通过函数构建 C 程序”是多么核心关键的思想。而现实中，人们往往并未关注“用积木搭建城堡”这个环节，而是纠结于“制作积木”的各种具体技巧（语法难点）上。作者觉得这有点本末倒置。因此，特别提醒学习者首先意识到这个问题的重要性。

1.1　C 程序的“特制砖头”结构

1.1.1　积木拼图游戏

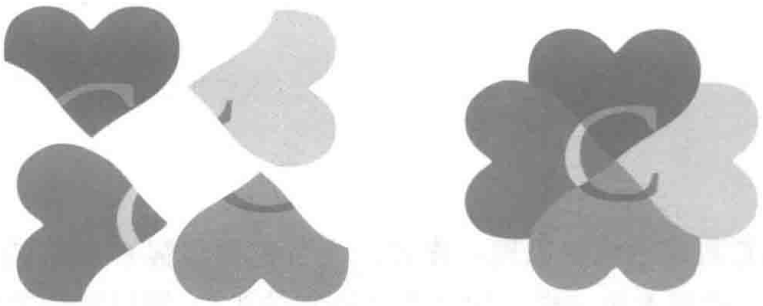
在方法论中，小目标和大目标谁更重要一点？这个问题其实很不容易回答。实际上大、小目标都很重要，在目标和战略规划层面，设定大目标更重要；在具体实施和执行层面，一步步实现小目标更重要。作者认为，该方法论也许适用于可以拆分的任何事项。

在编制 C 程序这件事上，这一方法论就特别有用，即先必须胸有大任务，并把该大任务划分成多个小任务分别予以实现，最后再将这多个小任务组合起来完成大任务。

图 1.1 示意了一个简单的积木拼图游戏：假定有如图 1.1（a）所示的 4 块积木块，请将它们摆放到适当的位置从而得到一个有“C”字的拼图图案。只要稍加分析即可拼出如图 1.1（b）所示的 C 图案。在这个游戏中，已经胸有 C 图案，并有图案各异的 4 个小积木块，只要把 4 个小积木块按照特定顺序组合在一起就可得到 C 图案。因此，在这个游戏中，肯定更关心得到 C 图案的具体组合方式。

假定把最终得到的 C 图案视为一个 C 程序的任务，把 4 块各有特点的小积木块视为 4 个不同的功能函数。显然，可以认为一个 C 程序任务就是按照特定顺序调用具有不同功能的函数而实现的。相对来说，更关心某小积木块承载了什么功能，应该摆放在什么位置才能发挥

它的作用获得整体图案，可能并不特别关心这个小积木块是如何制作的。也就是说，更关心某函数具有什么功能，怎样才能调用它得到这些功能为整体任务服务，而不关心该函数是如何实现的。就像上面的 4 个积木块，别人已经做好了，拿来按照特定顺序摆放在一起即可。没有必要关心它的具体制作方法，也没有必要自己去制作它们。但假定缺了其中某块小积木，那么就得自己动手制作它，不然就得不到 C 图案。



(a) 4 块积木块　　(b) C 图案

图 1.1　积木拼图游戏

可见，学 C 编程一定要先有森林——大目标，但是，如果没有一个个的小目标——树木，森林也将不可得。如果没有森林，那树木的意义也就大大减少了。总体来说，宏观和微观都不可缺少，但首先必须关注宏观。

1.1.2 “特制砖头”在 C 程序中的作用

学习 C 程序一定得先有宏观概念，即知晓 C 程序是由一个或多个“特制砖头”垒成的。而这里所谓的“特制砖头”就是 C 语言的函数。函数应该具有一定的功能。函数是构成 C 程序的基本单位。显然，把具有不同功能的函数（不同的“特制砖头”）垒在一起，就可能得到一个组合了这些功能的程序。而已经被别人实现的函数可以视其为一个透明体，无须关心其内部实现。所以，学习编程的第一步是首先养成把一个稍大的功能细分成多个较小的功能并分别予以实现的习惯和思想。其次，了解“特制砖头”即函数的模样，知晓如何编制一个简单函数——制作“特制砖头”（实现小目标）。任何时候，都要把函数当成一个整体对待，即函数是一个不可分割的有机整体。最后，懂得如何把不同功能的函数垒在一起，即学习函数声明、函数定义、函数调用方法（用小目标去支撑大目标）。

1．首先将程序整体功能划分为多个易于实现的小功能

比如，假定要建一个微信公众号作为大学生学习生活的有力助手，那么要做的第一步就是对其具体功能进行了解、分块。假定将该微信公众号取名为“易书桥”，它的主要功能被分成以下四大块（这里只介绍粗略分块，实际工程实现时还需继续细分）。

- A( )多彩社团筛选：针对社团管理者提供社团招新管理，社团活动推介，社团人员管理等功能；针对社团团员提供社团介绍，心仪、优质社团筛选，申请加入社团等功能。
- B( )书籍书房管理：记录阅读过的全部书籍的相关信息；随时对曾经阅读过的书进行评

论；记录、管理、分享书评；提供读书计划，并可将其分享给书友，以便相互监督，结交朋友，共同提升读书效果。

• C( )个性课表管理：通过登录教务处网站获取个人的准确选课、课程表和成绩等信息，同时还可以永久记录个人的全部课业信息，以供查阅；在未来进行择业评估时，这些信息将有助于提升评估的可信度。

• D( )历年试题集锦：平台将逐步搜集部分专业课程的考试题或模拟题（部分提供答案），这将对大家的期末复习有很强的指示性，平台也鼓励大家分享自己的专业考题以便帮助他人并获得积分，平台还计划搜集部分考研试题，以方便大家准确把握常见考试题型和考试重点。

假定将上述每一功能块都视为一个简单功能，且一个函数就能够实现一个这样的简单功能，那么就只需要编写完成这些小功能所对应的名为 A( )、B( )、C( )、D( )的函数（“特制砖头”）就可以了。图 1.2 就显示了该微信公众号相关功能模块的组成结构。从图 1.2 可以看到：易书桥的功能模块被分成了 4 个模块，假定这些功能由 4 个函数分别予以实现，那么每一个函数就是一个“特制砖头”。其中 main( )函数用于把 A( )、B( )、C( )、D( )各函数组合在一起实现整体功能。A( )、B( )、C( )、D( )函数则各自实现了自己的模块功能。大家一定要注意到这 4 个函数中的每一个函数都是一个独立整体，每一个函数都只能被当做一个整体看待。也就是说要用就用整块“特制砖头”，不能把任何一块“特制砖头”拆成两半使用。除非本来就把它设计成两个函数，否则不能分割任何一个函数。

2.“特制砖头”即函数的模样

每一个函数都有其特定的功能，因此可以认为它是特制的。那么，如何特制一个函数呢？其实，函数的模样都是固定的，这里的特制重点是讲函数都有特定的功能，并通过相关的 C 语句去实现这些功能。C 语言语法规定：一个函数定义必须有函数头和函数体两部分，这就是函数的模样。也就是说函数就长成这个样子，以后编写函数就按照这个样子写就可以了。所谓的函数头重点就是函数的名字，并在函数名后面的一对小括弧中说明函数的形式参数，在函数名的前面说明函数的返回值类型等。函数体其实就是函数头后面的一对大括弧之间的全部内容，即实现函数功能的相关代码。比如，以下就是一个完整的 C 语言函数：A( )函数。

图 1.2 程序的“特制砖头”结构

```
void A(void)     /*A( )函数的函数头，第 1 个 void 表示它没有返回值，第 2 个表示它没有参数*/
{                /* {}之间的全部内容就是 A( )函数的函数体，A 则是该函数的函数名 */
   printf("多彩社团筛选");/* 调用 printf( )函数，输出“多彩社团筛选” */
}
```

假定 A( )函数实现了多彩社团筛选的全部功能。只需要调用它就能够获得它所实现的全部功能。

3．把函数垒在一起的方法，即函数调用方法

一个程序必须有一个main( )函数，main( )的形式如下：

```
int main(void)  /* main( )函数的函数头，int 表示返回值为整型值，void 表示没有参数*/
{               /* {}之间的全部内容就是main( )函数的函数体 */
    …;          /* 各种符合C语言语法规则的语句 */
    return 0;   /* 返回语句，向系统返回0 */
}
```

可以在main( )函数中调用其他函数功能，把其他函数功能集成到main( )函数，这样就可以把多个不同函数功能垒在一起，形成整体功能。比如，在上述微信公众号中，就可以在main( )函数中通过调用A( )、B( )、C( )、D( )函数，从而实现“易书桥”的整体功能。当然，不止main( )函数可以调用其他函数，其他函数间也可以相互调用，比如，A( )函数就调用了系统库函数printf( )（注：库函数也是函数，只是它由编译器提供，不用用户自己定义）。事实上，函数调用还涉及参数匹配等诸多问题，但目前只需要了解通过函数调用可以实现功能集成即可。假设易书桥程序结构（yishuqiao.c）如下。

源代码1-1：易书桥示例程序

```
/**
 * 文件名：yishuqiao.c，程序的“特制砖头”结构
 */

/* 1. 编译预处理命令和函数声明 */
#include <stdio.h>                  /* 包含预处理命令，printf( )函数在该文件中声明 */
void    A(void );                   /* A( )函数的函数声明，因为它的调用在定义之前 */
void    B(void );                   /* B( )函数的函数声明，因为它的调用在定义之前 */
void    C(void );                   /* C( )函数的函数声明，因为它的调用在定义之前 */
void    D(void );                   /* D( )函数的函数声明，因为它的调用在定义之前 */


/* 2. main( )函数的定义——“特制砖头”1 */
int main(void)                      /* main( )函数函数头，int表示返回值为整型值，void
                                       表示函数没有参数*/
{                                   /* {}之间的全部内容就是main( )函数的函数体 */
    printf("易书桥具有以下功能：\n");/* 调用printf( )函数，输出“易书桥具有以下功能:” */
    A( );                           /* 调用A( )函数功能 */
    B( );                           /* 调用B( )函数功能 */
    C( );                           /* 调用C( )函数功能 */
    D( );                           /* 调用D( )函数功能 */
    return 0;                       /* 返回语句，向系统返回0 */
}


/* 3. A( )函数的定义——“特制砖头”2 */
```

```
void A(void)                          /*A()函数的函数头，第1个void表示它没有返回值，
                                        第2个表示它没有参数*/
{                                     /* {}之间的全部内容就是A()函数的函数体 */
    printf("多彩社团筛选;\n");        /*调用printf()函数，输出“多彩社团筛选；”*/
}


/* 4. B()函数的定义——“特制砖头”3 */
void B(void)                          /*B()函数的函数头，第1个void表示它没有返回值，
                                        第2个表示它没有参数*/
{                                     /* {}之间的全部内容就是B()函数的函数体 */
    printf("书籍书房管理;\n");        /* 调用 printf()函数，输出“书籍书房管理：”*/
}


/* 5. C()函数的定义——“特制砖头”4 */
void C(void)                          /*C()函数的函数头，第1个void表示它没有返回值，
                                        第2个表示它没有参数*/
{                                     /* {}之间的全部内容就是C()函数的函数体 */
    printf("个性课表管理;\n");        /* 调用 printf()函数，输出“个性课表管理：”*/
}


/* 6. D()函数的定义——“特制砖头”5 */
void D(void)                          /*D()函数的函数头，第1个void表示它没有返回值，
                                        第2个表示它没有参数*/
{                                     /* {}之间的全部内容就是D()函数的函数体 */
    printf("历年试题集锦;\n");        /* 调用printf()函数，输出“历年试题集锦：”*/
}
```

注：程序 yisuqiao.c 中位于“/*”和“*/”之间的内容全部是注释。C99 规定“//”后面的内容也是注释（仅限于同行，不允许跨行）。

1.1.3 C 程序的运行过程

将上述程序的文件命名为 yishuqiao.c，它属于源文件，通过文字编辑工具可以编辑它。通常，运行一个程序需要以下 4 步，即编辑、编译、连接、运行。编辑就是通过编辑工具得到源文件的过程（源文件名称的扩展名为 c，比如，yishuqiao.c）；编译就是利用编译器将源文件翻译为机器能够识别的目标文件的过程（目标文件名称的扩展名为 obj，比如，yishuqiao.obj）;连接就是将多个与程序相关的目标文件组装在一起，得到可执行文件的过程（可执行文件名称的扩展名为 exe，比如，yishuqiao.exe）；最后运行可执行文件，得到程序的设定功能。

通常 C 程序开发工具会集成从程序编辑、编译、连接到运行的全部管理软件。市面上的 C 程序开发工具很多，程序员只需要熟悉其中一种开发工具即可完成对程序的编辑、编译、

连接和运行工作。教学中用得比较多的开发工具包括 VC++ 6.0、Dev-C++等等。

练习：查阅资料，选一种 C 程序开发工具，熟悉其界面，并完成对一个简单 C 程序的编辑、编译、连接和运行完整过程。然后把 yishuqiao.c 程序运行起来，并得到输出结果。

1.2 计算机的基本硬件组成

从本节开始将介绍一些 C 程序运行的基础，建议有兴趣的读者都看一下。

1.2.1 计算机的五大部件

美籍匈牙利科学家冯·诺依曼（John von Neumann）定义了现代计算机的基本结构。它主要由存储器、运算器、控制器、输入设备和输出设备五大部件组成。这五大部件之间通过总线相互通信，如图 1.3 所示。

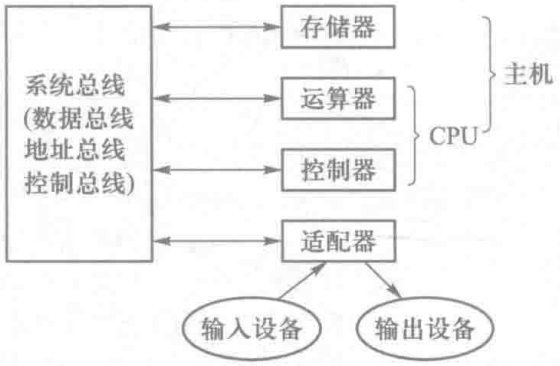


图 1.3 计算机硬件组成

计算机五大部件的基本功能分别如下。

输入设备——输入设备将信息从外界输入到计算机中，比如键盘、鼠标等。

控制器——控制器控制计算机的一切活动，是计算机的指挥中心，协调各部件的工作。

运算器——运算器完成计算机的运算或计算任务。像加、减、乘、除这样的算术运算，以及与、或、非这样的逻辑运算等都在运算器中完成。计算机的 CPU 主要就是由运算器和控制器这两部分组成的。

存储器——存储器是计算机的记忆装置，计算机的程序和数据都保存在存储器中。学习 C 语言一定要关注存储器的结构。

输出设备——输出设备是计算机的一个执行装置，程序运行的结果将通过输出设备输出，比如打印机、显示器等。

计算机五大部件之间通过数据总线、地址总线和控制总线相互连接。

计算机控制器所发出的控制命令就是控制信号，这些控制信号是通过控制总线传递的。

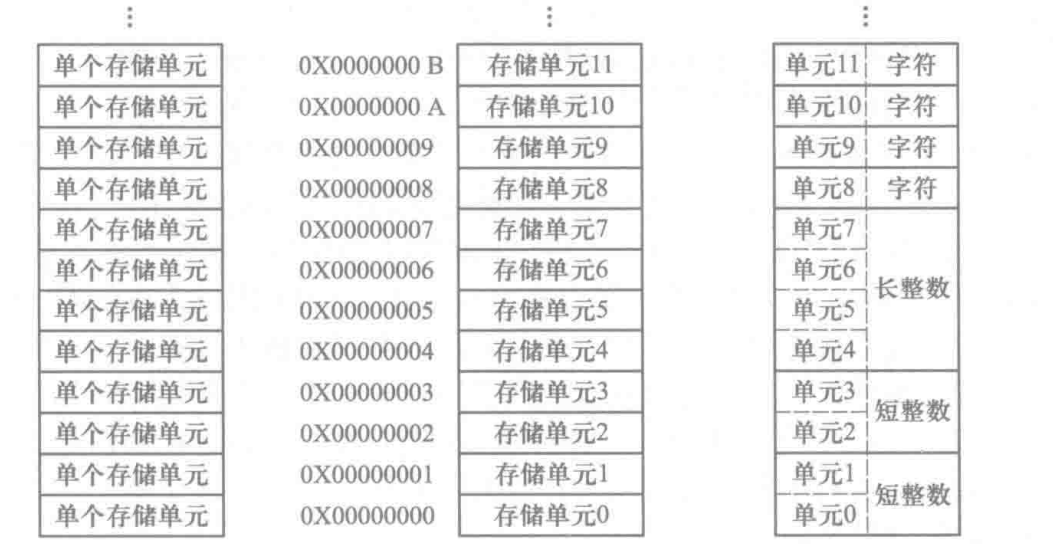
而程序和数据通过输入设备传到存储器，从存储器取到控制器参与运算，最后又将运算

结果通过输出设备输出，这些都是数据信号。数据信号通过数据总线进行传递。

存储器由很多存储单元组成，通常每个存储单元有 8 位，即 1 个字节宽度，为了区分不同的存储单元，每个存储单元都有一个独一无二的编号，这个编号实际上就是该存储单元的地址。在存取数据时首先就要确定存储单元的位置信息，即存取该数据的地址信号，这个地址信号就是通过地址总线传递的。

1.2.2 存储器抽象结构

图 1.4 示意了存储器的抽象结构。存储器由很多存储单元组成。存储单元是存储器的最小存取单位。单个存储单元可以看成是由 8 个开关组成的简单结构。存储单元的每一个开关就代表存储单元中的一“位”。单个位只能存放一个 0 或者 1，因为它只有两种状态，即“开”和“关”，分别代表二进制的 1 和 0。那么 8 个开关就只能表示 256 种 1 和 0 的不同组合。计算机正是用这种组合来代表具体的数据。也就是说一个存储单元最多只能表示 256 个数据。这就是计算机用单个存储单元来存储字符的道理——扩展的 ASCII 码只有 256 个字符（普通 ASCII 码只有 128 个，只需要一个字节的 7 位即可表示），一个存储单元中的 8 位可以有 256 种组合，足以表达它。而整数就远远不止 256 个，所以计算机就不得不用连续多个存储单元来表示整数。为了既满足存储需要又节约存储空间，计算机又对整数进行了分类，比如，分为短整型和长整型等。一般的系统仅给短整型分配两个连续存储单元，这就意味着它可用 16 个开关来表示 1 和 0，即有 65 536 种不同的 1 和 0 的组合。所以短整型只能表示 65 536 个整型数据。



(a) 所有存储单元按照顺序排列在一起组成存储器

(b) 为便于访问，给每个存储单元编号，这个编号就是存储单元的地址

(c) 单个或多个连续存储单元可以存放一个数据，比如一个字符或者一个整数

图 1.4 存储单元的地址

由于存储器的存储单元很多，如何寻找到每一个具体的存储单元是一个比较大的问题。为了寻找方便，计算机设计了一个地址系统给每一个存储单元赋予了一个唯一的编号。这个

编号就是地址，就是该存储单元的永久代号。通过该编号就能找到对应的存储单元。比如，当计算机要把一个短整数存放到存储器时，计算机将为该短整数分配两个存储单元。计算机如何知道是哪两个存储单元呢？计算机正是通过这两个存储单元的地址编号来关联上这两个存储单元的。这就是地址的重要作用。

计算机的地址编号一般用十六进制数表示。十六进制数有16个基本符号，分别为0、1、2、3、4、5、6、7、8、9、A、B、C、D、F。每一个基本符号对应一个4位二进制码，从0000到1111。也就是说每一个基本符号代表了4根地址总线。对于一个具有32根地址总线的计算机系统，通常用8个十六进制基本符号表示其地址编码，比如，0X0000000A，这里的0X表示后面跟的数字是十六进制数。

1.2.3 内存分区

计算机的存储器主要指内存，它一般是分区组织的。为了便于理解，可简单地将其分为程序区和数据区两部分。在内存中运行的程序代码放置在程序区，程序运行过程中所需的变量、常量等数据放置在数据区。数据区又分为静态数据存储区和动态数据存储区两个部分，静态数据存储区简称为静态存储区，动态数据存储区简称为动态存储区。其中动态存储区又分为“堆”和“栈”两种结构形式。动态数据存储区的数据在程序运行过程中随时可能消失，而静态数据存储区的数据将等到程序运行结束时才消失。静态数据区主要用于存放程序运行所需外部变量和内部静态变量等。动态数据区主要用于存放随程序运行而生成的内部动态变量。

“堆”和“栈”是两种不同的数据结构。“栈”空间大小有限，采用“先进后出”的方式管理其数据，这部分空间由操作系统自动管理，无须程序员操作；而“堆”空间较大，需要程序员自己管理。因此，栈的使用效率比堆高。

根据分配方式的不同，计算机内存分配方式分为以下3种。

① 在静态存储区中分配。在静态存储区中分配的变量称为静态变量，它在该程序的整个运行期间都存在，它的生命周期贯穿程序的整个运行周期。C 程序中的外部变量（也称为全局变量）、static 型内部变量（也称为局部变量）等都在静态存储区中分配存储单元。程序在编译时就确定了——程序运行时哪些变量会在静态存储区分配空间。

② 在动态存储区的“栈”上分配。C程序在函数内声明的变量是内部变量。调用函数时，其中未用static修饰的变量（这样的变量也称为自动变量）将在栈上分配存储空间；当函数调用结束时这部分存储空间会被自动释放，其所对应的变量也就不存在了。也就是说，这部分变量的生命周期与函数的执行时间相同，函数调用结束后变量的生命周期也就完结了。

③ 在动态存储区的“堆”上分配，亦称动态内存分配。这部分动态内存空间完全由程序员自己负责管理，它的分配和释放都由程序员自己负责。动态内存是唯一一个生存期可以由程序员自己决定的存储区间。

1.3 计算机基本工作原理

在二进制中，表示信息的基本符号只有两个：0和1。它采用“逢2进1、借1当2”的