

电子科技大学计算机学院

教师：韩宏

为什么 vc6 打开的文件却不能编译.....	1
为什么一个 vc6 工程中有两个 main 函数不能编译成功？我想写两个程序怎么办？	2
为什么 v6 不能编译程序，生成信息说 compile 和 link 都正确，到了 link 生成 exe 时，却报错无法生成该 exe 文件.....	2
为什么 Vc6 在编译时，下方消息窗体显现了 compiling...或 linking...后就死了，没有反应了，怎么办	2
如何用任务窗体关闭进程.....	3
如何生成一个 vc6 工程.....	3
如何查看工程中的文件.....	7
如何添加一个已有文件到工程.....	8
如何添加一个新文件到 vc6 中	9
为什么 Scanf 执行后程序报错死了	10
如何通过编译器的编译消息框发现语法错误.....	10
如何发现代码中混入了中文信息.....	11
如何打/取消断点	12
如何让程序可以调试方式运行	12
为何不能在断点停下来.....	12
为何我不能看到输出的结果，程序闪了一下就结束了	12
为何在 scanf 打断点后，程序停下来，再单步执行，没有动静了.....	12
如何单步执行.....	13
如何查看变量或某个表达式的值？	13
在 watch 中如何用 16 进制显示某一项变量或表达式	14
内存查看窗体.....	14
不用 ascii 表，用内存查看字符的 ascii 码值	15
用内存窗体证明“A”和'A'的区别	16
用内存窗体定位一个 scanf 错误	17
内存窗体可修改内存的值.....	17
利用内存窗体发现 a,b,c,d 的规律	18
用调试分析错误.....	18
用二分法查看编译错误.....	19
字符串 0 结尾的 bug.....	19

为什么 vc6 打开的文件却不能编译

因为你是打开的该文件，而不是将其加入到 ide 中。如果添加一个新文件，[请参见](#)
如果添加一个已有文件，[请参见](#)

为什么一个 vc6 工程中有两个 main 函数不能编译成功？我想写两个程序怎么办？

因为一个工程代表一个程序，一个程序只有一个入口点。Main 是入口函数，如果有两个 main，到底哪个才是入口能？

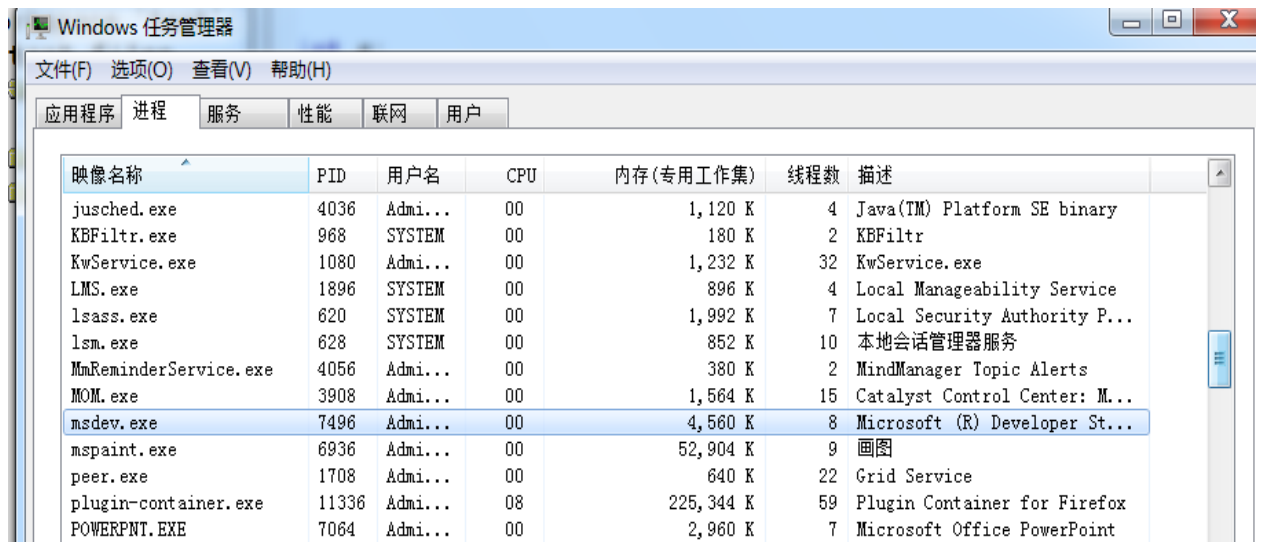
如果想编两个程序，那么生成两个工程即可。[见工程生成](#)。

为什么 v6 不能编译程序，生成信息说 compile 和 link 都正确，到了 link 生成 exe 时，却报错无法生成该 exe 文件

因为，你可能没有关闭该 exe 程序。关闭重编译即可。如果找不到该 exe 是否还在执行就[用任务管理器查看该进程并关闭](#)。

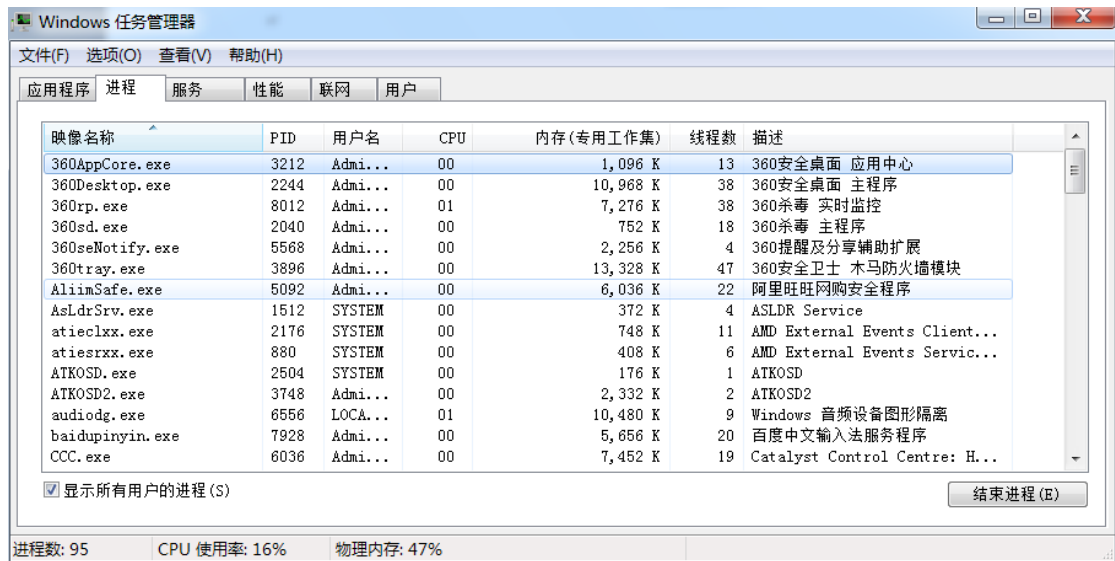
为什么 Vc6 在编译时，下方消息窗体显现了 compiling...或 linking...后就死了，没有反应了，怎么办

编译环境出问题了，[用任务管理器](#)关闭 vc6 进程，其名字是 msdev，如下图，鼠标左键单击选中它，结束该进程。



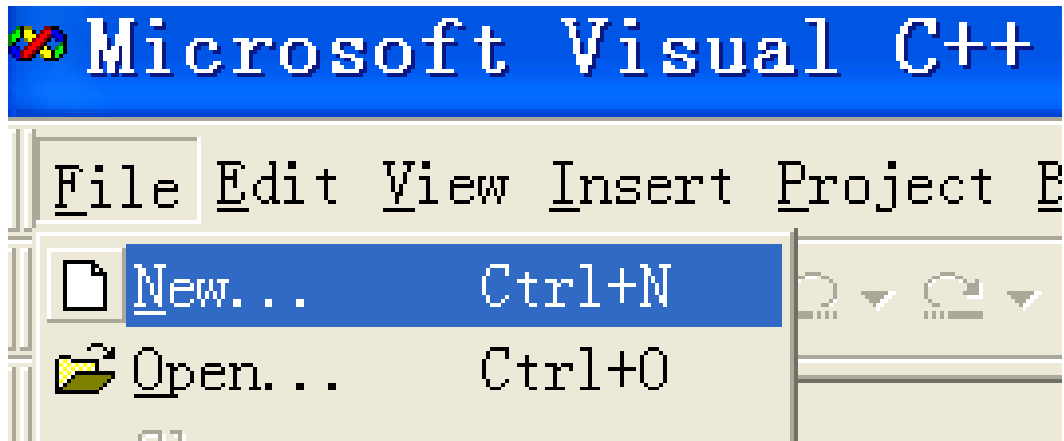
如何用任务窗体关闭进程

同时按照 **ctrl,alt** 和 **delete** 键，win7 上点击界面中“启动任务管理器”选项。将弹出如下窗体

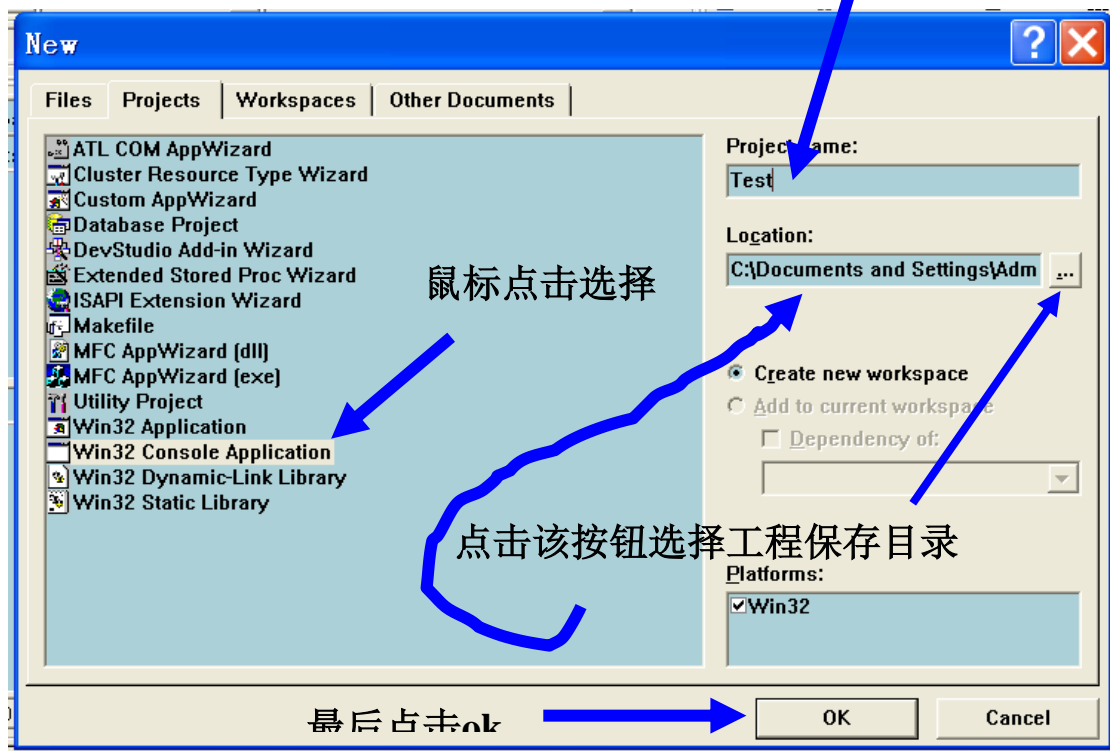


选中要结束的某个进程。然后点击右下角结束进程按钮结束之。

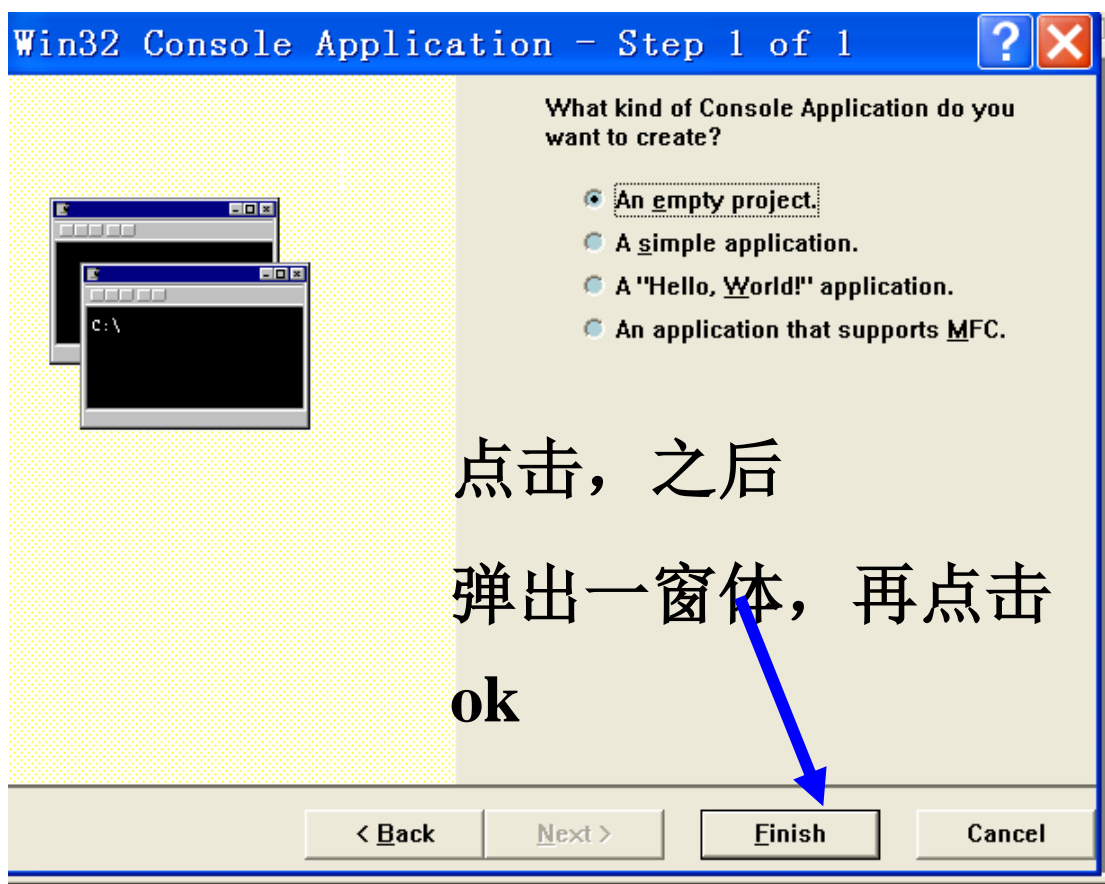
如何生成一个 vc6 工程



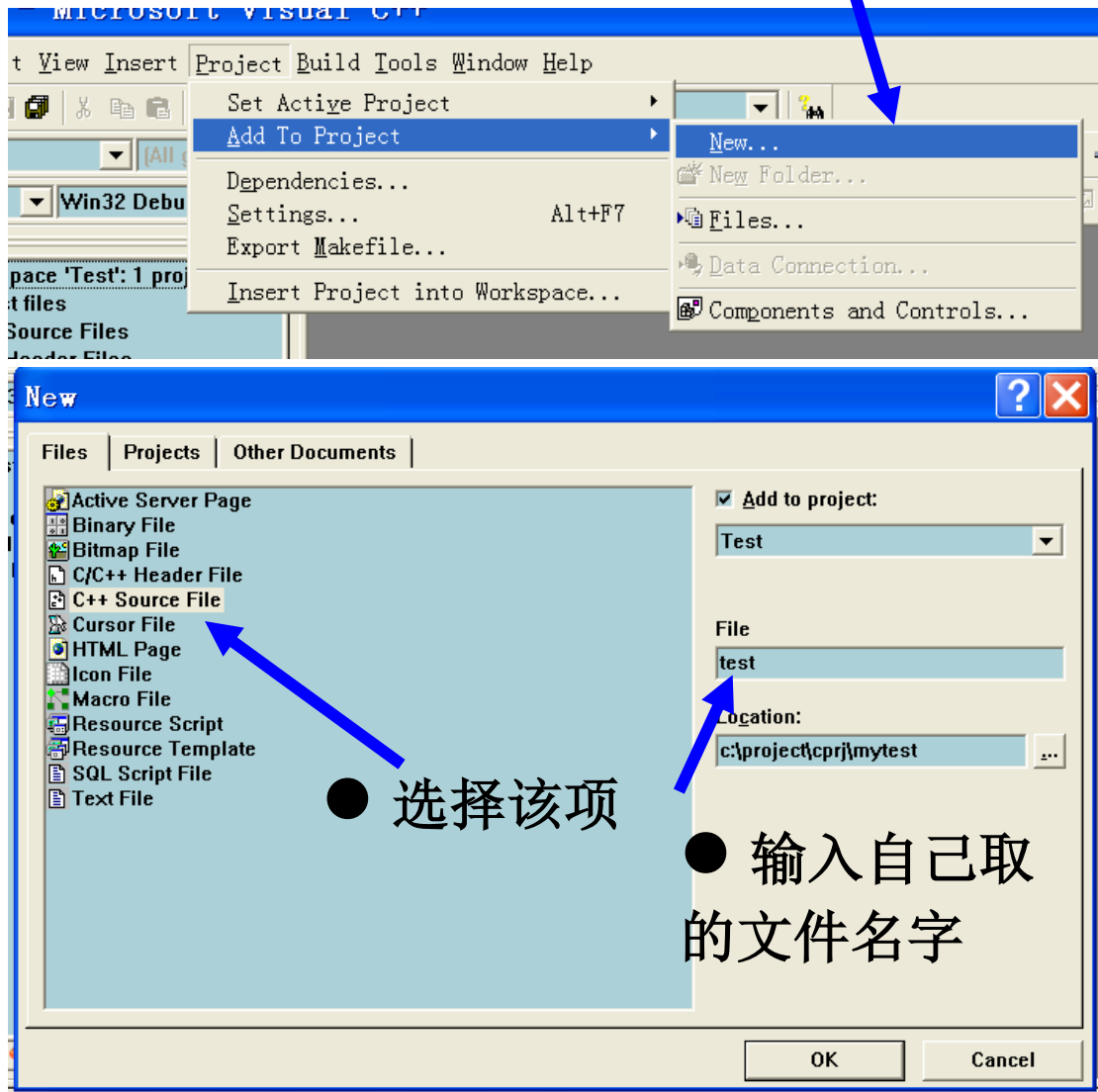
输入自己取的工程名字

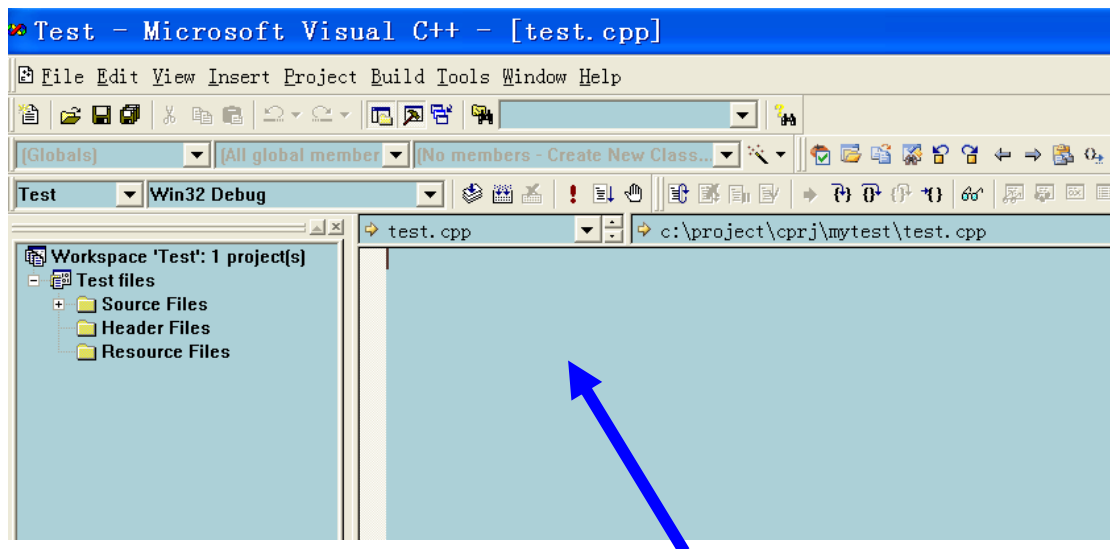


目录选择，点击上图的包含...的按钮后，如下图选择工程所在目录



● 选择该菜单

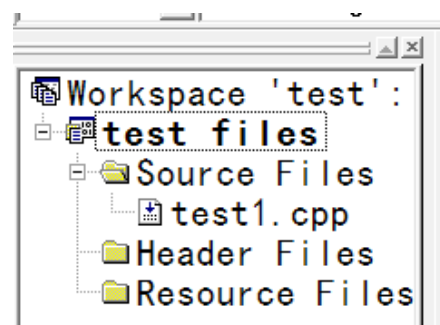
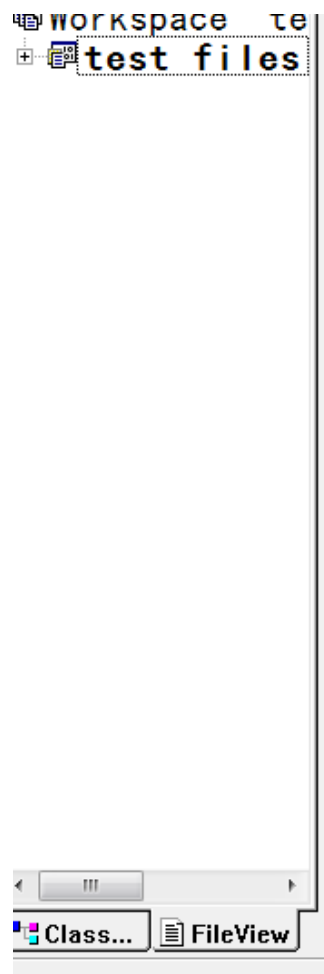




● 在该编辑框中输入程序

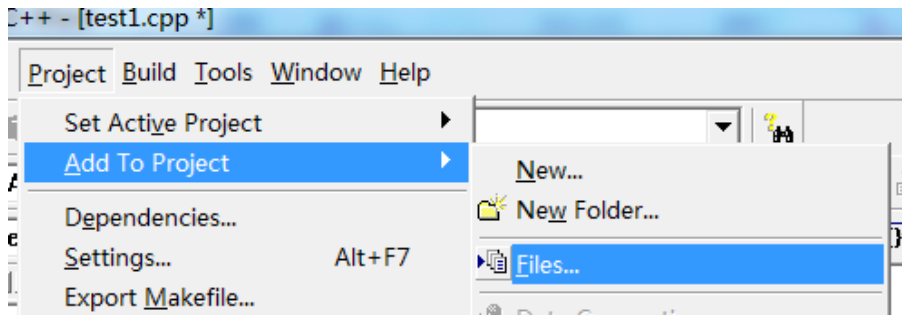
如何查看工程中的文件

在左边面板，鼠标左击选择右下方的 fileview，并点击左边的加号展开工程节点，并点击展开 source files 节点，即可见相关文件。点击文件即可在右边面板编辑文件了。

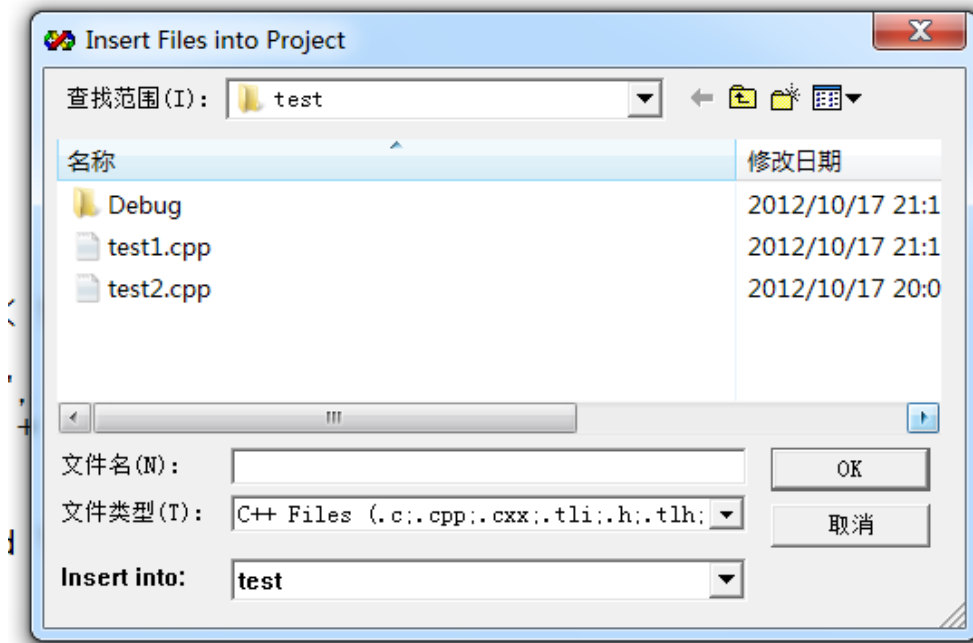


如何添加一个已有文件到工程

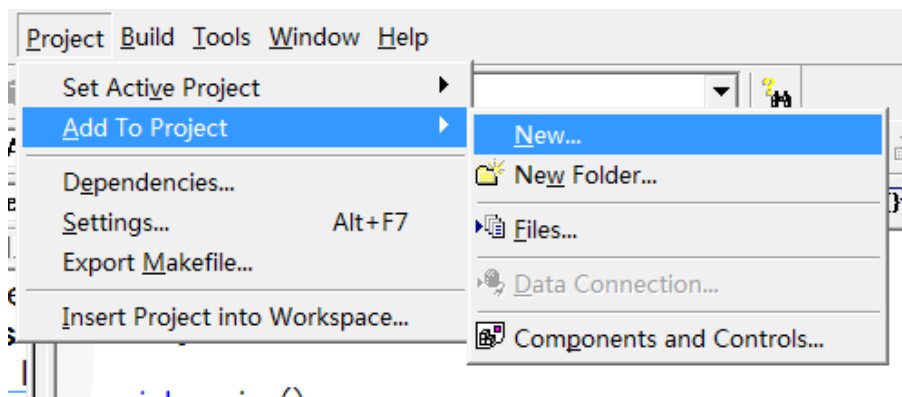
选择 files 菜单



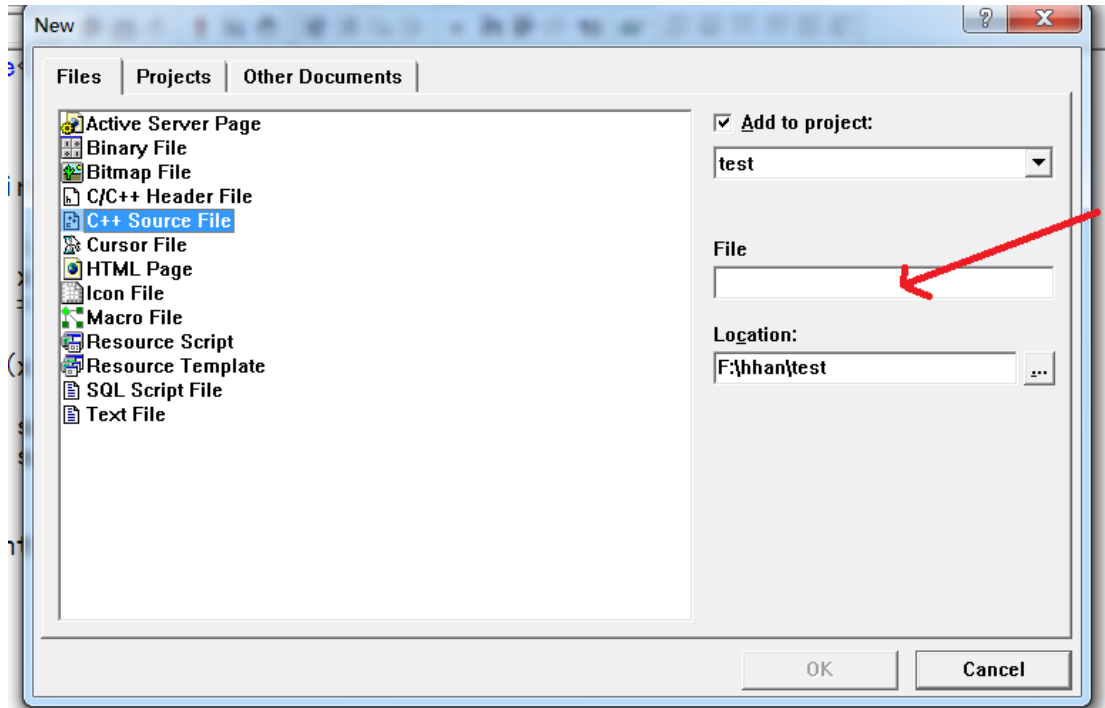
在弹出的文件对话框中选择相关文件（在顶部下拉框中选择目录，在中间用鼠标左键点击选择要添加的文件），然后按 ok 即可。



如何添加一个新文件到 vc6 中



在红色箭头所指处输入新文件名字，并按 ok



为什么 Scanf 执行后程序报错死了

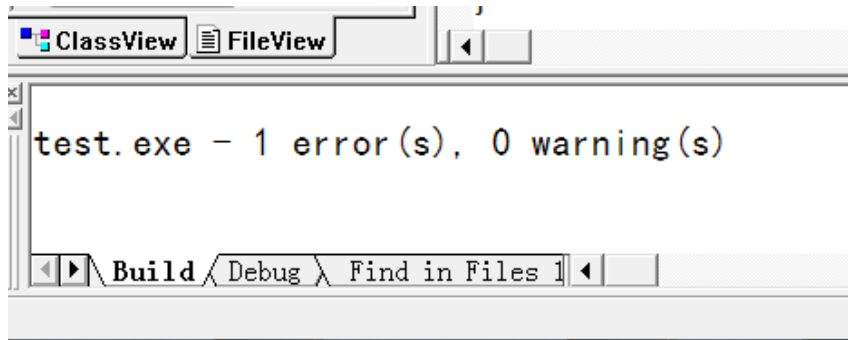
```
int I;  
scanf("%d", I);
```

因为要修改 I 的内容，必须传递给 I 的地址，即 &I。而此时却是将 I 的值传递给 scanf 了。此时 I 是随机数，比如 7777，那么，scanf 获取到键盘值后，就会将它写入地址为 7777 的内存。如果该内存不存在，就要出错。正确做法

```
int I;  
scanf("%d", &I);
```

如何通过编译器的编译消息框发现语法错误

编译出错后，我们会看见下方如下信息



拨动滚动条查看上面信息，加下图

```
test1.cpp
F:\hhan\test\test1.cpp(9) : error C2143: syntax error : missing ';' before '='
Error executing cl.exe.
```

说明 test1.cpp 的第 9 行出错了，错误如行中信息所示。双击上图中 error 所在行将跳到出错的行，如下图

```
int g,

void main()
{
    int i, sum;
    int x;
    sum := 0;

    for(x = 0; x < 10; x++)
    {
```

我们发现 sum 后的等号多了一个“：”

如何发现代码中混入了中文信息

比如我们在代码非注释部分敲入了中文的分号，逗号等，很不容易发现，但这是不允许的。编译器会报错。因为中文字比因为选择多，一个字节（256 种选择）无法代表几万个字，所以中文字由两个字节组成。如上面小节的错误，将 sum 后的冒号变成中文的冒号，编译错误是：

```
test1.cpp
F:\hhan\test\test1.cpp(9) : error C2018: unknown character '0xa3'
F:\hhan\test\test1.cpp(9) : error C2018: unknown character '0xba'
Error executing cl.exe.
```

这时还是指示 sum 所在行出错了。而提示有两个不识别的字符 0xa3,0xba 其实这两个字符就代表中文冒号。出现这种“unknown character '0x..'”信息，一般就说明可能混入了中文字符。

如何打/取消断点

断点，即是调试方式运行时，执行到断点所在行，程序将暂停，我们可以分析其状态。将光标放到要打断点的行，按 f9 即可。取消也按 f9

如何让程序可以调试方式运行

按 f5 运行即可。

为何不能在断点停下来

可能你是 ctrl+f5 运行的，或你点击了工具条上的惊叹号按钮运行。这不是调试运行，所以不会停止。用 f5 运行即可。

为何我不能看到输出的结果，程序闪了一下就结束了

因为在 f5 方式下运行，程序执行完毕自动就结束了。如果你只是打印了信息就退出，那么，非常快，程序打印完毕就退出了，所以只看到了黑色的 dos 界面闪了一下。

解决办法：1，用 ctrl+f5 运行。这时，程序会在最后退出前停下来等待你的键盘回车。

- 2 在 dos 界面中执行该程序
- 3 在程序最后打个断点，f5 运行，停在断点后，切换到 dos 界面查看结果
- 4 在程序最后加一个 `getchar`，这样就会停下来，等待你回车输入

为何在 scanf 打断点后，程序停下来，再单步执行，没有动静了

因为这时已经执行了 `scanf`，而 `scanf` 需要从黑色的 dos 窗体中接收键盘输入。这时我们应该切换到 dos 窗体，输入数据，回车，自然又在 v6 中可调试了。

如何切换到 dos 窗体？鼠标点击 dos 窗体的图标即可

如何单步执行

单步分为 step into 和 step over。

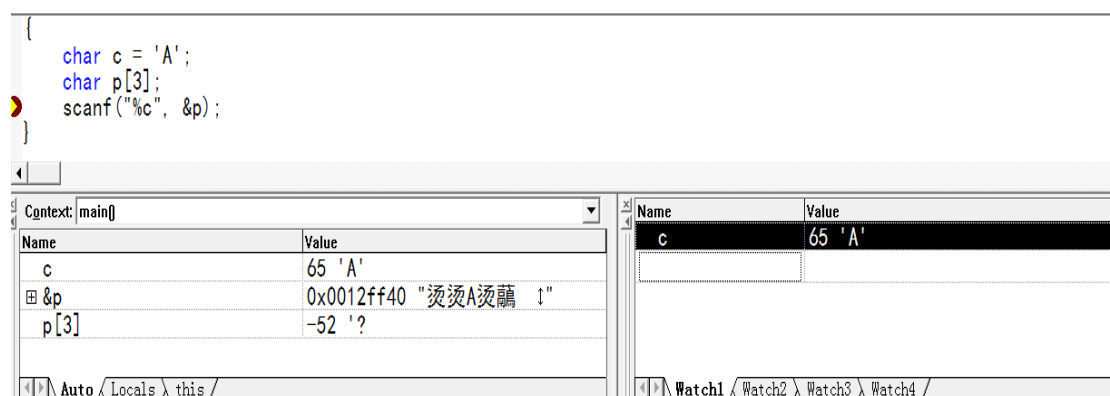
前者遇到函数会跟踪进函数，按 f11 即可

后者遇到函数不会跟踪进函数，而是执行下一句语句，按 f10 即可。很多时候，你认为被调用函数不会有问题，就按 f10，不跟踪进去，这样速度快

遇到诸如 scanf 这样的语句调用，不要按 f11，否则就跟踪进 scanf 了。

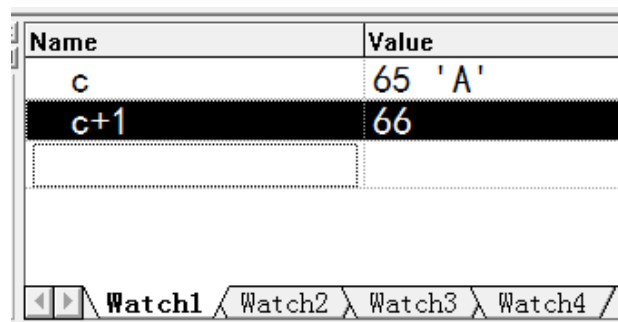
如何查看变量或某个表达式的值？

当程序在调试方式下运行，停在某个断点时，我们可用 watch 查看所有合法的表达式和变量。程序没执行，没停在某个断点，是不可能 watch 的。



上图中，我们在左下角的 watch1 中点击第一列 name，在第一行输入要观察的变量 c，即可看到 value 列给出了其值为字符 A，且 ascii 值为 65.

我们也可以输入任何合法的表达式，比如 c + 1



输入不合法的表达式就会报错，比如 c 1，因为没有这个变量就会报错。

Name	Value
c	65 'A'
c+1	66
<div> <div>◀ ▶</div> <div>Watch1</div> <div>Watch2</div> <div>Watch3</div> <div>Wat</div> </div>	

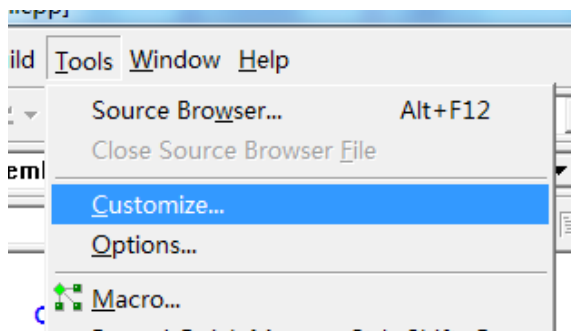
watch 可有多观察的 page，比如上图的 watch1，watch2...

在 watch 中如何用 16 进制显示某一项变量或表达式

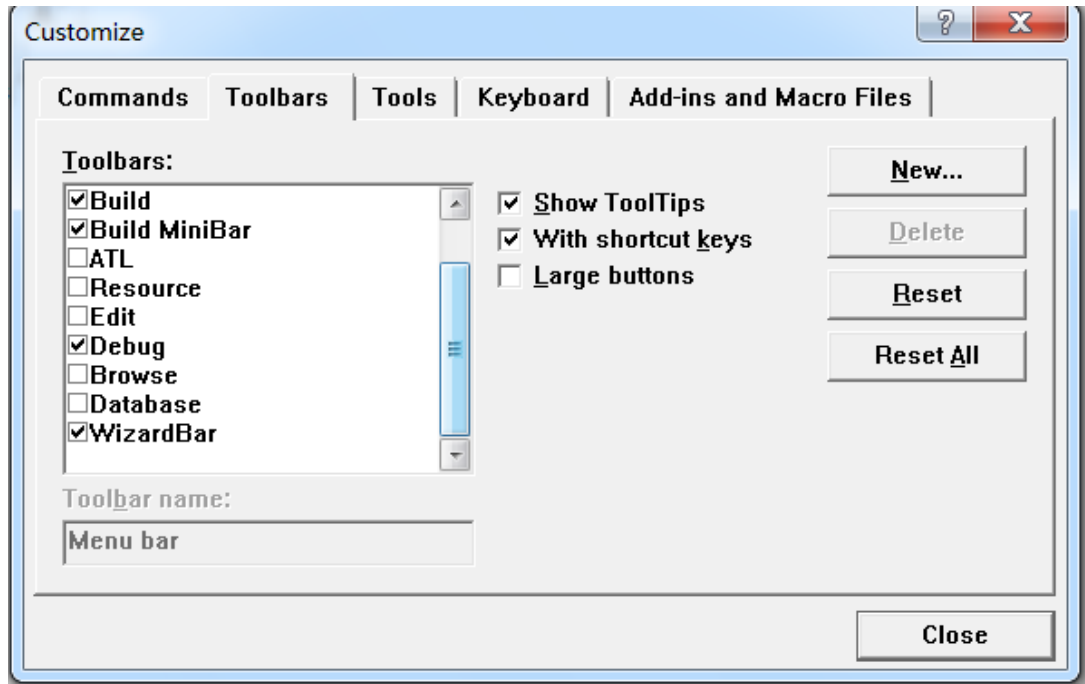
将该项强制为 void *，因为调试器会用 16 进制显示指针类型。比如显示 char c; 的值，可以(void*) c。即可

内存查看窗体

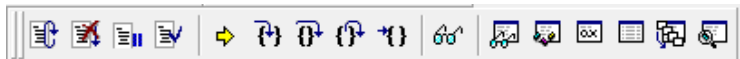
如果没有内存窗体的工具条，选如下选出该工具条，点击 customize 菜单




在如下窗体的 toolbars 列表中勾选，debug 复选框



勾选后，出现调试工具条



当调试运行在断点处停止后，

点击从右边数过来第 3 个按钮 ，得到如下界面

Address:	0x0012ff40	
0012FF40	CC CC CC CC 41 CC CC CC 88 FF 12 00 29 11 48 77 03 00 00 00 50 0E 1C 00 80	烫烫A烫藟...).Hw...P..
0012FF59	0D 1C 00 00 00 00 00 00 00 00 00 00 B0 FD 7F 00 00 00 00 00 00 5C FF衰.....
0012FF72	12 00 00 00 00 00 04 FF 12 00 84 23 48 77 20 F0 49 77 00 00 00 00 94 FF 12#Hw 餓w....

上部 Address 栏中输入要查看的内存地址，回车即可查看。

图中每个字节由 16 进制表示。左边 0012FF40 这些数字是地址

Name	Value
田 &c	0x0012ff44 "A烫藟 ↓"

然后在内存窗体地址栏输入该地址 0x0012FF40，回车

不用 ascii 表，用内存查看字符的 ascii 码值

```
void main()
{
    char c = 'A';
    char p[3];
    scanf("%c", &p);
}
```

在 scanf 打断点，运行停止后，现在 watch 中查看 c 的地址,即&c。获得地址为 0x0012ff44.

Address:	0x0012ff44
0012FF44	41 CC CC CC
0012FF5D	00 00 00 00

注意地址为 0x0012f44 的内存值为 41.这是 16 进制，0x41 就是 65.如果在 watch 中输入 c 的值，可证明这点。

用内存窗体证明”A”和'A'的区别

程序如下，调试，从 watch 获得 c 和 p 的地址，分别为 0x12ff44 和 0x0012ff40

<pre>void main() { char c = 'A'; char p[3] = "A"; }</pre>	
Context: main()	
Name	Value
p[3]	-52 '?'
Name	Value
&c	0x0012ff44 "A烫藹
p	0x0012ff40 "A"

先看 C 中的值，在内存窗体地址栏输入 0x12ff44

Address:	0x0012ff44
0012FF44	41 CC CC CC
0012FF5D	00 00 00 00

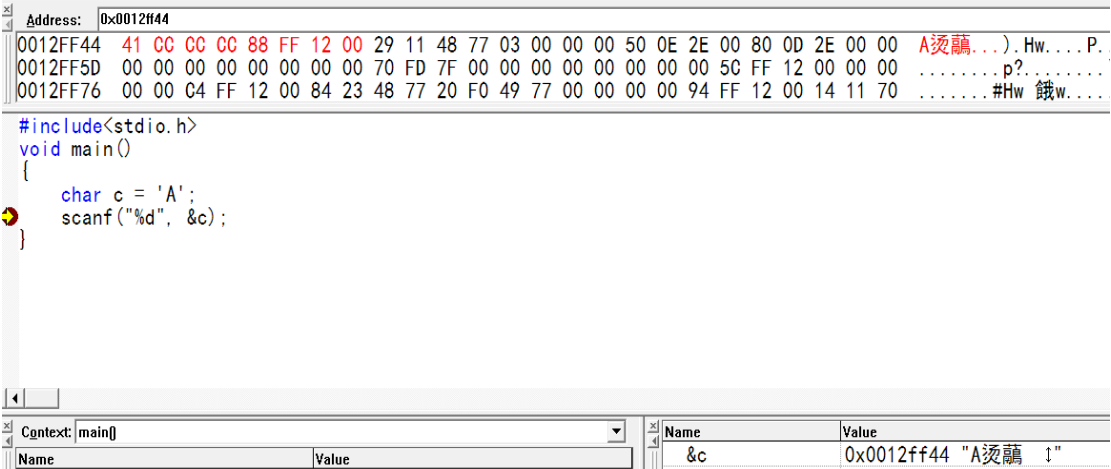
可见只有 12ff44 是 0x41，后面并没有 0 作为结尾字符

再看 p 的内容，在内存窗体地址栏输入 0x12ff40

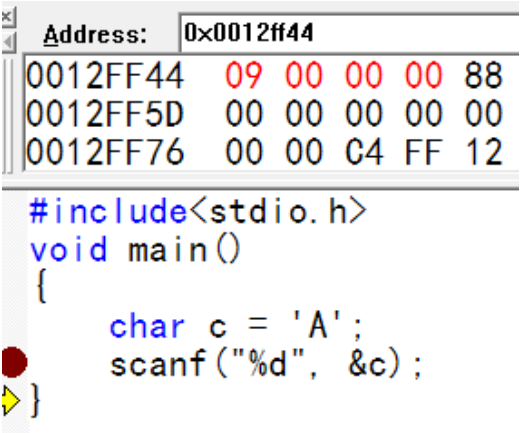
Address:	0x0012ff40
0012FF40	41 00 00

可见 41 后面是 00，字符串形式的 A 实际占用了 2 个字节，最后一个为 0.

用内存窗体定位一个 scanf 错误



这个错在哪里呢？调试执行 `scanf` 前，查看内存 `c` 的内容。
执行后，输入 9，再看内容



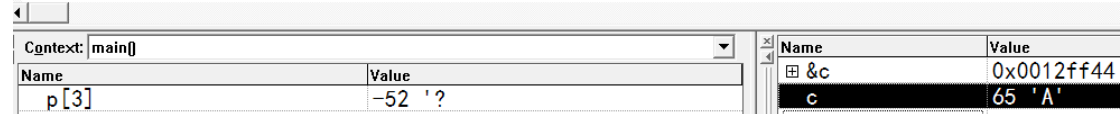
地址 0x12ff44 就是 `c` 的地址，其中的值变成了 09，不仅如此，后面 3 个字节的值也变成了 00.为什么会这样？因为 `d` 代表整数，是 4 字节，那么输入的 9 被翻译成整数表示，就会写 4 个字节，这时，`c` 本来只占一个字节，`scanf` 就导致越界写了，越了 3 字节。
如果输入的是 66，请问在 `scanf` 后加入 `printf("%c", c);`将打印什么值。

内存窗体可修改内存的值

在内存窗体中，将光标点在要修改字节上，直接按键盘输入即可。

利用内存窗体发现 a,b,c,d 的规律

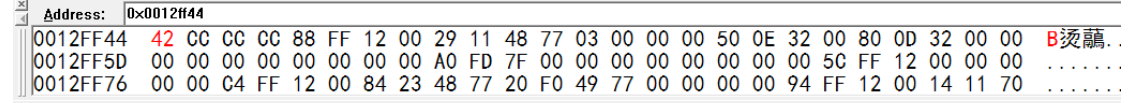
```
void main()
{
    char c = 'A';
    char p[3] = "A";
}
```



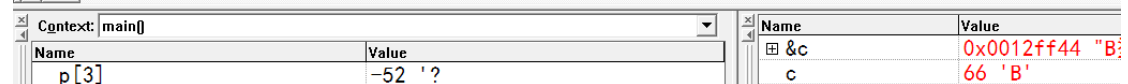
Name	Value
&c	0x0012ff44
c	65 'A'
p[3]	-52 '?'

然后打开内存窗体，查看 c 中的内容，即输入 c 的地址 0x0012ff44.

将内存中的值 0x41 改成 0x42.



```
void main()
{
    char c = 'A';
    char p[3] = "A";
}
```



Name	Value
&c	0x0012ff44 'B'
c	66 'B'
p[3]	-52 '?'

请看有下 watch 中的 c 值，变成了 B。本来 C 的值是 A。

用调试分析错误

只靠看，下面错误很难发现原因：

```
#include<stdio.h>

void main()
{
    int a,b,c;
    scanf("%d&d",&a,&b);
    if (a>b)

    {
        c=a;
        a=b;
        b=c;
    }
    printf("%d%d",a,b);
}
```

而在 main 开始打断点，执行当执行了 scanf 后（输入数据并回车），我们发现程序报错。

很明显是 scanf 执行出错，而 scanf 不会有问题，因此，仔细查看参数，第二个 d 前应该是 % 但错误写为了 &

用二分法查看编译错误

即使从编译错误提示，如果一时看不出哪行代码出了问题（极有可能是上下的其它地方的错）。我们可以删除或注释掉一些语句编译。如果依然错误可以继续扩大删除或注销的语句，直到正确。然后在慢慢恢复删掉的语句，直到出错。这样就能定位了。

字符串 0 结尾的 bug

```
struct Person{
    char name[4];
    int age;
};

void main(){
    Person p;
    scanf("%s", &p.name);
    scanf("%d", &p.age);
    printf("%s\n", p.name);
}
```

如果第一个输入 "hell", 第 2 个输入 18。分析输出结果的原因

答案：

因为第一个输入实际要占 5 字节，最后一个是 0 字节，它录入到 age 的第一个字节，再输入 age 后，内存中的内容实际是 'h' 'e' 'l' 'l' 0x12 00 00 00 这时 printf 打印时会认为 name 字符串的结尾在 0x12 后面那个 0，这时会将 0x12 的 ascii 符号打印出来。