

C语言进阶

- 嵌入式系统高级C语言编程

第一讲 概述

凌 明

trio@seu.edu.cn

东南大学国家专用集成电路系统工程技术研究中心

www.cnasic.com

目 录

- 关于本课程
- 小测验
- 测验讲解
- C语言复习

www.cnasic.com

为什么你要选这门课

- 会C语法与会用C语言写程序是两个概念
- C的灵活性以及由此而产生的陷阱非常非常多
- The Most Beautiful Language and Most Dangerous Language in the Programming World!
- 本课程将嵌入式系统中经常用到的C编程技巧与概念介绍给大家

www.cnasic.com

课程安排

- 学分 2；
- 先修课程：C语言编程（我们假设学员已经了解基本的C语法，并至少在一个编译器环境中编写过C语言，虽然我们在课程中有4个小时左右的C语言复习）
- 总学时40小时，
 - 其中授课30小时，分为10讲；
 - 实验（Project）30小时，要求学生独立完成一个中等规模的C语言项目。
 - 实验（Project）安排3小时左右的环境使用课程
- 成绩：
 - 考试 60%
 - Project 40%

www.cnasic.com

授课安排

- 第一讲 概论 Week3
- 第二讲 C语言基本语法复习 Week5
- 第三讲 编译，汇编，链接与调试 Week5
- 第四讲 存储器与指针 (Memory & Pointer) Week7
- 第五讲 数据结构与链表 (Data structure & Link List) Week9
- 第六讲 中断与设备驱动 (Interrupt & Device Driver) Week11
- 第七讲 编码风格 (Coding Style) Week11
- 第八讲 编写安全无错的代码 Week13
- 实验：ASIX OS Simulator的使用 Week13
- 第九讲 ASIX Windows GUI 设计详解一 Week15
- 第十讲 ASIX Windows GUI 设计详解二 Week15
- 课程项目：ASIX Windows的控件开发

www.cnasic.com

Course Project

- 掌握 MS VC++环境的使用
- 熟悉 ASIX OS Simulator环境
- 熟悉 ASIX Window的体系结构
- 为 ASIX Windows 编写一个控件，并挂接到ASIX Windows中
- 要求学生提供源程序、设计文档、演示程序
- 实验指导：博士生 + 研究生

www.cnasic.com

CNASIC

ASIX OS Simulator



www.cnasic.com

CNASIC

考试

- 考试形式：闭卷
- 时间：120分钟
- 考试内容：
 - 简答题
 - 代码分析（分析与排错）
 - 编程题

www.cnasic.com

参考书籍和资料

C语言初级教材

(1) **《C程序设计语言》** (The C Programming Language)

作者：Brian W.Kernighan, Dennis M.Ritchie

简介：本书是由C语言的设计者Brian W. Kernighan和Dennis M. Ritchie编写的一部介绍标准C语言及其程序设计方法的权威性经典著作。一本必读的程序设计语言方面的参考书。

(2) **《C Primer Plus中文版》** (C Primer Plus)

作者：Stephen Prata

简介：作为核心计算机技术成熟、完整的参考书籍，Primer Plus系列历经数十年不衰，通过学习《C Primer Plus(第五版)中文版》，你将奠定坚实的C编程基础。

(3) **《C和指针》** (Pointers on C)

作者：Kenneth A.Reek

简介：本书提供与C语言编程相关的全面资源和深入讨论。全书覆盖了数据、语句、操作符和表达式、指针、函数、数组、字符串、结构和联合等几乎所有重要的C编程话题。书中给出了很多编程技巧和提示。

www.cnasic.com

参考书籍和资料

C语言进阶书籍

(1) **《C陷阱与缺陷》** (C Traps and Pitfalls)

作者：Andrew Koenig

简介：作者以自己1985年在Bell实验室时发表的一篇文章为基础，结合自己的工作经验扩展成为这本对C程序员具有珍贵价值的经典著作。本书的出发点不是要批判C语言，而是要帮助C程序员绕过编程过程中的陷阱和障碍。

(2) **《C专家编程》** (Expert C Programming)

作者：Perter Van Der LinDen

简介：展示了最优秀的C程序员所使用的编码技巧。书中C的历史、语言特性、声明、数组、指针、链接、运行时、内存以及如何进一步学习C++等问题进行了细致的讲解和深入的分析。全书撷取几十个实例进行讲解，对C程序员具有非常高的实用价值。

(3) **《C语言编程常见问题解答》** (C Programming : Just the Faqs)

作者：Paul S.R.Chisholm 等

简介：这是一本专门解答C语言编程常见问题的著作。书中所覆盖的内容相当广泛，并附有大量鲜明的例子。

www.cnasic.com

嵌入式编程书籍

- (1) **《C/C++ 嵌入式系统编程》** (Programming Embedded Systems in C and C++)

作者：Michael Barr

简介：本书中的技术和范例代码均可直接应用于各种实际的嵌入式系统项目。有过嵌入式系统开发经验的读者，也将从中获益匪浅。

- (2) **《嵌入式实时操作系统 μ C/OS-II》** (MicroC/OS-II The Real-Time Kernel)

作者：Jean J.Labrosse

简介： μ C/OSII是源码公开的实时内核，是专为嵌入式应用设计的。通过对 μ C/OSII源代码的分析与描述，讲述了多任务实时的基本概念、竞争与调度算法、任务间同步与通信、存储与定时的管理以及如何处理优先级反转问题；介绍如何将 μ C/OSII移植到不同CPU上，如何调试移植代码。

- (3) **《嵌入式系统构件》** (Embedded Systems Building Blocks)

作者：Jean J.Labrosse

简介：本书介绍了构建嵌入式系统的一些通用模块，如键盘扫描器、显示器接口、计量器和输入/输出。大部分代码都是用可移植的C语言编写。

www.cnasic.com

网上资源

- (1) **中国软件开发网** <http://www.csdn.net/>

是面向IT专业技术人员和软件开发及应用企业，以专业社区为中心的IT专业知识传播网站。

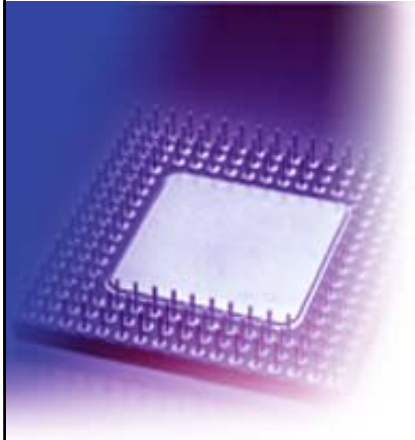
- (2) **C 语言常见问题集** <http://ols3.tnc.idv.tw/prg/cpp/ccfaq/ccfaq.html>

英文 C-FAQ (2004 年 7 月 3 日修订版) 的中文版。

- (3) **Programming in C** <http://www.lysator.liu.se/c/index.html>

关于C语言编程的资源非常多，但都是E文的

www.cnasic.com



QUIZ ?

www.cnasic.com

小 测 验

- 1、用变量a给出下面的定义
 - a) 一个整型数 (An integer)
 - b) 一个指向整型数的指针 (A pointer to an integer)
 - c) 一个指向指针的的指针，它指向的指针是指向一个整型数 (A pointer to a pointer to an integer)
 - d) 一个有10个整型数的数组 (An array of 10 integers)
 - e) 一个有10个指针的数组，该指针是指向一个整型数的 (An array of 10 pointers to integers)
 - f) 一个指向函数的指针，该函数有一个整型参数并返回一个整型数 (A pointer to a function that takes an integer as an argument and returns an integer)
 - g) 一个有10个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数 (An array of ten pointers to functions that take an integer argument and return an integer)

www.cnasic.com

小 测 验

- 2、关键字static的作用是什么？

- 3、关键字const是什么含意？

- 4、关键字volatile有什么含意？

小 测 验

- 5、嵌入式系统总是要用户对变量或寄存器进行位操作。给定一个整型变量a，写两段代码，第一个设置a的bit 3，第二个清除a的bit 3。在以上两个操作中，要保持其它位不变。

小 测 验

- 6、嵌入式系统经常具有要求程序员去访问某特定的内存位置的特点。在某工程中，要求设置一绝对地址为0x67a9的整型变量的值为0xaa66。编译器是一个纯粹的ANSI编译器。写代码去完成这一任务。

小 测 验

- 7、中断是嵌入式系统中重要的组成部分，这导致了很多编译开发商提供一种扩展—让标准C支持中断。具代表事实是，产生了一个新的关键字 `__interrupt`。下面的代码就使用了 `__interrupt` 关键字去定义了一个中断服务子程序(ISR)，请评论一下这段代码的。

```
__interrupt double compute_area (double radius)
{
    double area = PI * radius * radius;
    printf(" Area = %f", area);
    return area;
}
```

小 测 验

- 8、下面的代码输出是什么，为什么？

```
void foo(void)
{
    unsigned int a = 6;
    int b = -20;
    (a+b > 6) ? puts("> 6") : puts("<= 6");
}
```

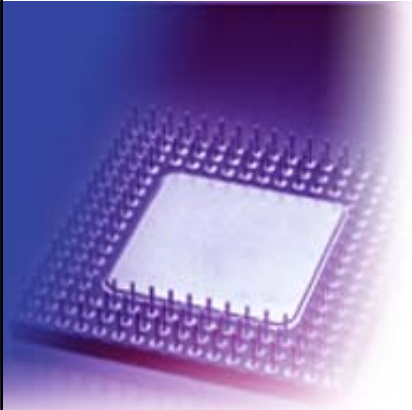
www.cnasic.com

小 测 验

- 9、C语言同意一些令人震惊的结构,下面的结构是合法的吗，如果是它做些什么？

```
int a = 5, b = 7, c;
c = a+++b;
```

www.cnasic.com



Answer!

Quiz 1 Answer

- a) `int a;` // An integer
- b) `int *a;` // A pointer to an integer
- c) `int **a;` // A pointer to a pointer to an integer
- d) `int a[10];` // An array of 10 integers
- e) `int *a[10];` // An array of 10 pointers to integers
- f) `int (*a)(int);` // A pointer to a function a that takes an integer argument and returns an integer
- g) `int (*a[10])(int);` // An array of 10 pointers to functions that take an integer argument and return an integer

Quiz 2 Answer

- 在C语言中，关键字static有三个明显的作用：
 - 1). 在函数体，一个被声明为静态的变量在这一函数被调用过程中维持其值不变。
 - 2). 在模块内（但在函数体外），一个被声明为静态的变量可以被模块内所用函数访问，但不能被模块外其它函数访问。它是一个本地的全局变量。
 - 3). 在模块内，一个被声明为静态的函数只可被这一模块内的其它函数调用。那就是，这个函数被限制在声明它的模块的本地范围内使用。

Quiz 3 Answer

- `const int a;`
`int const a;`
`const int *a;`
`int * const a;`
`int const * a const;`
- 前两个的作用是一样，a是一个常整型数。
- 第三个意味着a是一个指向常整型数的指针（也就是，整型数是不可修改的，但指针可以）。
- 第四个意思a是一个指向整型数的常指针（也就是说，指针指向的整型数是可以修改的，但指针是不可修改的）。
- 最后一个意味着a是一个指向常整型数的常指针（也就是说，指针指向的整型数是不可修改的，同时指针也是不可修改的）。

CONST的优点

- 关键字const的作用是为给读你代码的人传达非常有用的信息，实际上，声明一个参数为常量是为了告诉了用户这个参数的应用目的。如果你曾花很多时间清理其它人留下的垃圾，你就会很快学会感谢这点多余的信息。（当然，懂得用const的程序员很少会留下的垃圾让别人来清理的。）
- 通过给优化器一些附加的信息，使用关键字const也许能产生更紧凑的代码。
- 合理地使用关键字const可以使编译器很自然地保护那些不希望被改变的参数，防止其被无意的代码修改。简而言之，这样可以减少bug的出现。

www.cnasic.com

Quiz 4 Answer

- 一个定义为volatile的变量是说这变量可能会被子想不到地改变，这样，编译器就不会去假设这个变量的值了。精确地说就是，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。下面是volatile变量的几个例子：
 - 1). 并行设备的硬件寄存器（如：状态寄存器）
 - 2). 一个中断服务子程序中会访问到的非自动变量(Non-automatic variables)
 - 3). 多线程应用中被几个任务共享的变量

www.cnasic.com

Quiz 5 Answer

```
■ #define BIT3  (0x1<<3)
    static int a;
    void set_bit3(void)
    {
        a |= BIT3;
    }
    void clear_bit3(void)
    {
        a &= ~BIT3;
    }
```

Quiz 6 Answer

```
■ int *ptr;
    ptr = (int *)0x67a9;
    *ptr = 0xaa55;
```

Quiz 7 Answer

- 这个函数有太多的错误了，以至让人不知从何说起了：
 - 1). ISR 不能返回一个值。如果你不懂这个，那么你不会雇佣的。
 - 2). ISR 不能传递参数。如果你没有看到这一点，你被雇佣的机会等同第一项。
 - 3). 在许多的处理器/编译器中，浮点一般都是不可重入的。有些处理器/编译器需要让额处的寄存器入栈，有些处理器/编译器就是不允许在ISR中做浮点运算。此外，ISR应该是短而有效率的，在ISR中做浮点运算是不明智的。
 - 4). 与第三点一脉相承，printf()经常有重入和性能上的问题。

Quiz 8 Answer

- 这无符号整型问题的答案是输出是“>6”。原因是当表达式中存在有符号类型和无符号类型时所有的操作数都自动转换为无符号类型。因此-20变成了一个非常大的正整数，所以该表达式计算出的结果大于6。这一点对于应当频繁用到无符号数据类型的嵌入式系统来说是非常重要的

Quiz 9 Answer

- 这个问题将做为这个测验的一个愉快的结尾。不管你相不相信，上面的例子是完全合乎语法的。问题是编译器如何处理它？水平不高的编译作者实际上会争论这个问题，编译器应当能处理尽可能所有合法的用法。因此，上面的代码被处理成：

```
c = a++ + b;
```

因此，这段代码执行后 $a = 6$, $b = 7$, $c = 12$ 。

- 如果你知道答案，或猜出正确答案，做得好。如果你不知道答案，我也不把这个当作问题。我发现这个问题的最大好处是：这是一个关于代码编写风格，代码的可读性，代码的可修改性的好的话题。