**嵌入式研发工程师面试试题**

1. ANSI C/C++方面的知识   
   一.1、简答题。下面的题目必须全部答对才给分(20分)：   
   1、 如何在C中初始化一个字符数组。2、 如何在C中为一个数组分配空间。   
   3、 如何初始化一个指针数组。   
   4、 如何定义一个有10个元素的整数型指针数组。   
   5、 s[10]的另外一种表达方式是什么。   
   6、 GCC3.2.2版本中支持哪几种编程语言。   
   7、 要使用CHAR\_BIT需要包含哪个头文件。   
   8、 对(-1.2345)取整是多少？   
   9、 如何让局部变量具有全局生命期。   
   10、C中的常量字符串应在何时定义？   
   11、如何在两个.c文件中引用对方的变量。   
   12、使用malloc之前需要做什么准备工作。   
   13、realloc函数在使用上要注意什么问题。   
   14、strtok函数在使用上要注意什么问题。   
   15、gets函数在使用上要注意什么问题。   
   16、C语言的词法分析在长度规则方面采用的是什么策略？   
   17、a+++++b所表示的是什么意思？有什么问题？   
   18、如何定义Bool变量的TRUE和FALSE的值。   
   19、C语言的const的含义是什么。在定义常量时，为什么推荐使用const，而不是#define。   
   20、C语言的volatile的含义是什么。使用时会对编译器有什么暗示。   
     
   一.2、问答题。   
   1、———————————————————–   
     
   “匈牙利命名法”有什么优缺点？(2分)   
     
   2、———————————————————–   
     
   下面x, y, \*p的值是多少，有什么问题？(2分)   
   int x, y, z = 2;   
   int \*p=&z;   
   x=sizeof\*p;   
   y=x/\*p; /\* x=?, \*p=?, y=?, 有什么问题？\*/   
     
   3、———————————————————–   
     
   下面的语句是什么意思？如何声明或定义才使它们更易懂？(10分)   
   int (\*foo())();   
   int (\*foo())[];   
   int (\*foo[])();   
   (\*(void(\*)())0)();   
   void (\*signal(int,void(\*)(int)))(int);   
     
   4、———————————————————–   
     
   本题(2分)。一般使用malloc时，需要进行强制类型转换，如：   
   char \*s; s = (char \*)malloc(31);   
   下面中???该如何填写，才可以正确执行强制类型转换？   
   int (\*monthp)[31]; monthp = (???)malloc(31);   
     
   5、———————————————————–   
     
   关于C语言运算符优先级的记忆技巧是什么？(2分)   
   /\* 下面r的值是多少 \*/   
   int hi, low, r;   
   hi=7;low=3;   
   r=hi < <4+low;   
     
   6、———————————————————–   
     
   指针和数组的区别是什么？用一个简单的声明把它们区分开。(2分)   
   指针和数组的声明在什么情况下是相同的？(2分)   
     
   7、———————————————————–   
     
   C语言的左值(lvalue)和右值(rvalue)的含义是什么？(2分)   
     
   8、———————————————————–   
     
   为什么C语言可以实现printf(char \*format, …)这样可变参数的调用形式？这样有什   
     
   么缺点？(2分)   
     
   9、———————————————————–   
     
   说明C语言中术语”声明”"定义”"原型”的含义？(2分)   
     
   10、———————————————————–   
     
   举一个例子，说明使用assert和防错代码的区别。(5分)   
     
   11、———————————————————–   
     
   对语句 if else 与操作符 ? : 使用场合的比较。(2分)   
     
   12、———————————————————–   
     
   编写一个函数，输入一个的整型数字，可以选择按照8/10/16进制输出字符串。   
   注意边界值。(5分)   
     
   13、———————————————————–   
     
   本题(2分)。下面是一个16×16的黑白图标：   
   static unsigned short stopwatch[] = {   
   0×07c6,   
   0×1ff7,   
   0×383b,   
   0×600c,   
   0×600c,   
   0xc006,   
   0xc006,   
   0xdf06,   
   0xc106,   
   0xc106,   
   0×610c,   
   0×610c,   
   0×3838,   
   0×1ff0,   
   0×07c0,   
   0×0000,   
   };   
   如何修改声明，可以使之在源代码中形象地表现出图形的模样。   
     
   14、———————————————————–   
   说出可以使用calendar[11][30]变量的四种类型定义。(5分)   
   如：int calendar[12][31]; /\* 二维数组 \*/   
     
   15、———————————————————–   
   使用strcmp，当字符串相同时会返回’\0′。但’\0′一般作为逻辑假，   
   因此下面的语句不容易理解:   
   if (!strcmp(s, “string”)) return EQUATION;   
   如何经过简单修改，使之更易懂？(2分)   
     
   16、———————————————————–   
   编写一个自己的完全C语言版本的memset函数，并且评价这个实现的性能和可移植性。(5   
     
   分)   
     
   17、———————————————————–   
   在树和图这些数据结构中，通常使用指针来组织数据。如果我们要把这些数据保存到文   
     
   件   
   中，指针是没有意义的。我们该如何解决这个问题。(2分)   
     
   18、———————————————————–   
   用2种不同的方法计算long变量的”1″bit的个数。(2分)   
     
   19、———————————————————–   
   任意给出一个C的基本数据类型，如何编码判断这个数据类型是有符号还是无符号的？(2   
     
   分)   
   不得上机实验，写出下面代码的输出。解释这个行为是标准定义的，还是依赖实现的。(   
     
   2分)   
   int i;   
   for (i = 0; i  < 10; i++) {   
   int j = i;   
   printf (”%d\n”, j);   
   }   
     
   20、———————————————————–   
   列出5种以上你所看过的C编程的书籍，并写简要书评。(5分)   
   对C的评价。如果要你改造一把菜刀，使之更加安全，你是否会使用这样的菜刀，为什么   
     
   ？(5分)   
     
   一.3、分析题。   
   本题(各5分)。假设下面代码中的变量都是合法变量，调用外部的函数都是正确的。回答   
     
   几个问题：   
   这些代码意图要干什么？   
   是否有问题？   
   如果有问题，该如何修改，或者如何避免类似错误发生？   
   如果没有问题，如果代码有输出，输出是什么？   
   1、———————————————————–   
     
   int isvowel (char c)   
   {   
   return c==’a' ¦ ¦c==’e' ¦ ¦c==’i' ¦ ¦c==’o' ¦ ¦c==’u';   
   }   
     
   2、———————————————————–   
     
   while (c==’\t’ ¦ ¦c=’ ‘ ¦ ¦c==’\n’)   
   c=getc(f);   
     
   3、———————————————————–   
     
   /\* 当x=2, y=3, z=? \*/   
   if (x==0)   
   if (y==0)   
   z=-1;   
   else   
   z=x+y;   
     
   4、———————————————————–   
     
   /\* 处理网络事件 \*/   
   void process\_network\_code(int x, int y)   
   {   
   /\* 选择modes\_pointer资源 \*/   
   switch (line) {   
   case THING1:   
   /\* 处理异常1#, 使用老的modes\_pointer资源 \*/   
   doit1();   
   break;   
   case THING2:   
   /\* 处理异常2#, 需要重新启动服务 \*/   
   if (x == STUFF) {   
   /\* 重新申请modes\_pointer资源，没有初始化 \*/   
   do\_first\_stuff();   
   /\* 在这种条件下，有些资源不用重新申请 \*/   
   if (y == OTHER\_STUFF)   
   break;   
   /\* 申请剩下的资源，并初始化 \*/   
   do\_later\_stuff();   
   }   
   /\* 初始化modes\_pointer资源 \*/   
   initialize\_modes\_pointer();   
   break;   
   default:   
   /\* 处理普通事件, 使用老的modes\_pointer资源 \*/   
   processing();   
   }   
   /\* 使用modes\_pointer资源，处理事件 \*/   
   use\_modes\_pointer();   
   }   
     
   5、———————————————————–   
     
   int is\_gb2312\_char(char c1, char c2)   
   {   
   if (c1 >= 0xa1 && c2 >= 0xa1)   
   return 1;   
   else   
   return 0;   
   }   
     
   6、———————————————————–   
     
   下面x, y的值是多少，有什么问题？   
   int x = 10, y = 3;   
   x ^= y;   
   y ^= x;   
   x ^= y;   
   /\* x=?, y = ? \*/   
     
   7、———————————————————–   
     
   int days[]={31,28,31,30,31,30,31,31,30,31,30,31,};   
   int calendar[12][31];   
   int (\*monthp)[31];   
   int \*dayp;   
   int i;   
     
   memset(calendar, 0, sizeof(calendar));   
   i = 0;   
   for (monthp = calendar; monthp  < &calendar[12]; monthp++) {   
   for (dayp = \*monthp; dayp  < &(\*monthp)[31]; dayp++) {   
   if (dayp - \*monthp  < days[calendar - monthp]) {   
   \*dayp = i++ % 7 + 1;   
   }   
   }   
   }   
     
   8、———————————————————–   
     
   void printnum(long n)   
   {   
   if (n  < 0) {   
   putchar(’-');   
   n = -n;   
   }   
     
   if (n >= 10) {   
   printnum(n/10);   
   }   
     
   putchar (”0123456789″[n%10]);   
   }   
     
   9、———————————————————–   
     
   void \* memchr(void \*pv, unsigned char ch, size\_t size)   
   {   
   unsigned char \*pch = (unsigned char \*) pv;   
   unsigned char \*pchEnd = pch + size;   
     
   while (pch  < pchEnd) {   
   if (\*pch == ch)   
   return (pch);   
   pch++;   
   }   
     
   return NULL;   
   }   
     
   10、———————————————————–   
     
   void \* memchr(void \*pv, unsigned char ch, size\_t size)   
   {   
   unsigned char \*pch = (unsigned char \*) pv;   
   unsigned char \*pchPlant = pch + size;   
   unsigned char chSave = \*pchPlant;   
     
   \*pchPlant = ch;   
   while (pch != ch) {   
   pch++;   
   }   
   \*pchPlant = chSave;   
     
   return ((pch == pchPlant) ? NULL : pch);   
   }   
     
   11、———————————————————–   
     
   void UnsToStr(unsigned short int u, char \*str)   
   {   
   char \*pch;   
     
   assert(u  <= 65535);   
   pch = &str[5];   
   \*pch = ‘\0′;   
   do {   
   \*–pch = u % 10 + ‘0′;   
   } while ((u / 10) > 0);   
     
   strcpy(str, pch);   
   }   
     
   12、———————————————————–   
     
   void \*memmove(void \*pvTo, void pvFrom, size\_t size)   
   {   
   char \*pbTo = (char \*)pvTo;   
   char \*pbFrom = (char \*)pvFrom;   
     
   ((pbTo  < pbFrom) ? tailmove : headmove) (pbTo, pbFrom, size);   
   return (pvTo);   
   }   
     
   13、———————————————————–   
     
   void \*memcpy(void \*pvTo, void pvFrom, size\_t size)   
   {   
   char \*pbTo = (char \*)pvTo;   
   char \*pbFrom = (char \*)pvFrom;   
     
   while (size– > 0);   
   \*pbTo++ = \*pbFrom++;   
   return (pvTo);   
   }   
     
   14、———————————————————–   
     
   #include  <stdio.h>   
     
   int main(int argc, char \*argv[])   
   {   
   char s[]=”0123456789″;   
   int i = 0;   
     
   do {   
   printf (”%c”, i++[s]);   
   } while(s?1:printf(”\n”)-1);   
     
   return 0;   
   }   
     
   15、———————————————————–   
     
   int fibonacci(int x)   
   {   
   if (x == 1  ¦ ¦ x == 2)   
   return 1;   
   return fibonacci(x - 2) + fibonacci(x - 1);   
   }   
     
   16、———————————————————–   
     
   这里有一个程序cdecl.c。写出它的工作流程。写出它的使用方法。   
   给出一个典型输入用例，记录下它的输出。   
     
   一.4、综合编程题。   
   要求：   
   1、完成需求，程序运行正确。   
   2、工作原理文档，使用文档完整。   
   3、代码规整优美。注释得当。   
   4、运行速度足够快。   
   5、用工具分析出是哪些代码或函数造成速度瓶颈。   
     
   1、———————————————————–   
   编写一个排序程序。被排序的文件有8MB大小，一行一个随机整数(ASCII格式)。要求对   
     
   这些整数进行   
   排序，并计算平均值，打印出排序所需的时间。(20分)   
     
   2、———————————————————–   
   用dummy header技巧实现一个链表DEMO。要求具有create, insert, delete, search功   
     
   能。   
   编写一个应用程序，使用上面的函数。使用dummy header技巧有什么优点。(20分)   
     
   3、———————————————————–   
   用heapsort算法实现优先队列。要求具有create, insert, delete功能。   
   编写一个应用程序，使用上面的函数。使用heapsort算法有什么优点。(20分)   
     
   用trie(一种多叉树)实现一个字典。要求具有create, insert, delete, search功能。   
     
   编写一个应用程序，使用上面的函数。使用trie树有什么优点。(20分)   
     
   二、POSIX方面的知识。   
   二.1、简答题。下面的题目必须全部答对才给分：(5分)   
   1、在UNIX环境中，编译流程是什么？   
   2、ABI，ELF的英文全称是什么   
   3、一般UNIX的程序有多少段，举一个实际的例子说明。   
   4、如何在kernel二进制代码中找一个字符串。   
   5、段地址F000:FFF0转换为线形地址的值是多少(16进制表达)。   
   6、在一个UNIX文件系统中，文件的唯一性标志是什么？   
     
   二.2、问答题。   
   1、———————————————————–   
   写一些代码。如何用文件实现信号灯？要求如果程序崩溃了，这个文件也将自动被删除   
     
   。   
   为什么可以这样实现信号灯？(5分)   
   传统的signal函数信号处理为什么是不可靠的，信号和系统调用有何关系？(5分)   
   在图形库系统中往往提供timer的功能，除了使用ALARM信号外，你还可以使用什么系统   
     
   调用来实现timer？(5分)   
   2、———————————————————–   
   写一些代码，演示如何正确使用write系统调用。注意看好手册再回答。(2分)   
   如果一个文件以rw模式打开，在进行read/write操作转换时，需要进行什么操作。(2分)   
     
   3、———————————————————–   
   解释终端结构termios.c\_cc的MIN/TIME数值变化的带来read/write的特性。(5分)   
   如何理解终端、控制终端、虚拟终端、控制台、tty等类同的概念？(5分)   
   4、———————————————————–   
     
   解释计算机中Copy-On-Write的概念。(2分)   
     
   调用fork之后，子进程没有继承父进程的属性有哪些？(5分)   
   解释为什么每个程序在装入执行之后，总是已经预先打开了stdout、stdin、stderr？(2   
     
   分)   
   在fork之前，父进程打开了一个文件。在fork之后，如果子进程移动了文件指针，   
   父进程的文件指针有什么变化；如果子进程关闭了文件，父进程有什么变化？为什么会   
     
   这样？(5分)   
   标准输入、输出和错误输出分别是什么类型的缓冲，这些缓冲在用户空间还是在核心空   
     
   间？   
   怎样关闭他们的缓冲？如果父进程关闭了缓冲，在fork之后建立的子进程是否也关闭了   
     
   缓冲？(5分)   
   vfork和fork相比，有什么特色？(2分)   
   system函数是否等同于fork+exec？(2分)   
     
   wait系统调用有多少种条件可以退出？(2分)   
   5、———————————————————–   
   系统调用和库函数调用有什么区别。(2分)   
   在linux2.4.x上的glibc和newlib(一种嵌入式C库)的系统调用有什么不同？(2分)   
   在linux2.4.x上，对系统调用execve如果调用成功，它返回的值是多少。(2分)   
     
   6、———————————————————–   
   列出你所知道的2个内存跟踪库。(2分)   
   设计一个内存跟踪方案，为什么选择这个方案(5分)   
     
   二.3、综合编程题。   
   要求：   
   1、完成需求，程序运行正确。   
   2、工作原理文档，使用文档完整。   
   3、代码规整优美。注释得当。   
   4、运行速度足够快。   
     
   1、———————————————————–   
   这里有一个8MB的文件，编写一个copy程序，拷贝这个文件，并计算所需时间。用工具分   
     
   析   
   出是哪些代码或函数造成速度瓶颈。   
   提示：如果只是使用read/write调用，不是一个好的实现。(20分)   
     
   2、———————————————————–   
   Linux系统中，什么时候会出现类似Y2K的问题。写一个程序证明。(20分)   
     
   3、———————————————————–   
   编写一个程序，测试系统最小的睡眠时间间隔。(20分)   
     
   4、———————————————————–   
   编写一个pipe程序，测试有N个管道，size大小的buffer情况下，pipe的传输性能是多少   
     
   ？(20分)   
     
   5、———————————————————–   
   在ext2文件系统上，单个文件最大可以达到多少？   
   写一个程序获得这方面的限制。(20分)   
     
   三、Linux编程基本使用知识。   
   三.1、命令和shell   
   1）编写一个脚本，统计一个目录下面所有C代码的行数。(2分)   
   2）编写一个sed脚本，去除HTML文件中的HTML标记。(2分)   
   3）编写一个脚本获得当前系统eth0的IP地址。(2分)   
   4）编写一个脚本以交互的方式，进行DNS的设定。(2分)   
   5）使用ls命令编写一个脚本，实现ls -R，递归列出当前目录下的所有目录、文件。(2   
     
   分)   
   6）———————————————————–   
   在一个目录下，找soft-link files，用下面的命令，原理是可行的，但无法操作，   
   该如何解决？要2种方法。(2分)   
   ls -l  ¦ grep ->   
     
   7）———————————————————–   
   如何把标准错误输出，重定向到标准输出上。(2分)   
     
   8）———————————————————–   
   我们的系统中的软件包是使用RPM管理的。要求下面的问题写出shell命令和运行结果。(   
     
   5分)   
   如何知道系统中安装了几个软件包。   
   如何知道系统中安装了哪几类(group)软件包。   
   如何知道kernel软件包的简述。   
   如何知道kernel软件包的Changelog。   
   如何知道kernel软件包有几个文件。   
   如何知道kernel软件包安装后有多大。   
     
   三.2、编辑工具的使用。(各2分)   
   1）如何使用vi进行块拷贝、粘贴、删除的操作   
   2）如何设置Tab的长度，以及自动缩进的长度   
   3）如何使用tag进行代码阅读   
   4）如何在1～10行，有确认的进把所有的RedHat改为RedFlag   
   5）在SourceNavigator中，如何跳到一个变量的声明处？如何得知光标当前的位置在那   
     
   个函数体内？   
     
   三.3、编译器与调试器。(各2分)   
   1）如何使用gcc得到宏展开的中间代码   
   2）如何通过gcc在命令行中传入宏定义   
   3）在那一级优化的情况下，内联函数才真正的内联到代码中   
   4）gdb的watch命令如何使用，有何缺点   
   5）gdb中使用什么命令可以显式调用的栈帧，如何查看某个栈帧上的局部变量   
   6）如何使用gdb调试多进程   
   7）如何使能/禁止core dump？   
     
   三.4、库。(各2分)   
   1）如何知道XSetIMValues这个符号在那个X的库文件中   
   2）如何读取C++的符号名，更具可读性   
   3）请解释动态库的soname概念   
   4）解释在链接时rpath选项的含义   
     
   三.5、Makefile。(10分)   
   如果有一个简单的Test项目目录如下：   
   # tree Test   
   Test   
   ¦– common.h   
   ¦– main.cpp   
   ¦– test.cpp   
   `– test.h   
     
   1）使用wildcard、patsubst函数编写一个C++项目的Makefile文件   
   2）使用automake、autoconfig编写一个支持configure选项的configure.in脚本   
     
   三.6、CVS。(10分)   
   1）如果一个CVS项目中，甲添加了一个子目录模块，那么乙在update时是否会自动检   
   出这个新添加的子模块？如果不能，那么你有什么好的解决方案？   
   2）如何在一个CVS项目中添加一个二进制文件，这个二进制文件可以进行增量的版本管   
     
   理吗？   
   3）如何检出标记（tag）为milestone-1的版本，如何创建以milestone-1为基础的分支   
     
   milestone-1-b1，并如何把分支的修改合并到milestone-1的主分支上   
     
   三.7、Linux/GNU编程基本知识   
   1、———————————————————–   
   Linux2.4.x有多少种类型的设备文件？分别写出来。(2分)   
     
   2、———————————————————–   
   glibc动态库的搜寻次序是什么？(2分)   
   glibc 2.1.x 与 2.2.x的动态库的搜寻次序有什么不同？为什么会采用现在的方案？(2   
     
   分)   
     
   三.3、综合编程题。   
   要求：   
   1、完成需求，程序运行正确。   
   2、工作原理文档，使用文档完整。   
   3、代码规整优美。注释得当。   
   4、运行速度足够快。   
     
   1、———————————————————–   
   本题(20分)。写一个程序，有三个功能：   
   a)取得CMOS中的当前时间，按照YY:MM:DD:HH:MM:SS格式输出。   
   b)取得物理内存的大小，格式化输出。   
   c)取得从物理段地址F000:FFF0起，16个字节的值，每字节按照16进制格式输出。   
     
   2、———————————————————–   
   本题(20分)。对串口编程，编写一对类似ping的程序，作用在串口上，了解serial的联   
     
   通情况。   
   定义协议   
   要求：   
   定义一个联通和响应协议。至少具有：发送、回复、超时三种状态。除超时状态外，所   
     
   有   
   状态的数据必须经过校验才可使用。   
     
   ping\_serial\_client   
   要求：   
   1,发送数据包；   
   2,等待回复数据包，直到超时；   
   3,如果有回复数据包，对数据包进行校验；   
   4,计算校验通过的数据包之间的时间差；   
   5,重复直到用户退出；   
   6,统计丢包率，时间。   
   界面：   
   bash# ping\_serial\_client /dev/ttyS0 baud\_value   
   ??bytes from ttyS0, time=??ms   
   ……   
   ^C   
   — ttyS0 ping statistics —   
   ?? packets transmitted, ?? received, ??% loss   
   time ??ms, min/avg/max = ??/??/?? ms   
     
   ping\_serial\_server   
   要求：   
   1,等待对方的发送数据包；   
   2,对发送数据包进行校验；   
   3,校验通过则回复数据包；   
   4,重复直到用户退出；   
     
   界面：   
   bash# ping\_serial\_server /dev/ttyS0 baud\_value   
     
   3、———————————————————–   
   编写一对socket程序，要求类似于network echo procotol。   
   Client每隔1秒把自己的IP地址轮流循环发送给一个Server.   
   Server接到IP后，在屏幕上打印对方IP；然后把自己的IP发送回去。   
   Client收到回复后，在屏幕上打印对方IP。   
   程序一直运行，直到用户退出。各自分类统计接收到的各IP的数据包的个数。   
   注意：Client可以向多个Server发送请求，Server也可以接收多个Client请求。   
   注意：输出信息的美观，和对错误的处理。   
   ———————————————————–   
     
   四、C++题目。   
     
   ———————————————————–   
     
   1。给定下面的代码:   
     
   class Graph{   
   public:   
   Graph() { s\_gCount++; }   
     
   virtual ~Graph() { s\_gCount–;}   
     
   virtual int drawOut() = 0;   
     
   static int getTotalCount() { return s\_gCount; }   
     
   protected:   
   int m\_x;   
   int m\_y;   
   static int s\_gCount;   
   };   
     
   回答下列问题：(各5分)   
     
   1）这个类能否生成实例，为什么？   
   2）上述代码中有什么错误？   
   3）如果有一个Rectangle类公共继承该类，那么s\_gCount对Rectangle的成员属性是什么   
     
   ？   
   4）对Graph及其子类的计数是否有效，如果无效应该采取什么措施？   
   5）请画出该类的内存布局结构示意图   
     
   ———————————————————–   
     
   2。给定下面的代码:   
   class A   
   {   
   int a;   
   };   
     
   class AA : public A   
   {   
   int aa;   
   };   
     
   class X : public virtual A   
   {   
   int x;   
   };   
     
   class Y : public virtual A   
   {   
   int y;   
   };   
     
   class Z : public X, public Y {   
   int z;   
   };   
     
   回答下列问题：(各5分)   
     
   1）请画出上述5个类的UML类图   
   2）请画出A、AA、X、Z的内存布局结构示意图   
     
   ———————————————————–   
     
   3。给定下面代码:   
   #include  <stdio.h>   
   #include  <stdlib.h>   
     
   typedef struct \_AAA   
   {   
   int a;   
   }AAA, \*PAAA;   
     
   class CAAA   
   {   
   public:   
   CAAA() { m\_a.a = 0; }   
   CAAA(CAAA& in) { printf(”CAAA(CAAA& in)\n”); }   
     
   CAAA(AAA in) { printf(”CAAA(AAA in)\n”); }   
   CAAA(int in) { printf(”CAAA(int in)\n”); }   
     
   private:   
   AAA m\_a;   
     
   public:   
   operator PAAA ()   
   {   
   return &m\_a;   
   }   
   };   
     
   void test0(PAAA pa)   
   {   
   printf(”test0\n”);   
   }   
     
   void test1(CAAA ca)   
   {   
   printf(”test1\n”);   
   }   
     
   void test2(CAAA ca)   
   {   
   printf (”test2\n”);   
   }   
     
   void main(void)   
   {   
   CAAA a;   
   test0 (a);   
     
   AAA b;   
   test1 (b);   
     
   int i=0;   
   test2 (i);   
   }   
     
   请回答下列问题：(各5分)   
     
   1）找出程序中的错误   
   2）test0函数声明传入的参数为PAAA类型，而在main函数中，实际传入的是CAAA类型，   
     
   这是否是一个错误？如果不是，请解释原因。   
     
   4。回答下列问题：   
   1）C++中的struct和class类型有什么区别？(2分)   
   2）假定一个class名为Test，具有构造函数、拷贝构造函数、operator =，那么   
   Test a;   
   Test b=a;   
   分别执行的是哪一个函数？(2分)   
   3）C++程序进行链接时是否必须链接stdc++函数库，如果不必须情况下不需要，什么情   
     
   况下需要？(2分)   
   4）你所熟悉的关于C++的中英文经典书籍都有哪些，它们分别都侧重于什么方面？(5分)   
     
   五、QT编程。任选2题完成。(各20分)   
   要求：按照综合编程题目的要求编写代码和文档。   
   1.编写一个简单的clock控件, 要求从QWidget继承, 有一个指针和表盘,指针每12秒转一   
     
   圈,   
   并且可以用鼠标左键,起动和停止指针的转动.   
     
   2.给定一个designer生成的UI文件, 读取指定名字控件的几何信息,计算出该控件的面积   
     
   大小.   
   (用Dom和SAX都可)   
     
   3.设计一个程序, 建立个人信息库,包括:姓名,年龄,性别,住址.建立完之后可以查处特   
     
   定人   
   的信息,用designer设计你的界面, 不要求把信息库存在硬盘上.   
     
   4.把第三题变成国际化程序, 要求两种语言中文和英文.需要用qt-linquist等工具.   
     
   5. 用QSocket实现两台不同机器间的收发, 收发要求用户动态输入.   
     
   提示：以上各题需在qt3.0以上实现,推荐用qt-3.1.1,不限止qt-embedded或qt-desktop.   
     
   六、MiniGUI编程   
   要求：按照综合编程题目的要求编写代码和文档。   
   1.裁剪MiniGUI字体库,使它仅支持英文和繁体中文。(10分)   
   2.描述MiniGUI/MPP的消息机制(20分)   
   3.使用MPP和MySql,编写一使用在商场POS机上的应用,要求如下:(50分)   
   1)收银员帐户管理:每台POS机供若干个收银员和一个系统管理员使用.收银员凭密码   
     
   登录POS机,密码可修改.收银员只能查看自己的销售数据,   
   但不能删除.系统管理员可以增加删减收银员帐户,查询销售记录.   
   2)收款管理:计算顾客购买商品总金额;计算找零   
   3)顾客会员管理:根据顾客ID,判断会员等级及相应打折级别.   
   4)屏幕锁定功能:特定按健触发屏幕锁定,输入密码后解锁   
     
   七、Kernel & Driver   
   内核基础试题   
   1. 简答题。(各10分)   
   1.1 Spin Lock 相对于信号量的特点；在UP/MP/SMP这些不同环境下，Spin Lock的不同   
     
   实现方法。   
   1.2 简要说明e2compr压缩文件系统的压缩策略；经其处理后，上层文件系统（EXT2/3   
     
   等）是否会受到影响。   
   1.3 比较linux操作系统内核态和用户态的抢占性，及两者之间存在差异的原因。   
   1.4 linux2.2支持的最大进程数是多少，演进至2.4后，这一制约是如何克服的。   
   1.5 列举中断屏蔽的类型，并简单比较。   
   1.6 列举linux所要处理的定时/计数器（实际的外围硬件设备，不是指软件定时器），   
     
   及其用途。   
     
   2. 问答题。(各20分)   
   2.1 穷举在用户态空间，内核消息的获取方式、差异、以及差异的原因。   
   2.2 解释linux进程调度中的epoch概念，该机制的目的。   
   2.3 描述一个硬件中断信号发生，到为其注册的中断服务例程开始执行，之间所经过的   
     
   内核处理过程。并进一步分析，制约linux中断响应时间的因素存在于哪些方面。   
   2.4 比较底半机制（BH）和Tasklet的特点，及运行方面的差异。   
   2.5 说明引导过程中initrd.img和linuxrc作用，给出制作initrd.img的伪代码形式的   
     
   流程。   
   2.6 在VFS层中如何区分设备文件和正规文件?   
   2.7 内核有几种方式访问硬件设备?   
   2.8 内核如何访问iomem?请举例说明。   
     
   3. 分析题。   
   3.1 系统时间设置问题。(10分)   
   同样采用date命令，目的在于修改系统时间，在不同的系统上却有不同的现象：   
   假定初始时间（T1）为 03月01日16：00 2003年，期望的修改后时间（T2）为 12   
     
   月31日00：00 2002年，   
   <1> 在EDK系统中，执行命令 #date 123100002002，系统时间立即会变为T2；   
   但系统重新启动之后，系统时间仍恢复为T1的时间区间： 03月01日16：02   
   2003年，修改并未保留下来。   
   <2> 在RedHat7.2/8.0中，重复上述设置，系统重新启动之后，时间设置得到了保   
     
   留，系统运行于T2的时间区间：12月31日00：02 2002年。   
   请分析 <1> 中所表现出的现象是由于什么原因造成的；   
     
   3.2 在大家相关于串口的工作中，可能会遇到这样的问题：(20分)   
   串口在发送数据时，不是连续且迅速地发送，而是以一个固定的节拍（10秒），且   
     
   每次只发送等长度的一段数据，请列举可能造成这一问题的所有原因；   
   3.3 Oops分析。(50分)   
   以下是一段经过符号解析过的Oops信息，我们知道造成这个Oops的直接原因在于对   
     
   于   
   内核函数\_\_wake\_up的调用过程之中出现了问题；基于上述判断，请基于已经给出   
   的\_\_wake\_up相关源码和反汇编码，大致分析问题出现在哪个源码行，并给出分析结论；   
     
   ———————————————————–   
   ksymoops信息：   
   ksymoops 2.4.4 on i686 2.2.19-rthal3. Options used   
   -v /usr/src/linux-2.2.19/vmlinux (specified)   
   -k /proc/ksyms (specified)   
   -l /proc/modules (specified)   
   -o /lib/modules/2.2.19-rthal3/ (default)   
   -m /usr/src/linux-2.2.19/System.map (specified)   
     
   current->tss.cr3 = 07d29000, %cr3 = 07d29000   
   \*pde = 00000000   
   Oops: 0000   
   CPU: 0   
   EIP: 0010:[ <C01110C1>]   
   Using defaults from ksymoops -t elf32-i386 -a i386   
   eax: 00000014 ebx: c0eabf74 ecx: 00000013 edx: 00000021   
   esi: 00000000 edi: 00000020 ebp: c0eabf6c esp: c0eabf60   
   ds: 0018 es: 0018 ss: 0018   
   Process in.identd (pid: 828, process nr: 6, stackpage=c0eab000)   
   Stack: 00000000 c807ca04 00000021 c0eabf74 c807be21 c0eabfb0 c807bf74   
   c807c940   
   00000000 00000000 00000000 c803fb60 c807e000 00000e20 2b124c28   
   0000027c   
   0010a000 c807c900 00000000 0000f944 bffff944 c803d2ec 00000000   
   00000000   
   Call Trace: [ <c807ca04>] [ <c807be21>] [ <c807bf74>] [ <c807c940>]   
   [ <c803fb60>] [ <c807e000>]   
   [ <c807c900>] [ <c803d2ec>] [ <c803c0a0>]   
   Code: 8b 02 85 45 fc 74 1b 85 ff 74 10 83 7a 44 00 74 0a 85 f6 75   
     
   >>EIP; c01110c1  <\_\_wake\_up+2d/6c>  <=====   
     
   Trace; c807ca04  <[rt\_das]timeout+c4/c8>   
   Trace; c807be21  <[rt\_das]read\_timeout+25/28>   
   Trace; c807bf74  <[rt\_das]pulse\_isr+150/19c>   
   Trace; c807c940  <[rt\_das]timeout+0/c8>   
   Trace; c803fb60  <[rtai]global\_irq\_handler+0/80>   
   Trace; c807e000  <.bss.end+14a1/????>   
   Trace; c807c900  <[rt\_das]board+0/28>   
   Trace; c803d2ec  <[rtai]dispatch\_global\_irq+28/90>   
   Trace; c803c0a0  <[rtai]GLOBAL0\_interrupt+18/34>   
   Code; c01110c1  <\_\_wake\_up+2d/6c>   
   00000000  <\_EIP>:   
   Code; c01110c1  <\_\_wake\_up+2d/6c>  <=====   
   0: 8b 02 movl (%edx),%eax  <=====   
   Code; c01110c3  <\_\_wake\_up+2f/6c>   
   2: 85 45 fc testl %eax,0xfffffffc(%ebp)   
   Code; c01110c6  <\_\_wake\_up+32/6c>   
   5: 74 1b je 22  <\_EIP+0×22> c01110e3   
   <\_\_wake\_up+4f/6c>   
   Code; c01110c8  <\_\_wake\_up+34/6c>   
   7: 85 ff testl %edi,%edi   
   Code; c01110ca  <\_\_wake\_up+36/6c>   
   9: 74 10 je 1b  <\_EIP+0×1b> c01110dc   
   <\_\_wake\_up+48/6c>   
   Code; c01110cc  <\_\_wake\_up+38/6c>   
   b: 83 7a 44 00 cmpl $0×0,0×44(%edx)   
   Code; c01110d0  <\_\_wake\_up+3c/6c>   
   f: 74 0a je 1b  <\_EIP+0×1b> c01110dc   
   <\_\_wake\_up+48/6c>   
   Code; c01110d2  <\_\_wake\_up+3e/6c>   
   11: 85 f6 testl %esi,%esi   
   Code; c01110d4  <\_\_wake\_up+40/6c>   
   13: 75 00 jne 15  <\_EIP+0×15> c01110d6   
   <\_\_wake\_up+42/6c>   
     
   Unable to handle kernel paging request at virtual address 66fe4603   
   current->tss.cr3 = 00e94000, %cr3 = 00e94000   
   \*pde = 00000000   
   Oops: 0000   
   CPU: 0   
   EIP: 0010:[ <c01113e6>]   
   EFLAGS: 00010a83   
   Warning (Oops\_read): Code line not seen, dumping what data is available   
     
   >>EIP; c01113e6  <interruptible\_sleep\_on+5a/78>  <=====   
     
   1 warning issued. Results may not be reliable.   
     
   <附录>   
   1. \_\_wake\_up的源码：   
   void \_\_wake\_up(struct wait\_queue \*\*q, unsigned int mode)   
   {   
   struct task\_struct \*p, \*best\_exclusive;   
   struct wait\_queue \*head, \*next;   
   unsigned int do\_exclusive;   
     
   if (!q)   
   goto out;   
   /\*   
   \* this is safe to be done before the check because it   
   \* means no deference, just pointer operations.   
   \*/   
   head = WAIT\_QUEUE\_HEAD(q);   
     
   read\_lock(&waitqueue\_lock);   
   next = \*q;   
   if (!next)   
   goto out\_unlock;   
     
   best\_exclusive = 0;   
   do\_exclusive = mode & TASK\_EXCLUSIVE;   
   while (next != head) {   
   p = next->task;   
   next = next->next;   
   if (p->state & mode) {   
   if (do\_exclusive && p->task\_exclusive) {   
   if (best\_exclusive == NULL)   
   best\_exclusive = p;   
   }   
   else {   
   wake\_up\_process(p);   
   }   
   }   
   }   
   if (best\_exclusive)   
   wake\_up\_process(best\_exclusive);   
   out\_unlock:   
   read\_unlock(&waitqueue\_lock);   
   out:   
   return;   
   }   
     
   2. \_\_wake\_up的反汇编码：   
   c0111094  <\_\_wake\_up>:   
   c0111094: 55 pushl %ebp   
   c0111095: 89 e5 movl %esp,%ebp   
   c0111097: 83 ec 08 subl $0×8,%esp   
   c011109a: 57 pushl %edi   
   c011109b: 56 pushl %esi   
   c011109c: 53 pushl %ebx   
   c011109d: 89 55 fc movl %edx,0xfffffffc(%ebp)   
   c01110a0: 85 c0 testl %eax,%eax   
   c01110a2: 74 50 je c01110f4  <\_\_wake\_up+0×60>   
   c01110a4: 8d 48 fc leal 0xfffffffc(%eax),%ecx   
   c01110a7: 89 4d f8 movl %ecx,0xfffffff8(%ebp)   
   c01110aa: 8b 18 movl (%eax),%ebx   
   c01110ac: 85 db testl %ebx,%ebx   
   c01110ae: 74 44 je c01110f4  <\_\_wake\_up+0×60>   
   c01110b0: 31 f6 xorl %esi,%esi   
   c01110b2: 89 d7 movl %edx,%edi   
   c01110b4: 83 e7 20 andl $0×20,%edi   
   c01110b7: 39 cb cmpl %ecx,%ebx   
   c01110b9: 74 2d je c01110e8  <\_\_wake\_up+0×54>   
   c01110bb: 90 nop   
   c01110bc: 8b 13 movl (%ebx),%edx   
   c01110be: 8b 5b 04 movl 0×4(%ebx),%ebx   
   c01110c1: 8b 02 movl (%edx),%eax   
   c01110c3: 85 45 fc testl %eax,0xfffffffc(%ebp)   
   c01110c6: 74 1b je c01110e3  <\_\_wake\_up+0×4f>   
   c01110c8: 85 ff testl %edi,%edi   
   c01110ca: 74 10 je c01110dc  <\_\_wake\_up+0×48>   
   c01110cc: 83 7a 44 00 cmpl $0×0,0×44(%edx)   
   c01110d0: 74 0a je c01110dc  <\_\_wake\_up+0×48>   
   c01110d2: 85 f6 testl %esi,%esi   
   c01110d4: 75 0d jne c01110e3  <\_\_wake\_up+0×4f>   
   c01110d6: 89 d6 movl %edx,%esi   
   c01110d8: eb 09 jmp c01110e3  <\_\_wake\_up+0×4f>   
   c01110da: 89 f6 movl %esi,%esi   
   c01110dc: 89 d0 movl %edx,%eax   
   c01110de: e8 2d f9 ff ff call c0110a10  <wake\_up\_process>   
   c01110e3: 3b 5d f8 cmpl 0xfffffff8(%ebp),%ebx   
   c01110e6: 75 d4 jne c01110bc  <\_\_wake\_up+0×28>   
   c01110e8: 85 f6 testl %esi,%esi   
   c01110ea: 74 08 je c01110f4  <\_\_wake\_up+0×60>   
   c01110ec: 89 f0 movl %esi,%eax   
   c01110ee: e8 1d f9 ff ff call c0110a10  <wake\_up\_process>   
   c01110f3: 90 nop   
   c01110f4: 8d 65 ec leal 0xffffffec(%ebp),%esp   
   c01110f7: 5b popl %ebx   
   c01110f8: 5e popl %esi   
   c01110f9: 5f popl %edi   
   c01110fa: 89 ec movl %ebp,%esp   
   c01110fc: 5d popl %ebp   
   c01110fd: c3 ret   
   c01110fe: 89 f6 movl %esi,%esi   
     
   ———————————————————–   
   内核驱动题   
   以下设计应该包括设计文档,实现策略说明,代码包,测试用例,使用说明.   
   要求：按照综合编程题目的要求编写代码和文档。   
   参考资料:  <Linux设备驱动程序>第二版.   
     
   1.设计并实现一个软件watchdog设备，以监视系统运行情况.(50分)   
   说明:watchdog设备用于监测系统运行状态,正常运行的系统定期写watchdog以使其   
   不会超时,一旦超时,意味系统已挂起;watchdog应该重启系统. 现在的软件watchdog   
   不重启系统,只用于监视应用程序的运行.   
     
   2.设计并实现一个简化的、容量可以变化的内存FIFO设备.(50分)   
     
   八、用中文Linux办公。   
   要求本考卷完全在Linux完成，并要求进行版面的格式化或美化。如果只提交TEXT版本，   
     
   本题不得分。(各10分)   
   本题要求是：   
   1、提交进行版面的格式化或美化的PDF版本。为方便修改和评分，还须提交一个lyx或te x版本。   
   2、PDF内嵌的中文字体使用的是红旗桌面版4.0所携带的新版宋体。

九、娱乐。(没有分数)   
不会娱乐的程序员不是好的程序员。本部分没有分数，但是聪明的人在这里是不会留下 空白的。   
1、运行RedHat 7.3上面的推箱子游戏。用方向-数字的方式(如左3，右4)记录下通过第 一关 的过程，同时记录下过关的时间。看谁的步数少，时间短。   
2、运行emacs + doctor游戏。摘录下10句以上的对话，看谁的对话最精彩，同时也看谁