C程序开发高手工程实践

Effctive C Programming and Engineering

ECPE

张亮

2017-3-22

作者前言

目前，面向初学者讲述C语言程序开发的书林林总总，高校计算机相关专业的学生一般在大学低年级就接触了《C语言程序设计》这门课，但此时真正能熟练运用好这门语言的不多，觉得它就是只能做做数学题，并不能编出一个游戏或者网站出来。很多学生还要在毕业之后重新学习。这不但使教育资源产生了浪费，而且造成了学生动手能力不足，无法快速适应从学校到工作岗位的角色转变。

大多数C语言程序设计课程只是把语言作为手册来讲解，重点讲解语言本身的规则、语法和数据结构等细节，并没有从工程实践角度很好地阐释如何把程序设计本身与工程项目管理结合起来，没有让学生树立起良好的编程习惯，多数停留在做作业、写DEMO程序的阶段，从而造成学生缺乏良好的科学研究、工程实践的基本素质。

软件产品就是给用户提供自动化的解决方法，然而经验不足的开发人员在软件产品本身的开发过程中，往往忽视了项目规范化与自动化管理。很多从事开发工作的工程技术人员，常带有业务化的思维，缺少真正工程化的方法。工程化的方法相当于软件工程师之间交流的“语言”，只有通晓这门“语言”，才能有效利用开源代码、工具等，才能参与到开源组织，贡献自己的力量。

程序设计是一门实践性很强的工程化学科，讲究标准化和规范化，存在最佳实践。程序设计同时又是一门创造性、独立性很强的工作。不同的开发人员层次不一，习惯差别很大，造成不同的公司、不同的项目组之间都存在各种差别。而遵从有关国际化组织、开源社区的标准，既是一个软件从业人员应该着力自我修养的基本素质，也是让我们在实践当中少走弯路，提高整个软件开发队伍人才质量的途径。

作者认为，在初学者与有经验的程序开发者之间的差别就在于后者有大量的实践，因此积累了丰富的经验和经过实践验证的代码和工具集。本书作者从自己多年的软件开发实践中不断学习，总结规律，形成了一套适合Ｃ程序开发者使用的工程化方法。通过学习和实践本书介绍的方法，初学者能在较短的时间内掌握C程序项目开发的整个流程，具备工程化的思维，从而快速成为具有生产力的软件开发人员。

本书适合刚刚接触C语言程序开发的在校学生、程序开发初学者，也可以作为软件工程技术人员的参考书，或作为高校的《C语言程序设计》课程的辅助参考书。

本书力求简洁、简单，因此极少讲解C本身的语言规则，读者在阅读和使用本书的时候，还需要参考Ｃ语言相关的教材或者互联网资源。学习编程最好的方法就是实践，本书也不例外。限于作者水平，本书难免有错误和疏漏之处，敬请读者批评指正！

作者

2017年3月上海

目 录

第1章 简单的C程序

1.1 从hello.c开始

1.2 创建一个github.com帐号

1.3 几条git命令

1.4 make与Makefile

1.5 函数

1.6 结构与对齐

1.7 数据类型和指针

第2章 动态库和静态库

2.1 使用现有库

2.2 编译动态库和静态库

2.3 使用动态库和静态库

2.4 库的版本管理

第3章 C程序的输入和命令行参数

3.1 main函数

3.2 交互式命令

3.3 命令行参数解析

3.4 配置文件

ini, xml

第4章 C程序的输出和日志

4.1 屏幕彩色文本输出

4.2 日志文件

4.3 使用log4c

4.4 定制log4crc

第5章: 服务程序的基本架构

5.1 信号处理

5.2 daemon模式

5.3 daemontools配置

5.4 多线程与线程池

5.6 多进程服务

5.7 高并发服务器

第6章: C语言编写库程序

6.1 函数库接口

6.2 代码隐藏与不透明指针

6.3 内存管理

6.4 引用计数

第7章 跨平台C程序

7.1 编译预处理

7.2 条件编译

7.3 编译工具链

cygwin, mingw, gcc, msvc,android,object c

7.4 跨平台项目的框架

7.5 几种开发IDE

第8章 GNU make工具套件

8.1 automake

第9章 C语言程序调试

9.1 GDB

9.2 MSVC

9.3 Eclipse

9.4 内存泄露检测

9.5 性能分析工具

第10章 C程序嵌入脚本

10.1 Lua

10.2 Python

10.3 TCL

10.4 与Java互操作

第11章 常见的几种数据结构

10.1 动态数组

10.2 Hash Map

10.3 链表

10.4 红黑树

10.5 RTree

第12章 C语言与数据库操作

12.1 sqlite

12.2 MySQL

12.3 Oracle

第13章 C语言与网络服务程序

字节编码次序

日志实时同步程序

第14章 C语言实时程序

有限状态机

（？？）

第15章 结语

第1章 简单的C程序

C语言既有相当的效率来取代汇编语言，且又足够地抽象而流畅，能够用于描述各种各样的环境之下的算法与交互。

——C 语言之父，Dennis M. Ritchie

C语言最初由Dennis M. Ritchie于1972年设计并实现。C语言的主要特征为：由函数构成的模块化程序结构；由基本数据类型及其组合形成的结构化类型描述；以运算符和优先级组合形成的灵活的表达式；以指针概念统一地处理地址计算、数组和结构类型的存取；宏定义预处理手段；多个源程序文件组成完整程序的模块化方法。

C语言规模小，灵活性高，描述能力强，可移植性好，执行效率高。不断增长的函数库为C语言程序开发源源不断地提供了可复用的代码，成为人类认知世界里最宝贵的智力财富之一。C语言还是许多高级语言的母语言，几乎所有现代的计算机语言都是在C的基础上发展而来的。C语言的地位对于整个计算机工业界是举足轻重的，是工业界最副盛名的语言。了解C语言也是深刻理解其他语言的基础，学好C语言是作为一个根基扎实的软件开发人员必备的要求。

我是在安装Ubuntu操作系统的电脑上完成全书的写作和示例代码的编译。读者可以到Ubuntu的网站（[https://www.ubuntu.com](https://www.ubuntu.com/)）上下载免费的操作系统，全新安装到电脑上或在已有Windows操作系统的电脑上安装Ubuntu构成双系统。网络上有大量这方面的安装指导，安装前做好重要资料的备份。

在Ubuntu系统上，按住Ctrl+Alt键，再按字母T键，就可以调出一个命令行终端。输入下面的命令，查看gcc编译器（http://gcc.gnu.org/）的版本：

$ gcc --version

gcc (Ubuntu 4.8.4-2ubuntu1~14.04.3) 4.8.4

Copyright (C) 2013 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

不同的电脑上gcc的版本也许并不一致，一般不会影响代码的编译。

本书的实例程序，如未特殊说明，都是在Ubuntu上采用gcc进行编译。读者需要了解基本的Linux操作命令，如创建、删除目录、文件等。读者可以采用任何文本编辑器来书写源代码文件，如：vi、vim、nano、gedit 、emacs。我采用一个小巧的文本编辑器：Geany。读者可以打开Ubuntu Software Center程序，在其中搜索Geany并安装之。

1.1 第一个C语言程序

在1978年由Brian W. Kernighan和Dennis M. Ritchie合著的《The C Programming Language》一书的第一章中，以一个“hello, world”例子为开场白。本书依然借用这个例子（下面的内容保存成文件：hello.c，存放在~/ecpe/examples/ch1/目录下）:

/\*\*

\* hello.c

\*/

#include <stdio.h>

int main()

{

printf("hello, world\n");

return 0;

}

打开命令行终端，进入hello.c文件所在的目录（~/CStudy/ch1/），执行编译命令：

$ cd ~/ecpe/examples/ch1/

$ gcc -o hello hello.c

这样就创建了hello程序，同样在命令行中执行之：

$ ./hello

hello, world

本文由一个最简单的例子说明了一个C程序是由函数构成的事实。main函数是可执行的C程序的入口函数，程序总是要从main函数开始执行，到main函数的结尾结束。如果不调用其他函数，C程序基本什么事情也做不了。 例子调用了系统库函数printf，它是在 stdio.h头文件中声明的，因此需要使用#include 把它包含进来。查看printf函数的用法，可以在命令行中输入：

$ man printf

1.2 创建一个github.com帐号

假定我们创建了项目ch1，目前项目ch1已经有了第一个源文件：hello.c。随着项目的进展，将会有更多的源文件加入进来，因此需要有服务器能保存我们的程序源代码，并能查看每次提交更改的历史版本，看看代码发生了哪些变化。如果我们想换台电脑只要能连上服务器，就能查看并修改代码。或者其他人也想参与到项目中来，共同维护修改代码。因此项目应该有个适合的版本管理机制。Git就是这样一个工具（<https://git-scm.com/>）。Git是一个开源的分布式版本控制系统，用以高效、快速地处理从很小到非常大的项目版本管理。

Git分为客户端和服务端。Ubuntu上安装Git客户端很简单，就一个命令：

$ sudo apt-get install git

服务端需要自备服务器并搭建Git的服务（如开源的gitlab），这是一个相当繁琐的过程，不过可以下载bitnami的gitlab一键安装包：

<https://bitnami.com/redirect/to/39039/bitnami-gitlab-7.1.1-0-linux-x64-installer.run>

这里我们采用https://github.com开源网站的免费服务，存放我们的项目代码。首先到github.com上注册一个帐号，然后登录网站，创建自己的项目（project）。

1.3 几条git命令

$ git clone

$ git pull

$ git push

$ git add -A

1.4 关于.gitignore

1.5 make与Makefile

1.5 函数

1.6 结构与对齐

1.7 数据类型和指针