

uC/OS 在 PC 上的移植

成都信息工程学院通信工程系

计算机通信 071 班

作者：宁志强

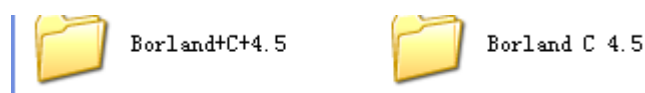
目录

一、Borland C++ 4.5 的安装.....	2
二、TASM5.0 的安装.....	7
三、设置映射路径.....	12
四、Borland C++ 4.5 开发环境.....	15
五、uC/OS 的移植.....	16

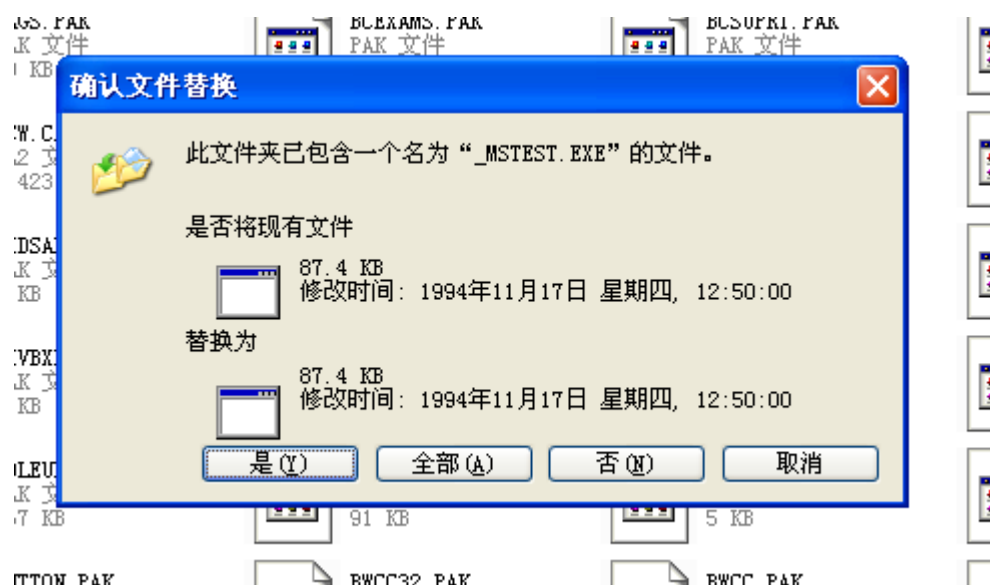
一、Borland C++ 4.5 的安装



从网络上下载 “Borland c++ 4.5” 编译环境，我是从 CSDN 网站上下载的，如上图。



对压缩文件进行解压，解压完成如图。

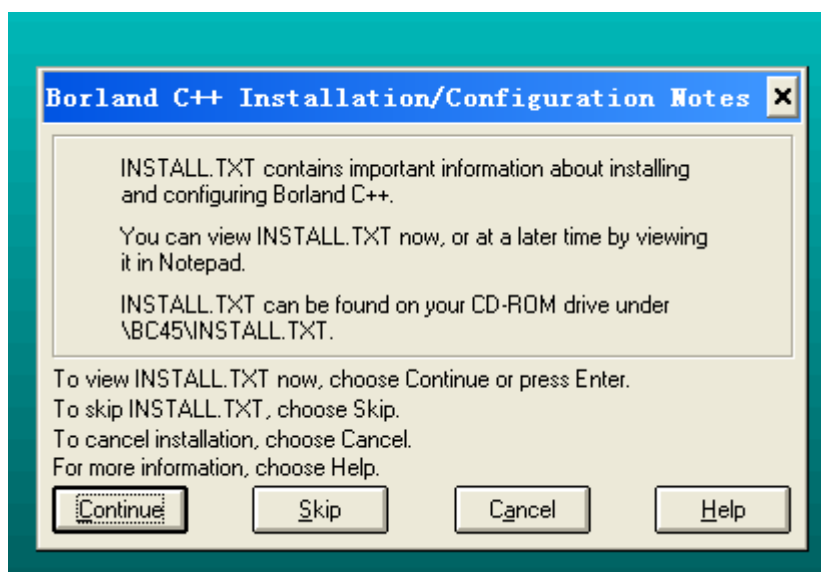


将文件夹 “Borland+C4.5” 中的所有内容拷贝到 “Borland C 4.5” 中，如出现上图提示，则直接点 “全部” 即可。



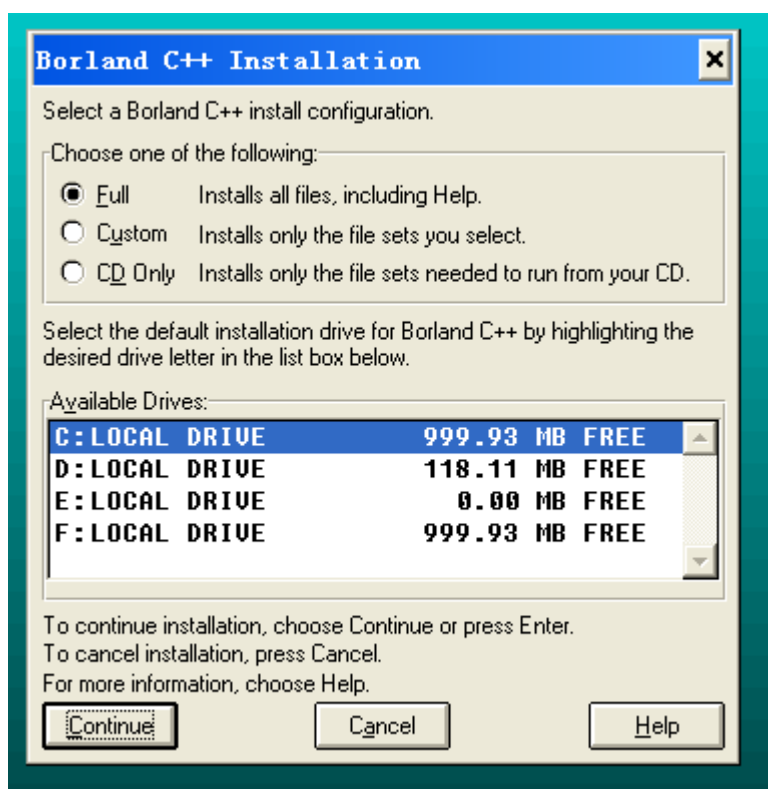
拷贝完成后，在 “Borland C 4.5” 中，找到 “INSTALL.EXE” 文件(如上图)，双击打开。

uC/OS 在 PC 机上的移植



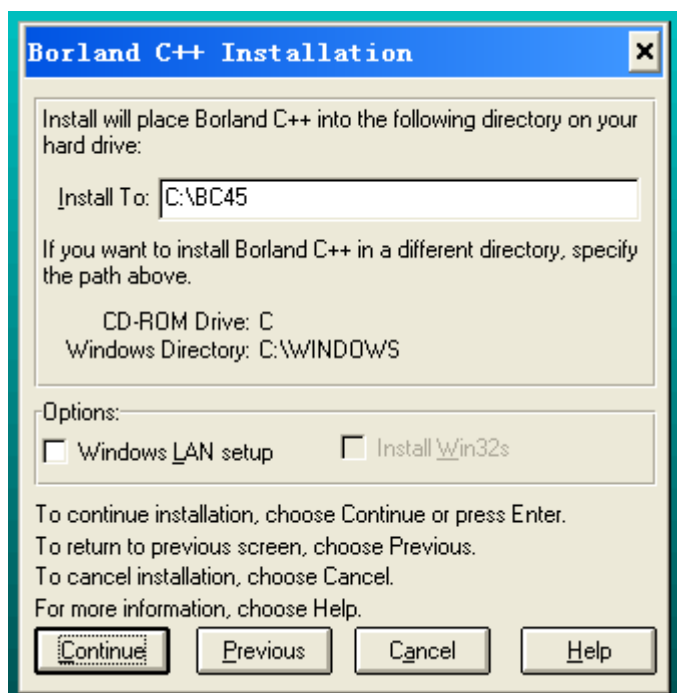
进入安装界面。

点击“Skip”跳过一下文档的说明，进入下一个安装界面。文档没有必要看，是英文的。



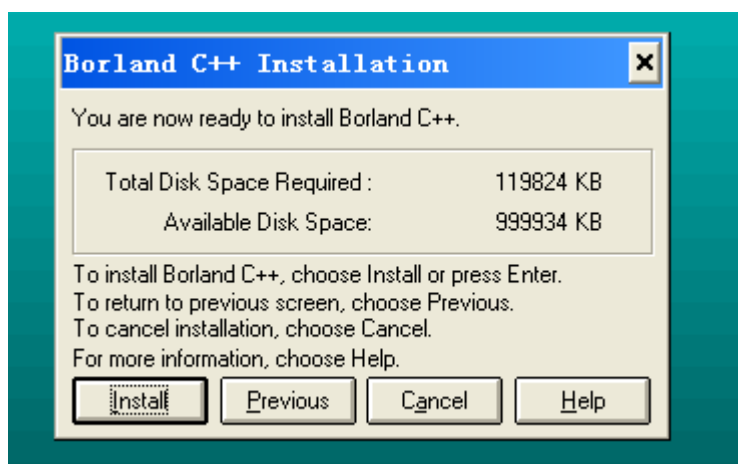
进入上图界面，在“Choose one of the following”中选择“Full”，即完全安装。点击“Continue”进入下一个安装界面。

uC/OS 在 PC 机上的移植



进入此界面，主要是选择安装路径等。

我们选择默认路径，即“C: \BC45”中。点击“Continue”，进行下一步。

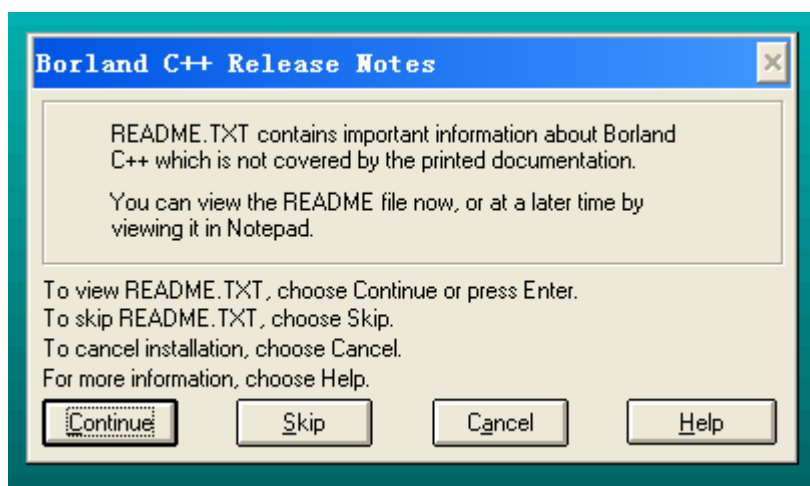


在此界面，我们直接点击“Install”，就直接开始安装软件。

uC/OS 在 PC 机上的移植

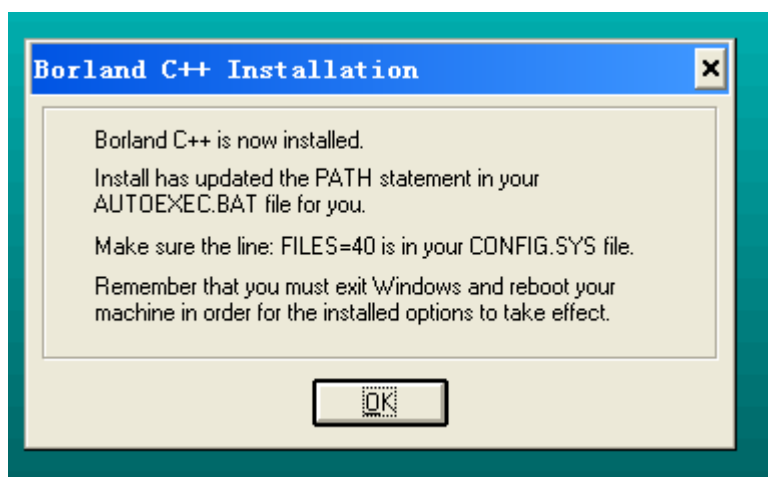


上图，则为正在安装程序。

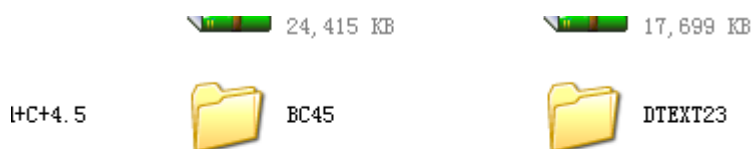


安装完成后，进入此界面，我们点击“Skip”，跳过文档浏览，和上面的文档一样，是英文的。

uC/OS 在 PC 机上的移植



最后，点击“OK”，Borland C++ 4.5 安装完成了。



恭喜你，第一步已经完成。

二、TASM5.0 的安装



从老师课件中下载“Tasm5”压缩包(如上图)。



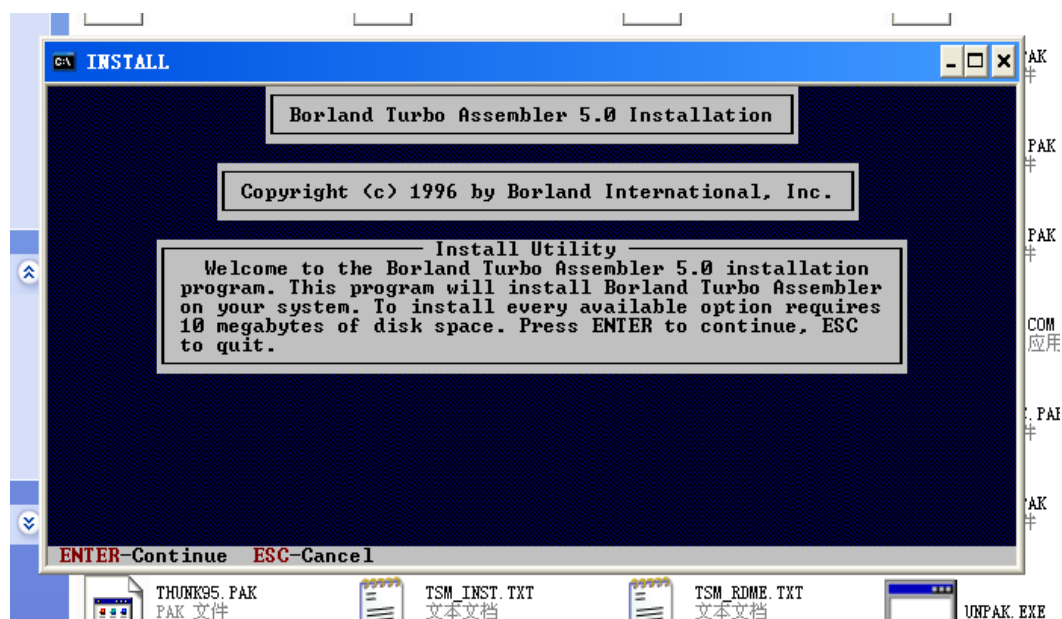
对“tasm5.zip”进行解压(如图)。

uC/OS 在 PC 机上的移植

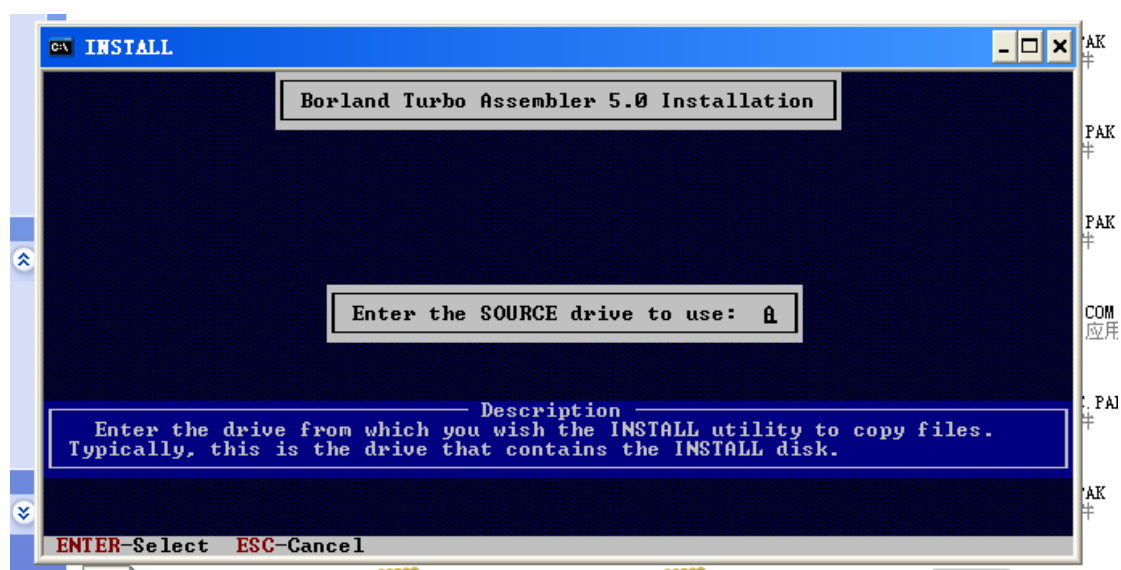


打开“tasm5”文件夹，找到里面“INSTALL.EXE”文件。

双击打开。

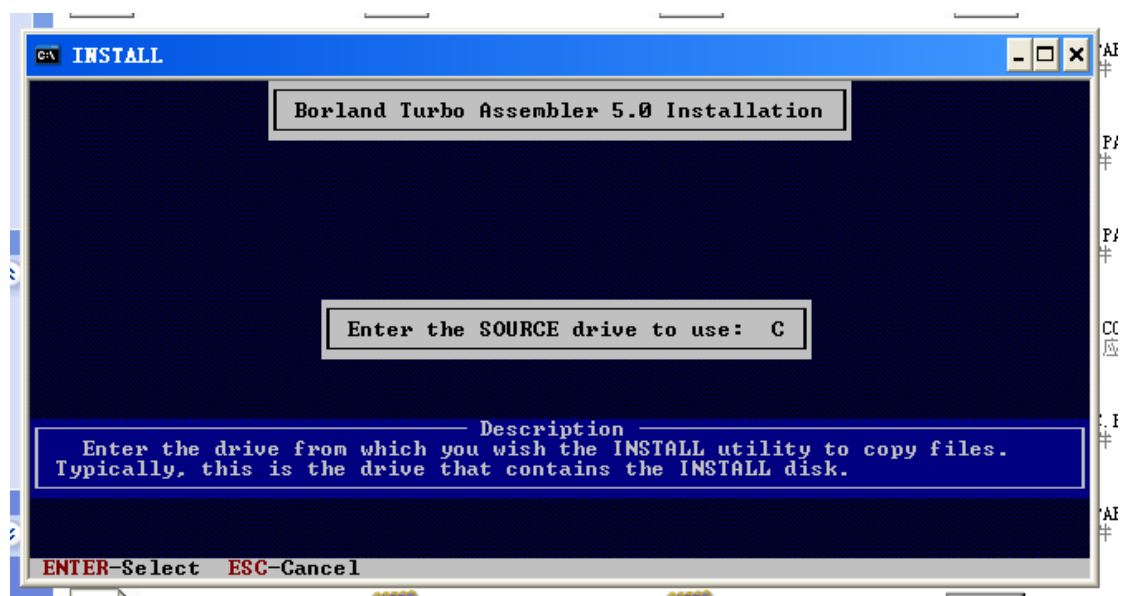


进入安装界面，DOS 界面，这个东西历史悠久啊！按“Enter”键。



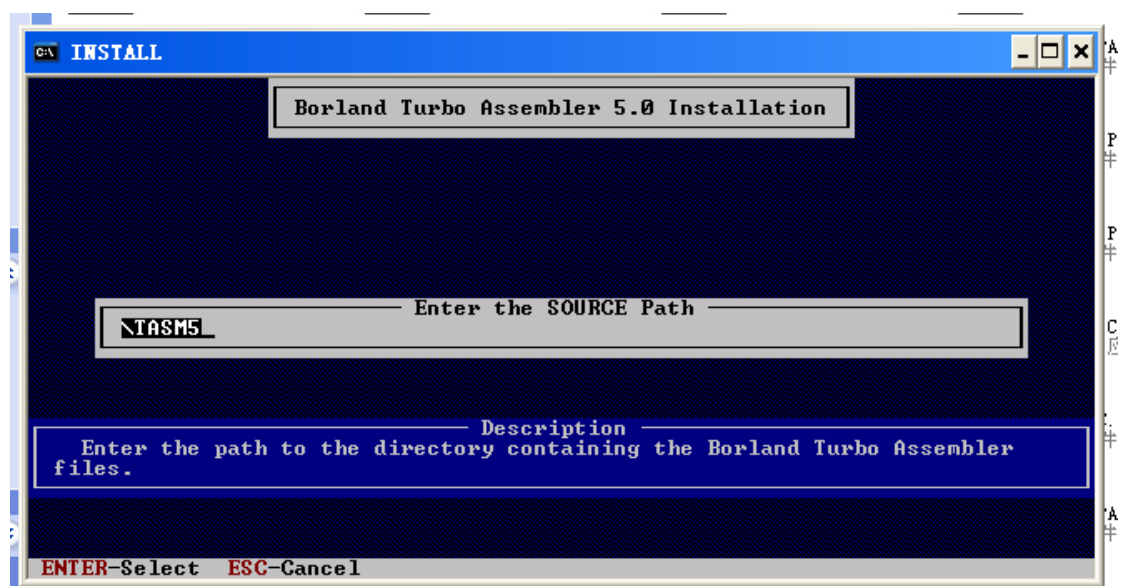
在此界面中，选择你安装源文件的路径，默认是“A”盘。

uC/OS 在 PC 机上的移植



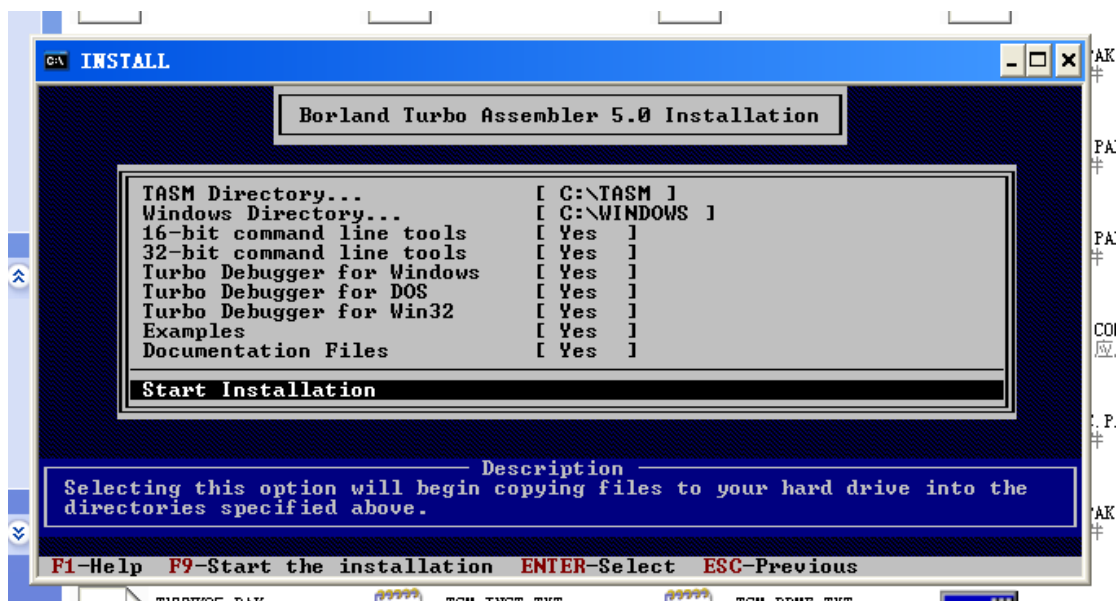
按“C”键，即改变了，盘符。这里我们是把源文件放在C盘下的，如果你放在其他盘符下，自己改就OK。

选择完成后，按“Enter”键。

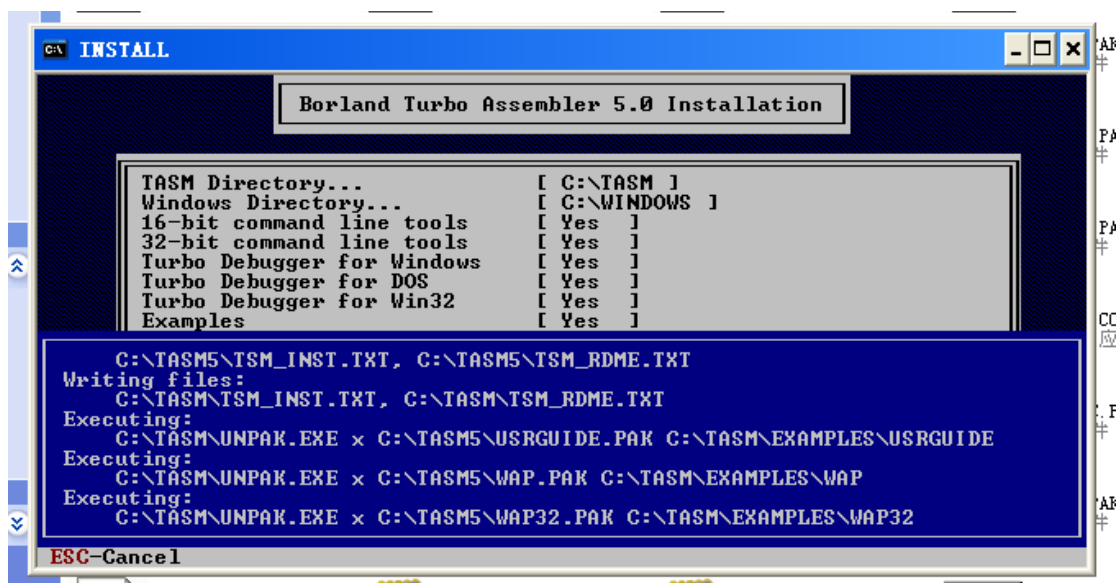


这里它已经自动的现了源文件的路径，我们就直接点击“Enter”键。

uC/OS 在 PC 机上的移植



选择安装路径，我们使用默认路径，其他信息我们都不管，按“向下”键，选择“Start Installation”，点“Enter”键。

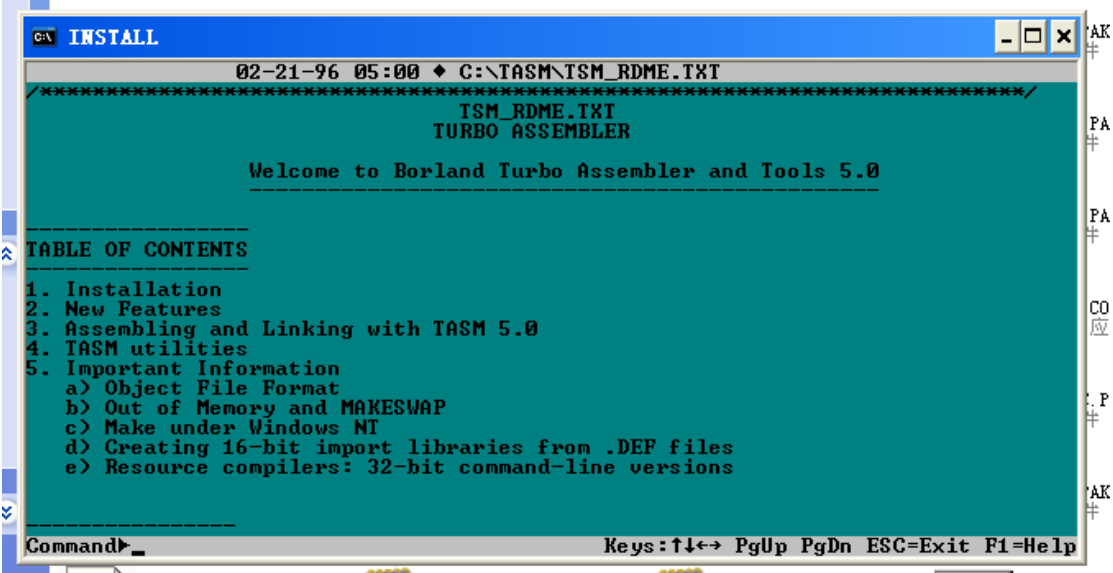


开始安装了。

uC/OS 在 PC 机上的移植



根据自己想不想创建来选择。我选择了“确认”。



安装完成之后的 Message Box。

按“ESC”退出即可。

恭喜你！第二步已经完成。

三、设置映射路径

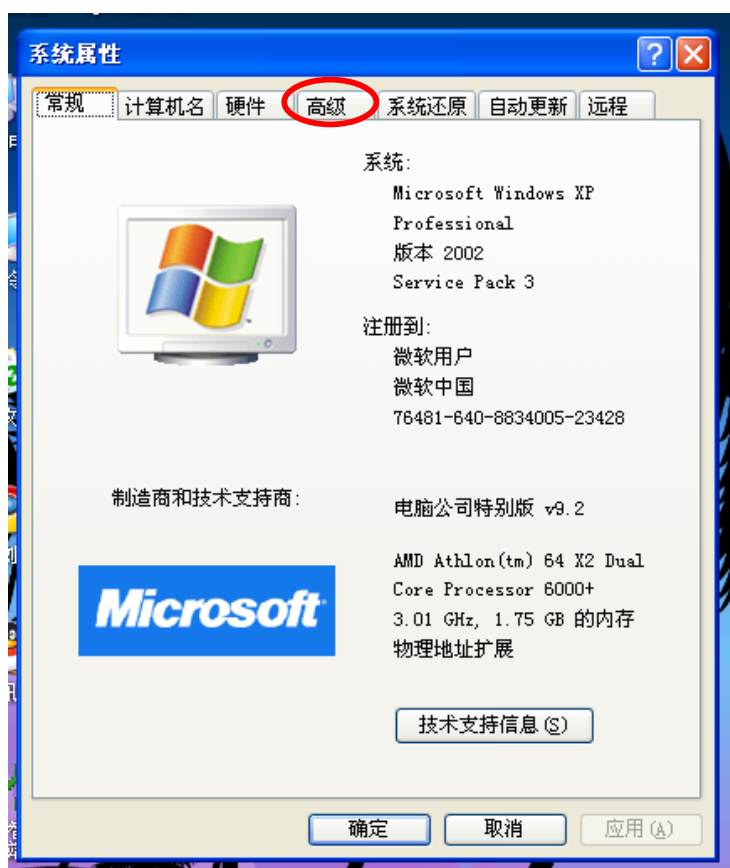


在“我的电脑”上点击鼠标右键。

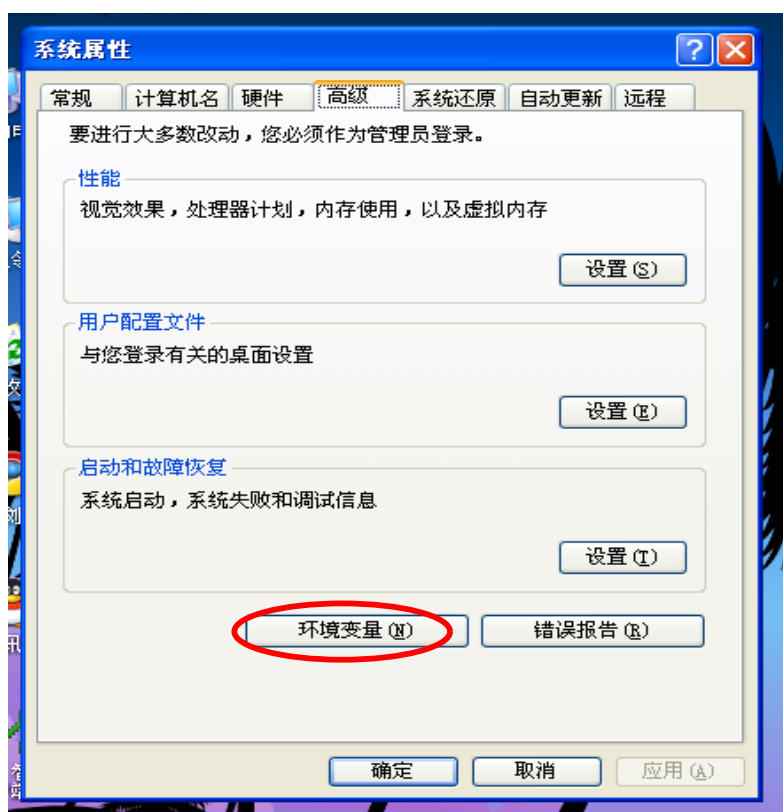


选择“属性”。

uC/OS 在 PC 机上的移植



进行系统属性界面。点击标签“高级”。



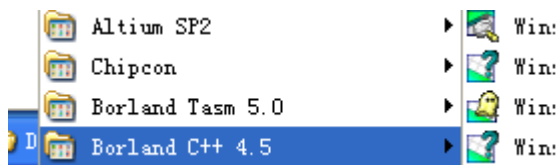
在“高级”界面下，点击“环境变量”按钮。

uC/OS 在 PC 机上的移植

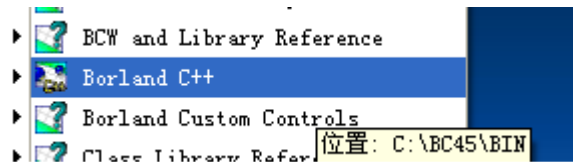
点击“确认”。知道退出设置即可。

这样就完成了环境变量设置。

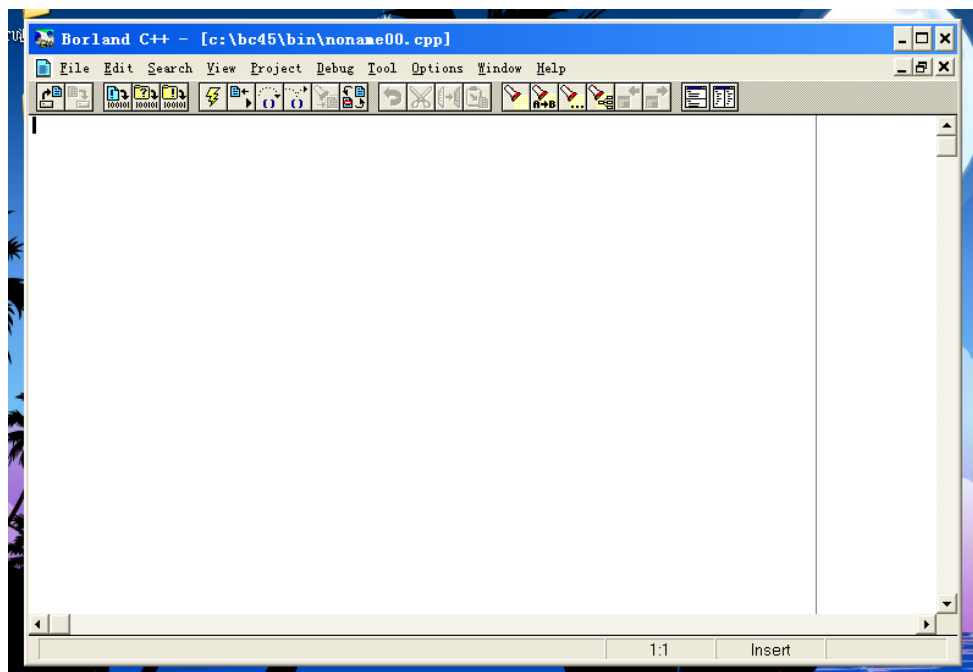
四、Borland C++ 4.5 开发环境



开始菜单-->程序-->Borland C++ 4.5-->Borland C++.



单击，进入开发环境。

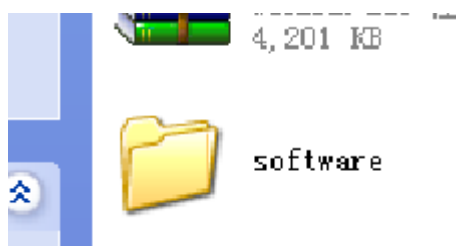


这个就是 Borland C++ 的环境啦！感觉比较古老！

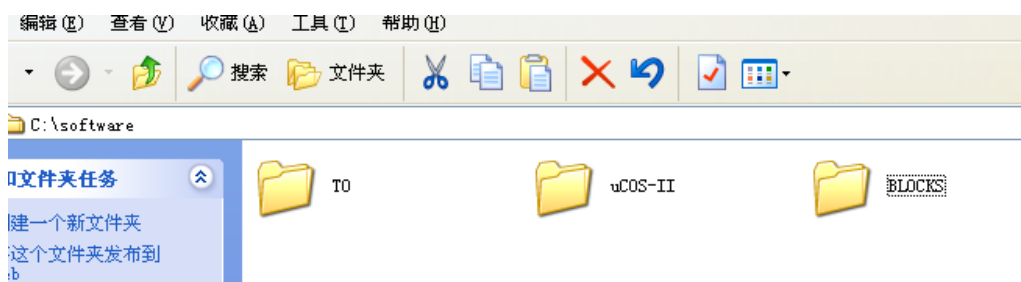
五、uC/OS 的移植



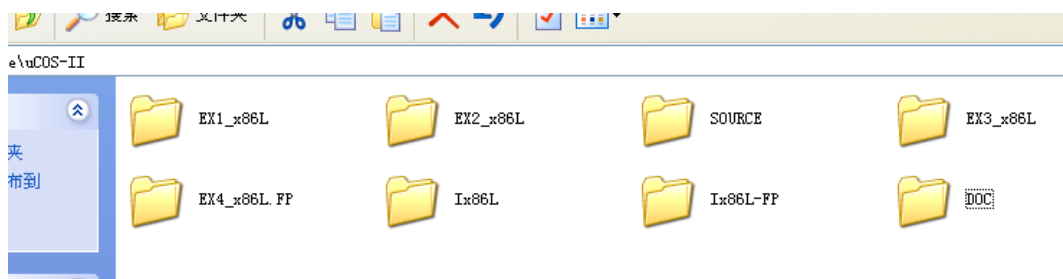
在老师课件中下载代码压缩包 “v2.62.rar”。



在 C 盘根目录下解压，解压后生成一个 “software” 的文件夹。



双击进入文件中，则可看到三个文件夹，“T0” 这次移植没有用到，“BLOCKS” 存放着 PC.C 源代码，主要提供一些显示，键盘等的 PC 机接口。“uCOS-II” 文件夹中则存放着与 uC/OS 相关的文件了。

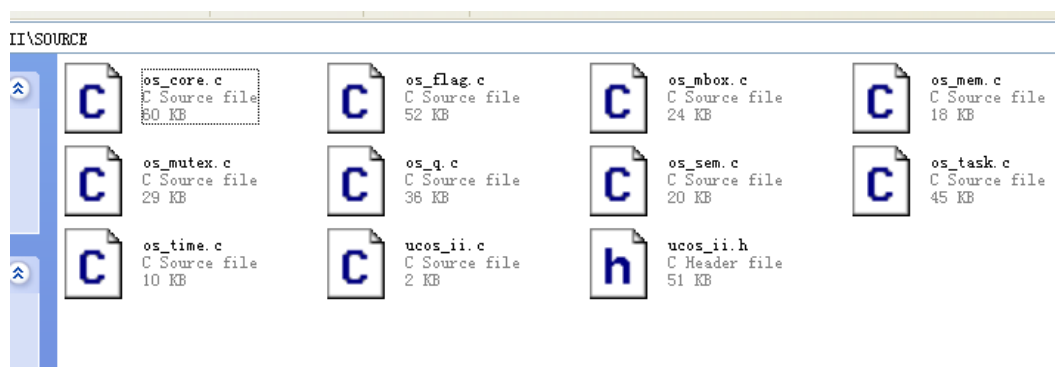


进入 “uCOS-II” 文件夹中，可看到很多的文件，其中只有

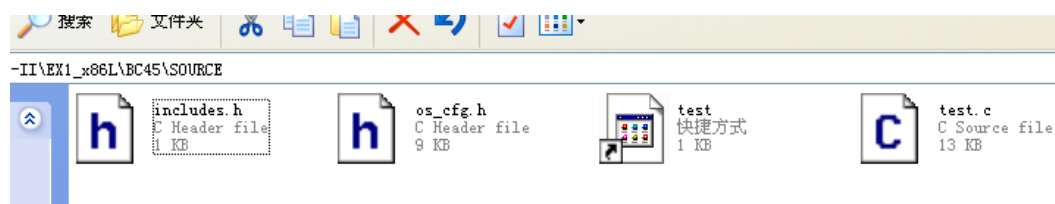
“EX1_x86L” “SOURCE” “Ix86L” 等文件夹在这次移植中使用到了，

uC/OS 在 PC 机上的移植

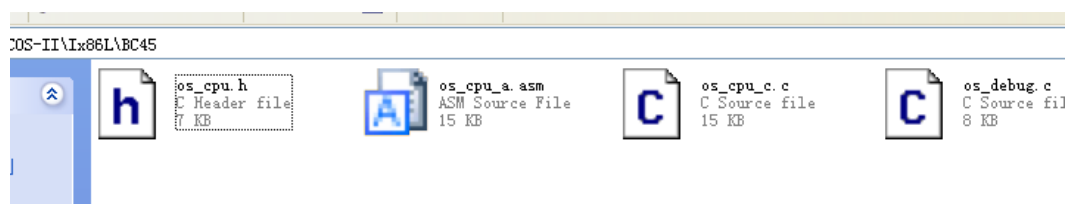
下面我们来一一看看这些文件夹。



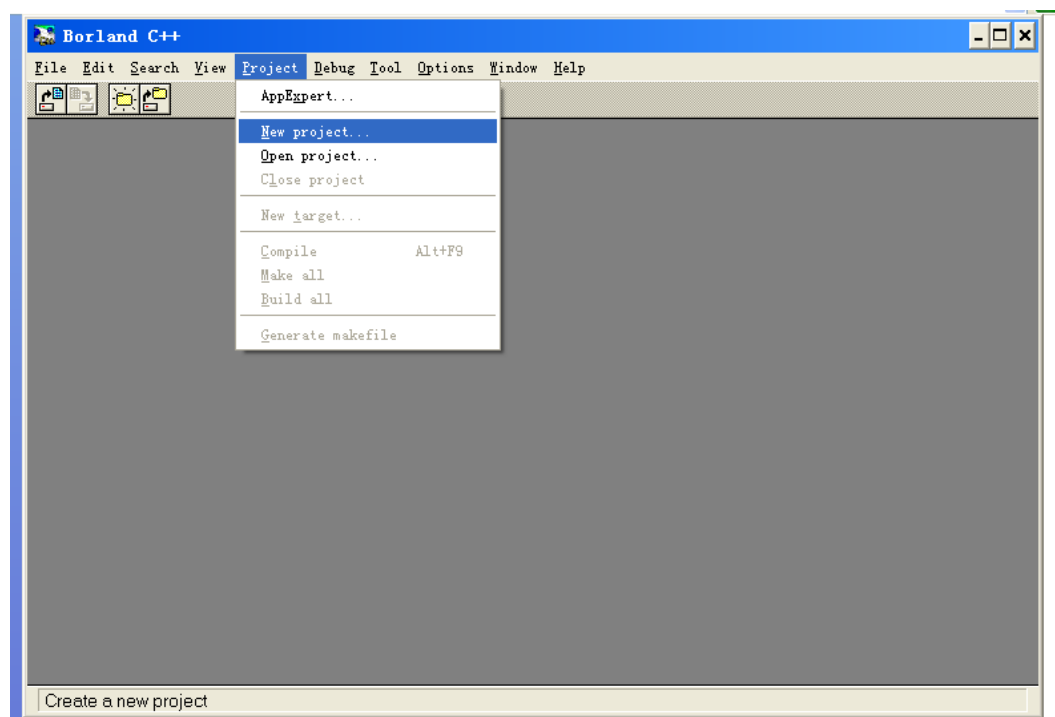
“SOURCE” 文件夹，主要是 uC/OS 操作系统(与硬件无关的)。



“EX1_x86L” 文件夹，主要是用户代码。

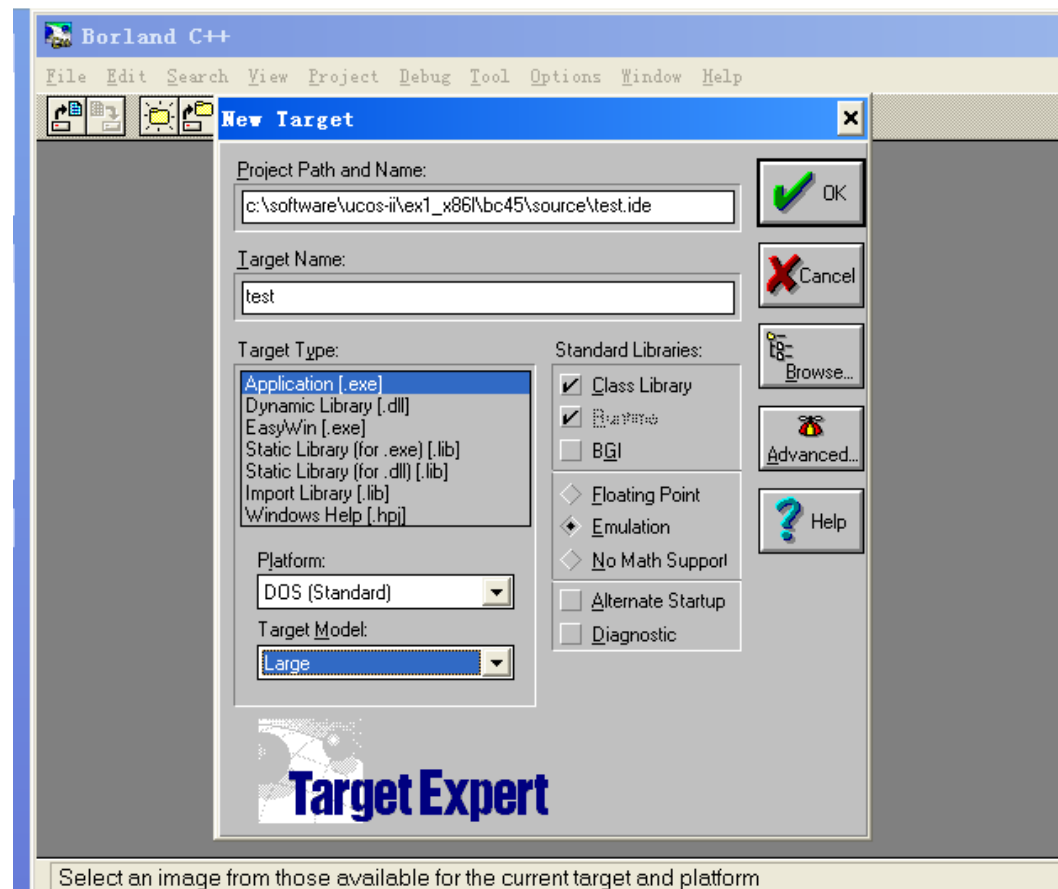


“Ixx86L” 文件夹，主要是和 CPU 相关的文件。



uC/OS 在 PC 机上的移植

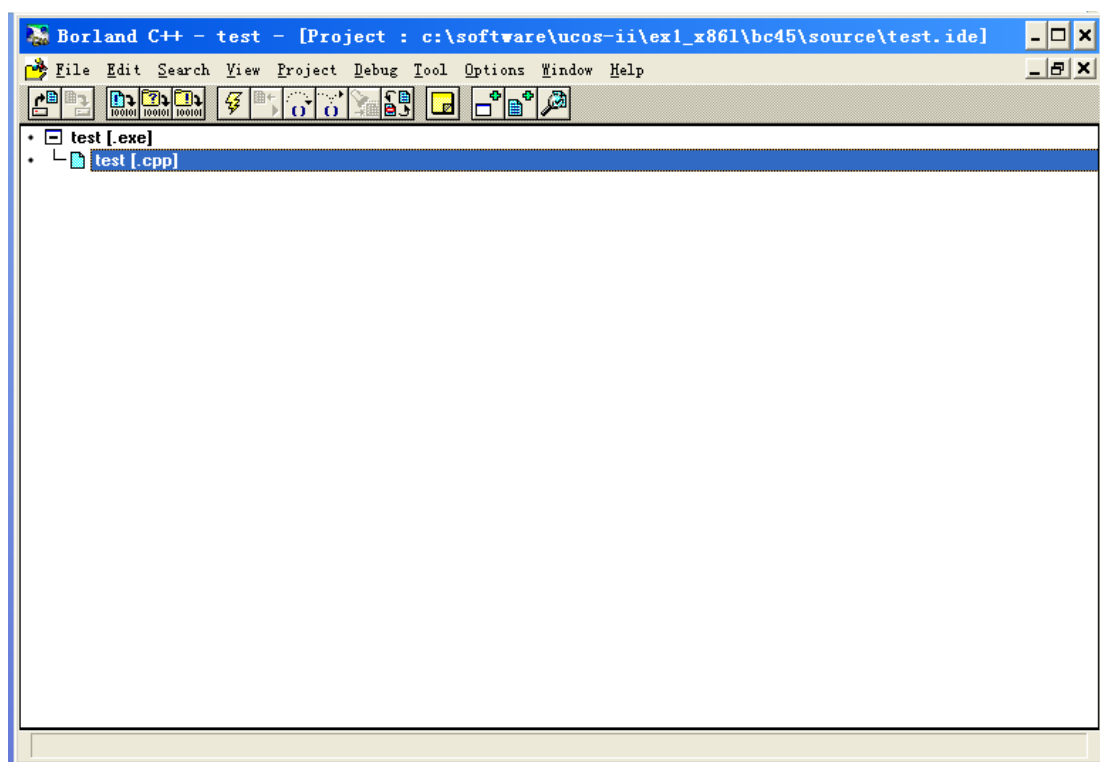
打开“Borland C++”开发环境。选择菜单栏中-->Project-->New Project，新建一个工程。



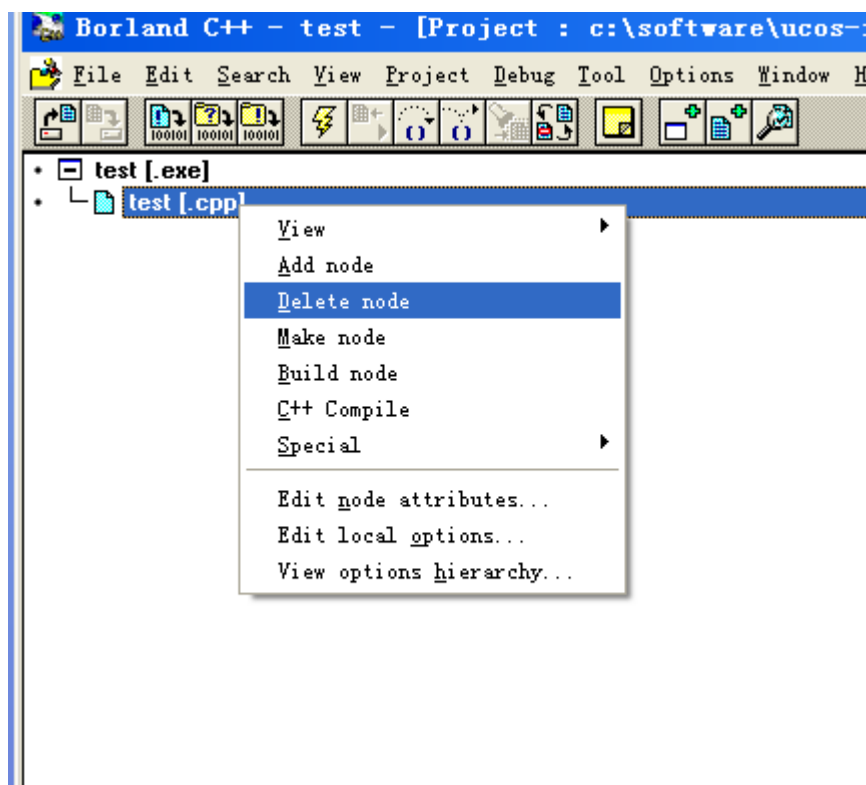
在新建工程界面中进行设置。在 Project Path and Name 中设置为“C: \software\uCOS-II\EX1-x86L\BC45\source\test.ide”。在 Target Type 中选择“Application[.exe]”。在 Platform 中选择“DOS (Standard)”。在 Target Model 中选择“Large”。其他选择都为默认。（默认是将 uC/OS 放在 C 盘根目录下的，放在别的盘符下自己进行修改。）

设置完成后，点击“OK”。

uC/OS 在 PC 机上的移植

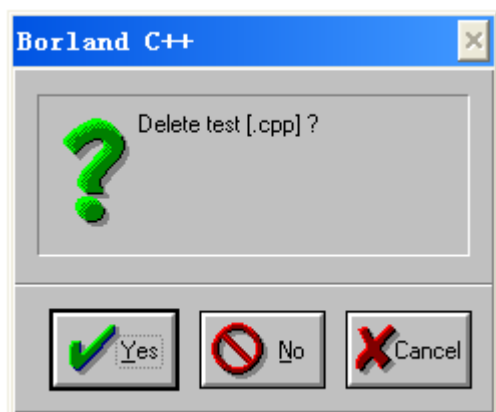


这就是新建的工程。默认包含了一个 “.cpp” 的文件。

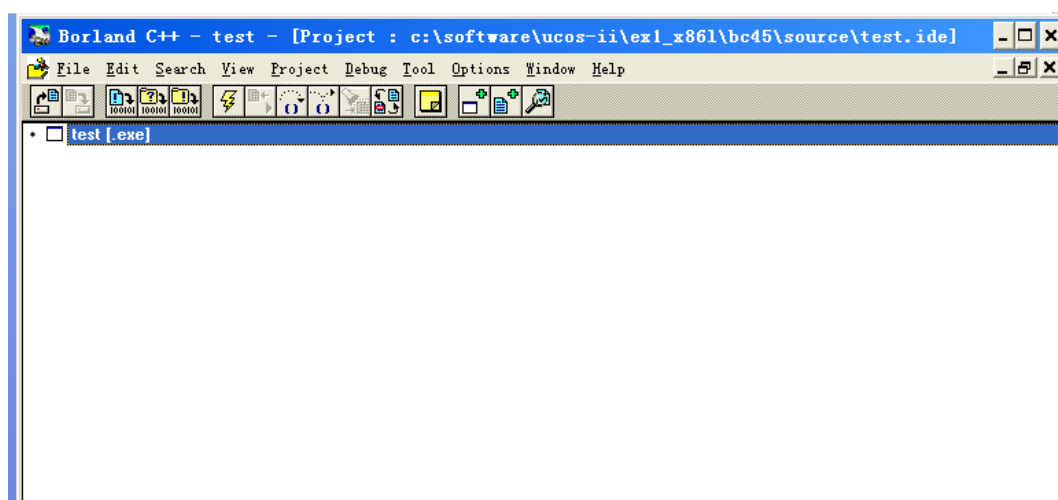


将默认的 “.cpp” 文件删除。在该文件上单击鼠标右键，选择 “Delete node”。

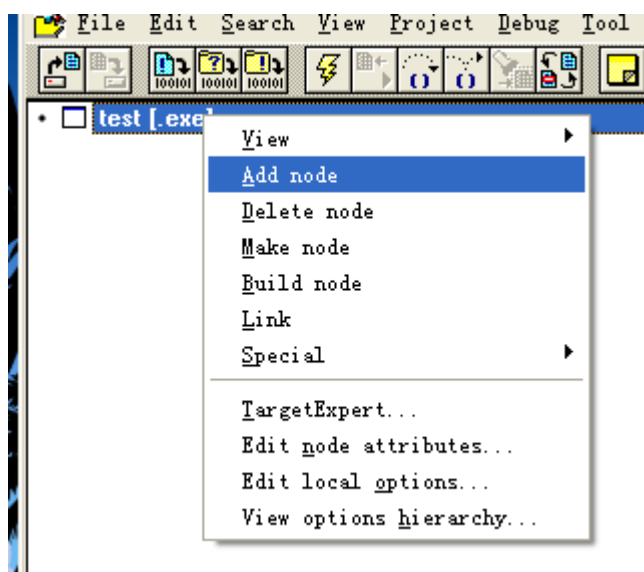
uC/OS 在 PC 机上的移植



确认删除？点击 “Yes”。即可删除。



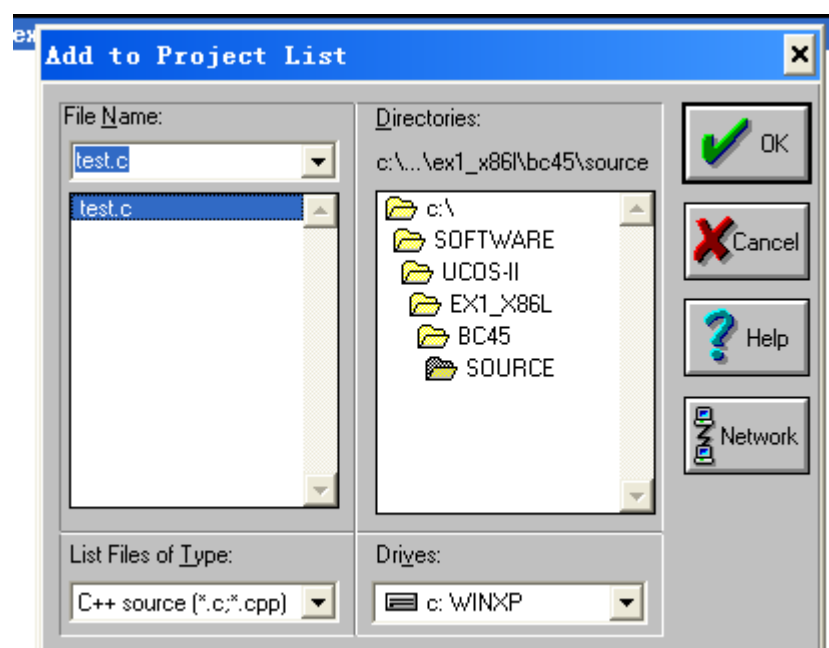
只剩工程了！



开始添加，我们要移植的 uC/OS 文了。

在工程上点击鼠标右键，选择 “Add node”，进行添加。

uC/OS 在 PC 机上的移植

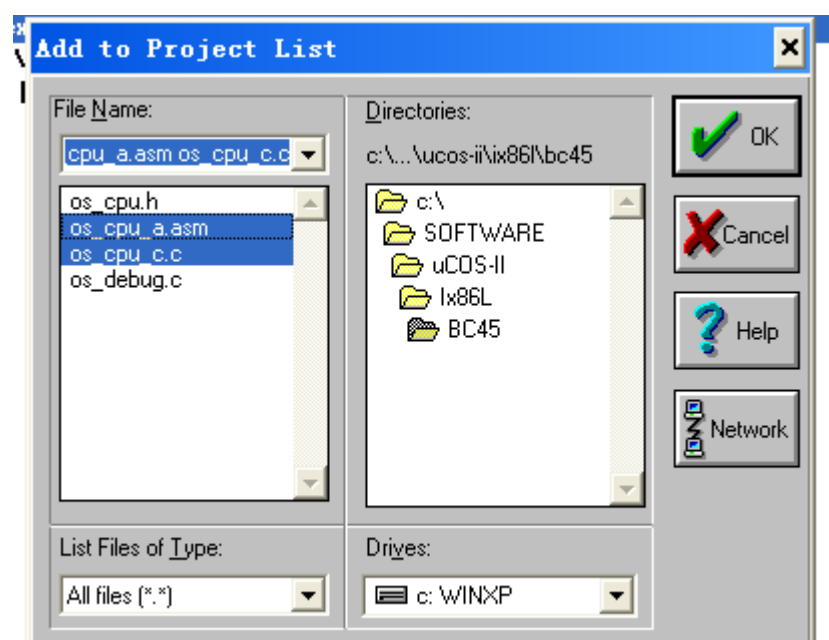


添加界面。添加如图，找到所在的目录，即：

“C: \software\uCOS-II\EX1-x86L\BC45\SOURCE”。里面有个

“test.c” 文件，选中点击 “OK”。

依次添加其他文件。

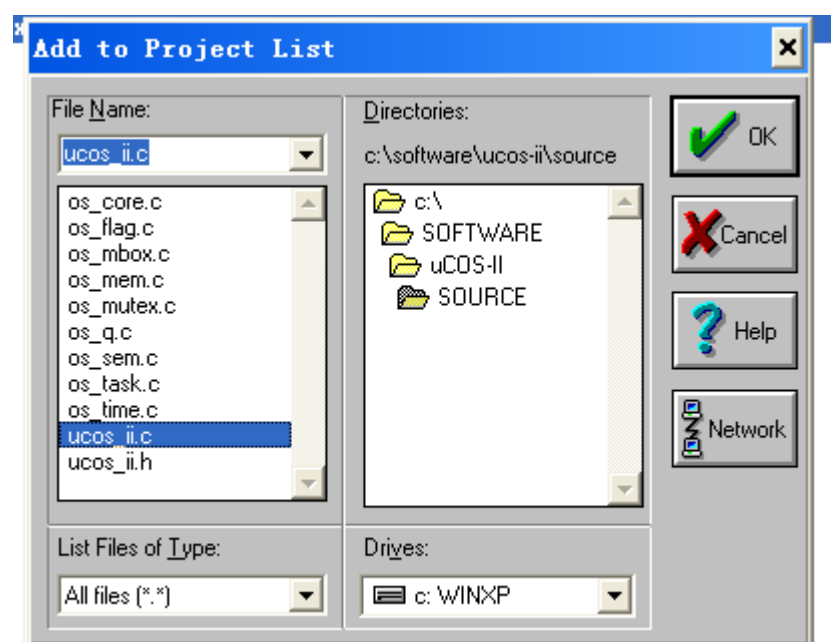


在 “C: \software\uCOS-II\Ix86L\BC45” 中，在 List Files of Type

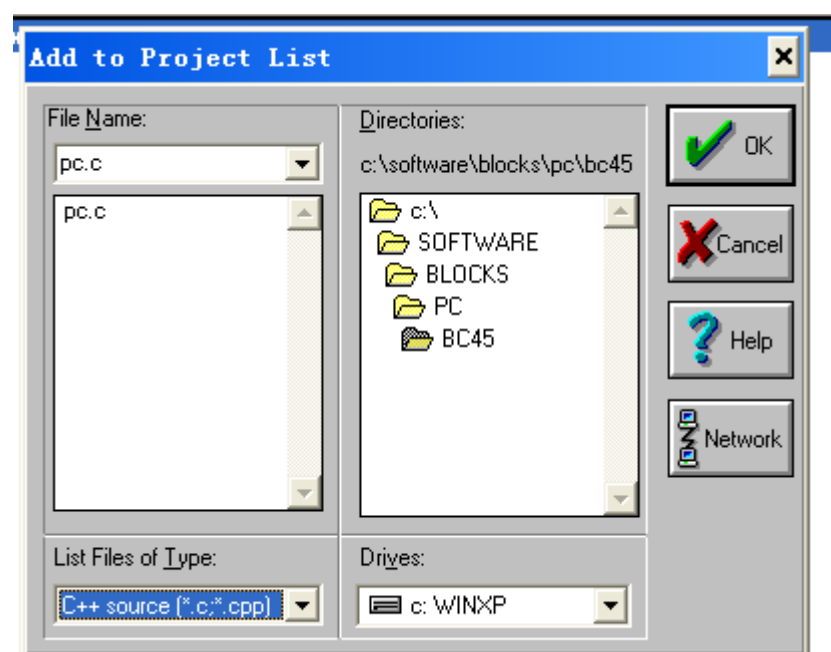
中选择 “All file (*.*)”。然后在选 “os_cpu-a.asm” 和 “os_cpu-c.c”

uC/OS 在 PC 机上的移植

文件，点击“OK”。

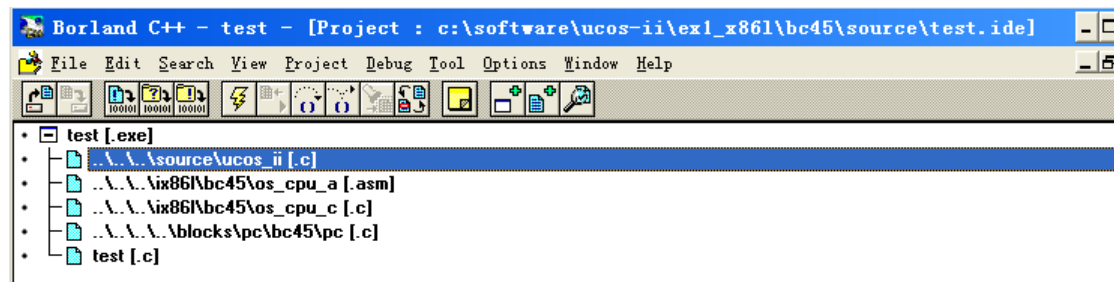


在“C:\software\uCOS-II\SOURCE”中，选择“ucos-ii.c”文件，点击“OK”。



最后一个文件，在“C:\software\BLOCKS\PC\BC45”中，选择“pc.c”文，点击“OK”。

uC/OS 在 PC 机上的移植

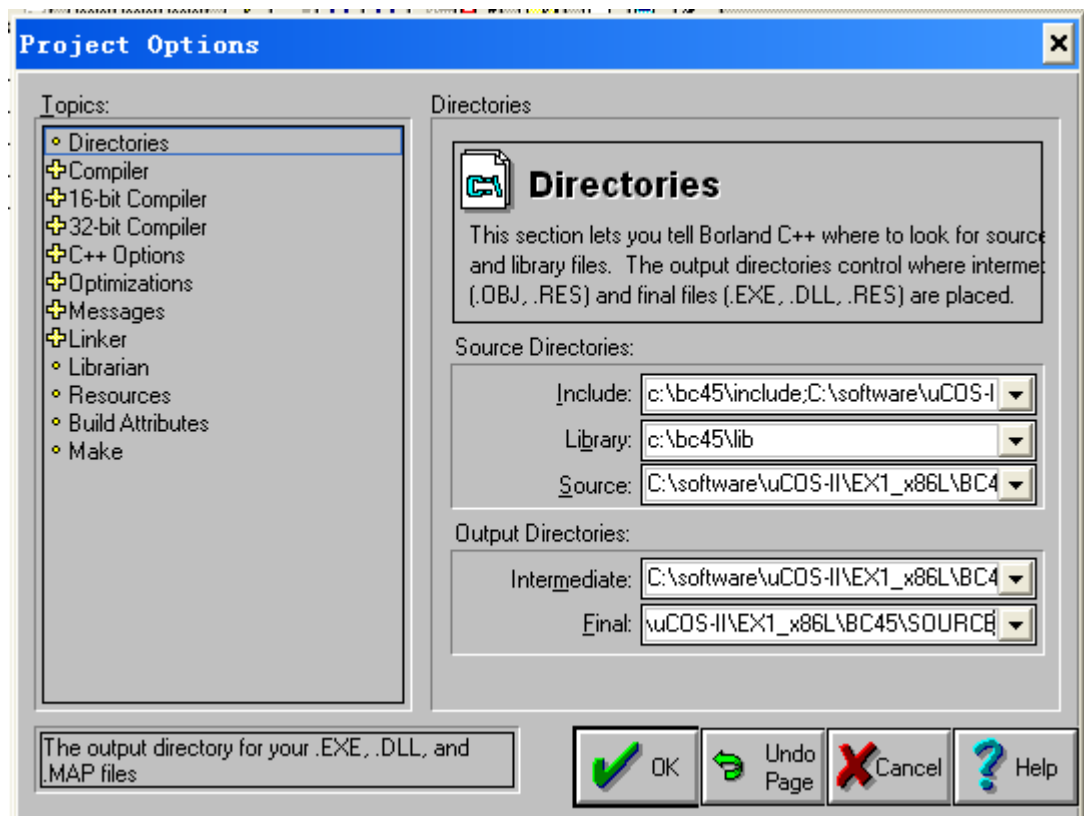


嘿嘿，所有的文件都添加完成了。

下面，我们开始设置环境属性。



在菜单栏-->Options-->Project



这就是属性设置界面了！在 Topics 中点 Directories, 只设置这个选

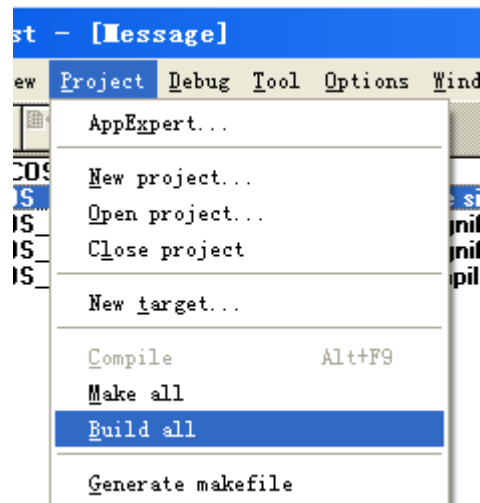
uC/OS 在 PC 机上的移植

项中的路径等内容就 OK! 分别在

Include, Library, Source, Intermediate, Final 中添加

“; C: \software\uCOS-II\EX1_x86L\BC45\SOURCE” 这个路径即可。

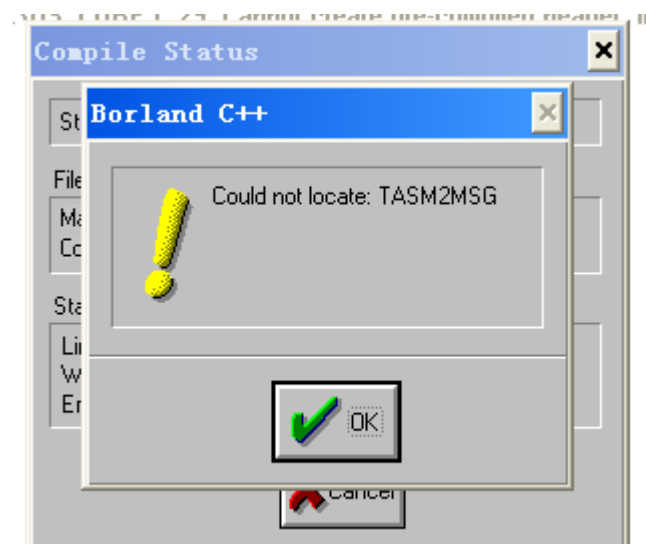
点击 O K 完成。



我们开始进行编译啦! 希望就在前方。

在菜单栏-->Project-->Build all

进行编译。



呃! 出了这么错误。咋回事呢?

我百度啊。。。嘻嘻。。。终于还试找到了，原来是在工程要加入这个文

uC/OS 在 PC 机上的移植

件。

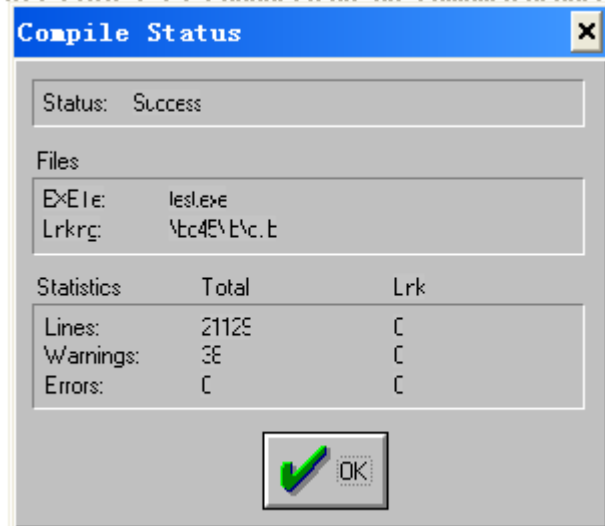


就这个！这个文件在“C:\BC45\BIN”文件夹中(也就是开发环境的目录下)，把它拷贝到“C:\software\uCOS-II\EX1_x86L\BC45\SOURCE”目录下(也就是我们工程目录下)。



就这儿。

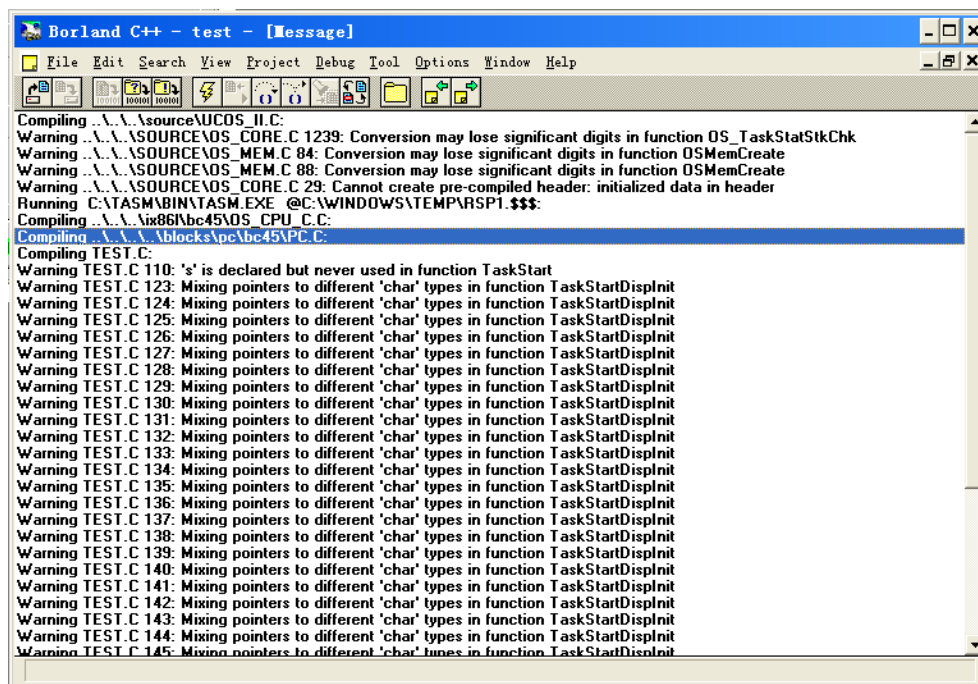
ERROR CORE C 29: Cannot create pre-compiled header; in



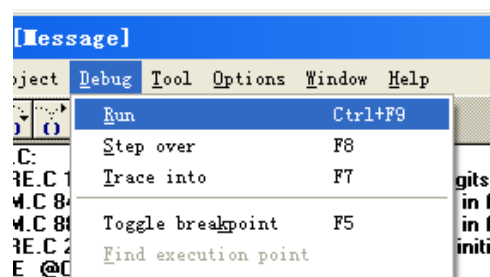
再重新编译哈！嘿嘿，那个错解决了。

编译没有错哎！不过警告好多，38 个。

uC/OS 在 PC 机上的移植

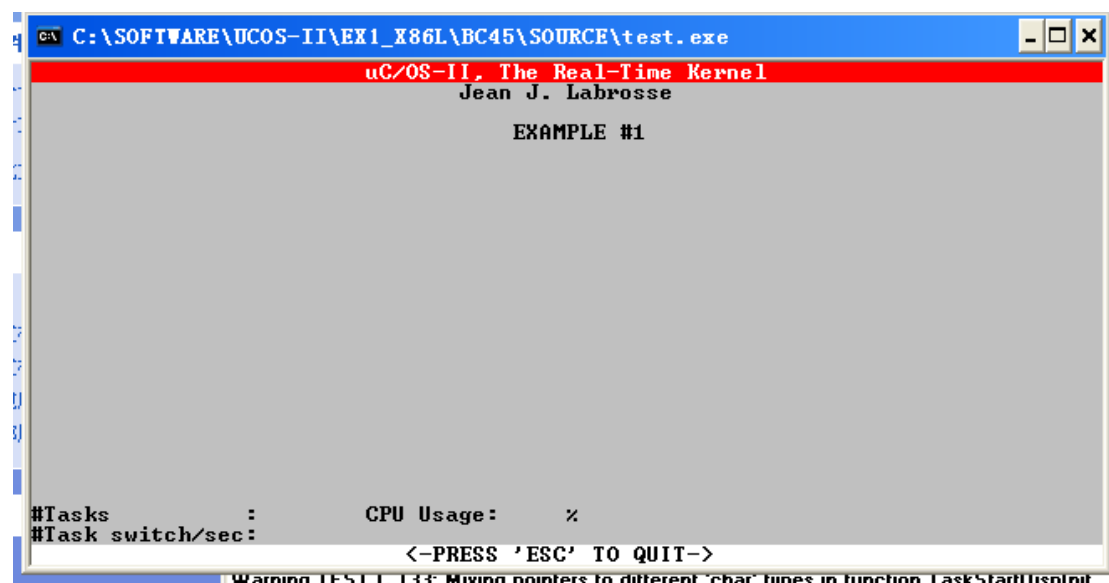


警告 ing!!!



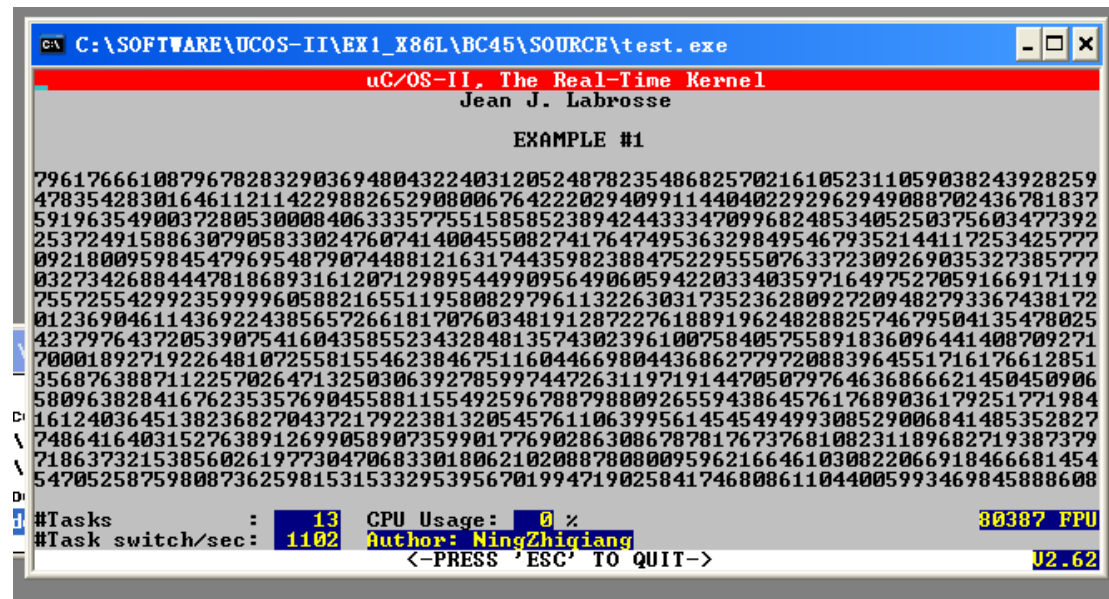
我们准备运行吧！

菜单栏-->Debug-->Run



uC/OS 在 PC 机上的移植

开始运行了, 好兴奋! 0(n-n)0 哈哈~

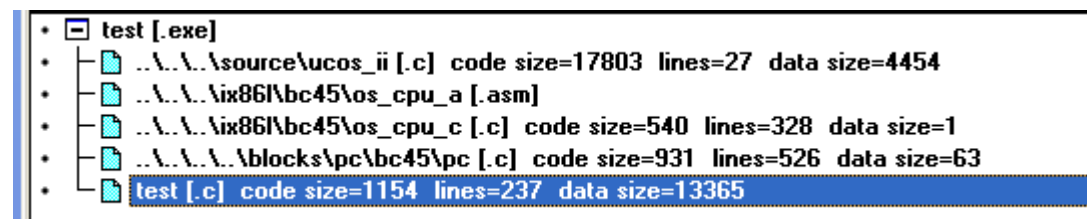


```
C:\SOFTWARE\UCOS-II\EX1_X86\BC45\SOURCE\test.exe
uC/OS-II, The Real-Time Kernel
Jean J. Labrosse

EXAMPLE #1
79617666108796782832903694804322403120524878235486825702161052311059038243928259
47835428301646112114229882652908006764222029409911440402292962949088702436781837
59196354900372805300084063335775515858523894244333470996824853405250375603477392
25372491588630790583302476074140045508274176474953632984954679352144117253425777
09218009598454796954879074488121631744359823884752295550763372309269035327385777
03273426884447818689316120712989544990956490605942203340359716497527059166917119
75572554299235999960588216551195808297961132263031735236280927209482793367438172
01236904611436922438565726618170760348191287227618891962482882574679504135478025
42379764372053907541604358552343284813574302396100758405755891836096441408709271
70001892719226481072558155462384675116044669804436862779720883964551716176612851
35687638871122570264713250306392785997447263119719144705079764636866621450450906
58096382841676235357690455881155492596788798809265594386457617689036179251771984
16124036451382368270437217922381320545761106399561454549499308529006841485352827
74864164031527638912699058907359901776902863086787817673768108231189682719387379
71863732153856026197730470683301806210208878080095962166461030822066918466681454
54705258759808736259815315332953956701994719025841746808611044005993469845888608

#Tasks      : 13      CPU Usage: 0 %      80387 FPU
#Task switch/sec: 1102  Author: NingZhiqiang
<-PRESS 'ESC' TO QUIT->      V2.62
```

这个运行是什么啊? 看来得去分析哈代码啦!



```
test [exe]
├── ..\..\source\ucos_ii [c] code size=17803 lines=27 data size=4454
├── ..\..\ix86\bc45\os_cpu_a [asm]
├── ..\..\ix86\bc45\os_cpu_c [c] code size=540 lines=328 data size=1
├── ..\..\blocks\pc\bc45\pc [c] code size=931 lines=526 data size=63
└── test [c] code size=1154 lines=237 data size=13365
```

我们双击打开 “test.c” 的代码!

uC/OS 在 PC 机上的移植

```
File Edit Search View Project Debug Tool Options Window Help
[Icons]
/*
*****
*
*                               uC/OS-II
*                               The Real-Time Kernel
*
*          (c) Copyright 1992-2002, Jean J. Labrosse, Weston, FL
*                               All Rights Reserved
*
*                               EXAMPLE #1
*****
*/

#include "includes.h"

/*
*****
*
*                               CONSTANTS
*****
*/

#define TASK_STK_SIZE           512      /* Size of each task's stacks (# of WORDs) */
#define N_TASKS                 10      /* Number of identical tasks */

/*
*****
*
*                               VARIABLES
*****
*/

OS_STK      TaskStk[N_TASKS][TASK_STK_SIZE]; /* Tasks stacks */
OS_STK      TaskStartStk[TASK_STK_SIZE];
char        TaskData[N_TASKS]; /* Parameters to pass to each task */
OS_EVENT    *RandomSem;

/*
```

呵呵，人家写的代码号规范哦，惭愧！

```
*/
void TaskStart (void *pdata)
{
    #if OS_CRITICAL_METHOD == 3 /* Allocate storage for CPU status register */
        OS_CPU_SR cpu_sr;
    #endif
    char s[100];
    INT16S key;

    pdata = pdata; /* Prevent compiler warning */
    TaskStartDispInit(); /* Initialize the display */

    OS_ENTER_CRITICAL();
    PC_VectSet(0x08, OSTickISR); /* Install uC/OS-II's clock tick ISR */
    PC_SetTickRate(OS_TICKS_PER_SEC); /* Reprogram tick rate */
    OS_EXIT_CRITICAL();

    OSStatInit(); /* Initialize uC/OS-II's statistics */

    TaskStartCreateTasks(); /* Create all the application tasks */

    for (;;) {
        TaskStartDisp(); /* Update the display */

        if (PC_GetKey(&key) == TRUE) {
            if (key == 0x1B) {
                PC_DOSReturn(); /* Return to DOS */
            }
        }

        OSCtxSwCtr = 0; /* Clear context switch counter */
        OSTimeDlyHMSM(0, 0, 1, 0); /* Wait one second */
    }
}
```

这个是第一任务哦！经过分析，它主要是初始化显示及其创建其他任

uC/OS 在 PC 机上的移植

务的功能哈，其次就是判断键盘的按键，当按下了“Esc”键，则结束退出。

```
static void TaskStartCreateTasks (void)
{
    INT8U i;

    for (i = 0; i < N_TASKS; i++) {
        TaskData[i] = '0' + i;
        OSTaskCreate(Task, (void *) &TaskData[i], &TaskStk[i][TASK_STK_SIZE - 1], i + 1);
    }

    /*
    *****
    *                                     TASKS
    *****
    */

    void Task (void *pdata)
    {
        INT8U x;
        INT8U y;
        INT8U err;

        for (;;) {
            OSSemPend(RandomSem, 0, &err);
            x = random(80);
            y = random(16);
            OSSemPost(RandomSem);
            PC_DispChar(x, y + 5, *(char *)pdata, DISP_FGND_BLACK + DISP_BGND_LIGHT_GRAY);
            OSTimeDly(1);
        }
    }
}
```

这个是其他的任务了！这些任务是产生随机数，并且显示在屏幕上，共 10 个任务啦。

这个程序共有 13 个任务，其中 11 任务是用户任务，其他 2 个任务为系统任务，即空闲任务和统计任务。

不知道你发现了没有？在第二幅结果图中，运行结果里有作者名字哈，(*^__^*) 嘻嘻……

这个是我自己加上去的。。。。。

在此，全部的 uC/OS 在 PC 机上的移植已经将完了！预祝大家能够参考此文档顺利的移植。