

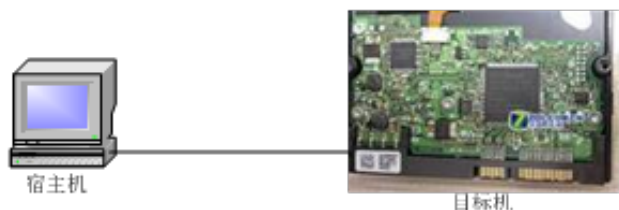
VxWorks5.5 环境搭建

一. VxWorks概述

VxWorks操作系统是美国WindRiver公司于1983年设计开发的一种嵌入式实时操作系统(RTOS)，它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通讯、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机、B-2隐形轰炸机和爱国者导弹上，甚至连1997年4月在火星表面登陆的火星探测器上也使用到了VxWorks。VxWorks原先对中国区禁止销售，自解禁以来，在我们的军事、通信、工业控制等领域得到了非常广泛的应用。

VxWorks®

如下图，嵌入式系统的调试方法一般为通过PC（宿主机）上的集成开发环境交叉编译针对特定电路板（目标机）的程序，然后将程序通过目标板的JTAG、串口或网口等途径下载到目标板上运行。因此，为了构造一个嵌入式系统的研究环境，拥有一块包含CPU、存储器及I/O电路的目标电路板往往是必要的。虽然许多集成开发环境附带模拟软件，但仅限于指令集的模拟，均无法模拟物理的目标机硬件平台，因而在其上只能进行应用程序的象征性模拟开发。但是，并非所有人都能拥有一块物理的电路板。



在这种情况下，我们如何构造一个模拟的开发环境，其学习效果就如同拥有完全真实的电路板一样呢？本文试图解答此问题，主体内容包括五个方面：

- 利用VMware等软件模拟真实的目标机；
- 安装Tornado开发环境
- 构建VMware虚拟PC上VxWorks BSP，建立Bootrom和OS映像；
- 修改Tornado相关设置，连接宿主机与目标机，建立调试通道；
- 写一个简单的应用程序并下载到目标系统运行。

本文工作的最终目标为：

- (1) VxWorks在VMware启动成功并顺利运行，如下图

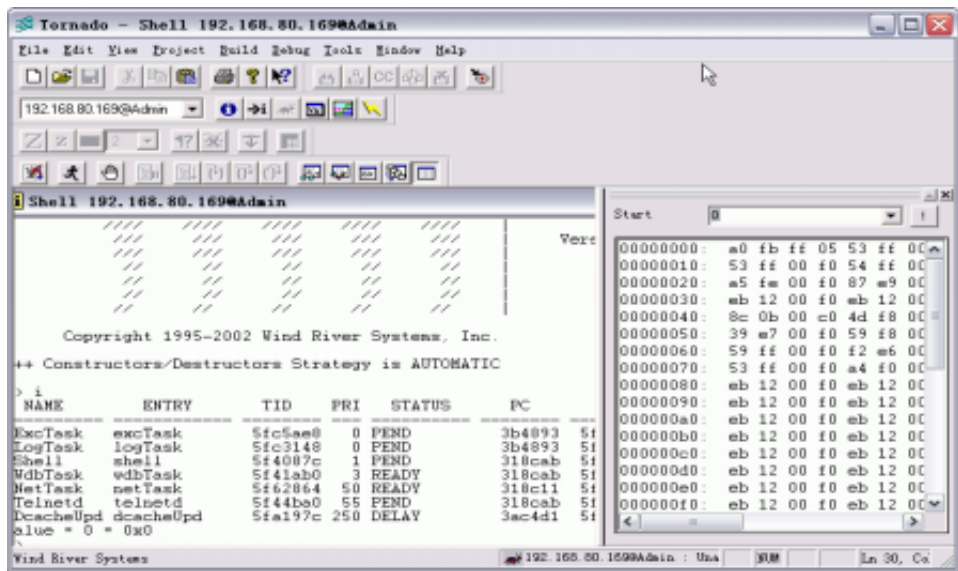
```
UxWorks

Copyright 1984-2002 Wind River Systems, Inc.

CPU: PC PENTIUM
Runtime Name: UxWorks
Runtime Version: 5.5
BSP version: 1.2/3
Created: Dec 2 2015, 13:21:12
WDB Comm Type: WDB_COMM_END
WDB: Ready.

->
Page Fault
Page Dir Base : 0x00ff0000
Esp0 0x00f6b844 : 0x00000001, 0x003a7fe8, 0x00fff24c, 0x00000100
Esp0 0x00f6b854 : 0x00ffff25c, 0x00ffff090, 0x00f92cc4, 0x00f92cec
Program Counter : 0x003b717c
Code Selector : 0x00000008
Eflags Register : 0x00010206
Error Code : 0x00000000
Page Fault Addr : 0x22602d84
Task: 0xf6bb94 "tPortmapd"
```

(2) 可在Tornado上针对目标板编译程序并进行调试，如下图



二. VmWare安装

PC也具有目标机的所有特点，实际上，我们可以把PC作为嵌入式系统的目标机，从而构造如图所示的开发模型：



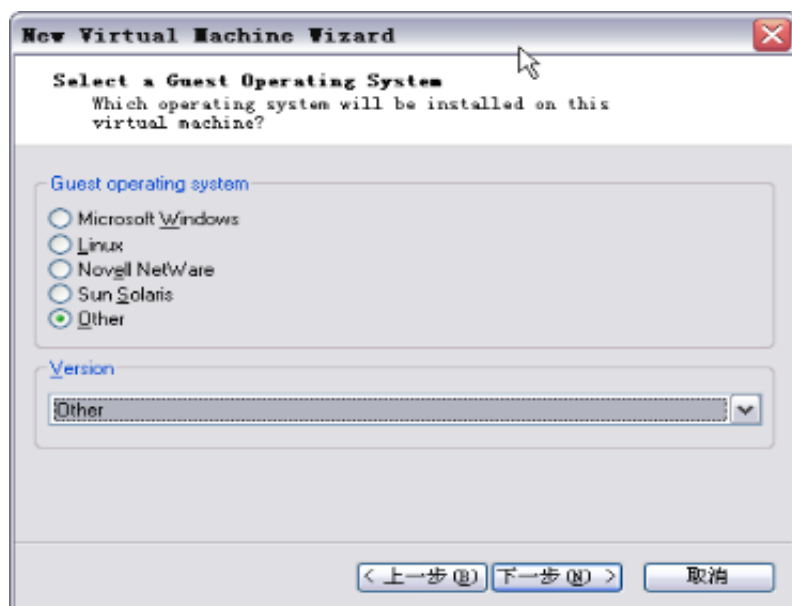
很遗憾，这种方法实际上非常麻烦，同时开动两台PC进行调试将使我们饱受折磨。因此，我们可以借助VMware来在本机上虚拟出另一PC。

VMware的确是天才的作品！在同一PC上，利用VMware几乎可以安装所有的操作系统，而且操作系统之间的切换不需要重新启动电脑。VM的意义是Virtual Machine，即虚拟出一个逻辑的电脑，它可以提供基于Intel CPU的虚拟PC系统环境，包括CPU、内存、BIOS、硬盘和其他外围硬件设备。

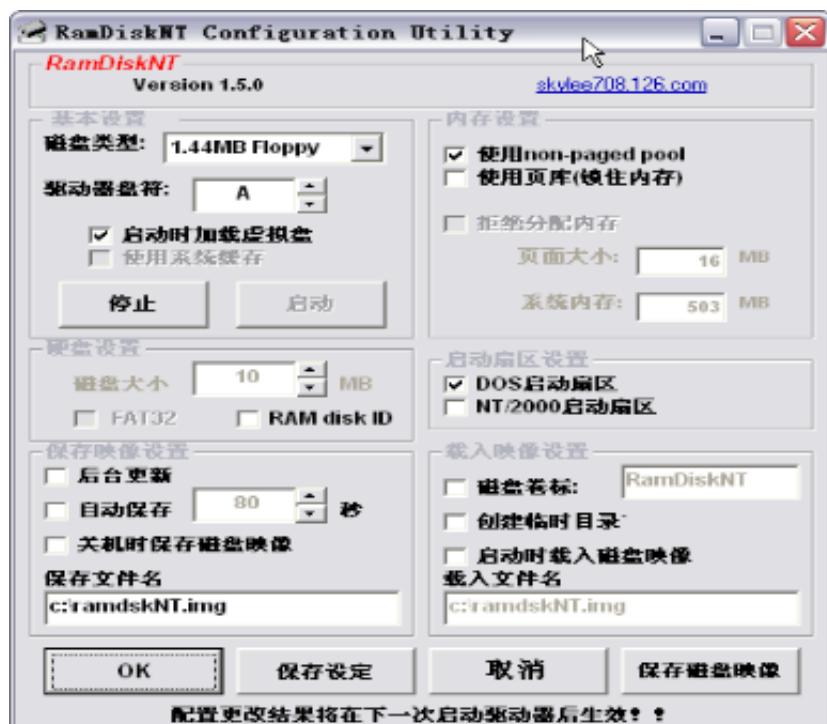
下面我们讲解用VMware来建立一台虚拟PC的步骤：

- (1) 下载并安装VMware；
- (2) 使用VMware向导建立一个针对VxWorks的虚拟机；

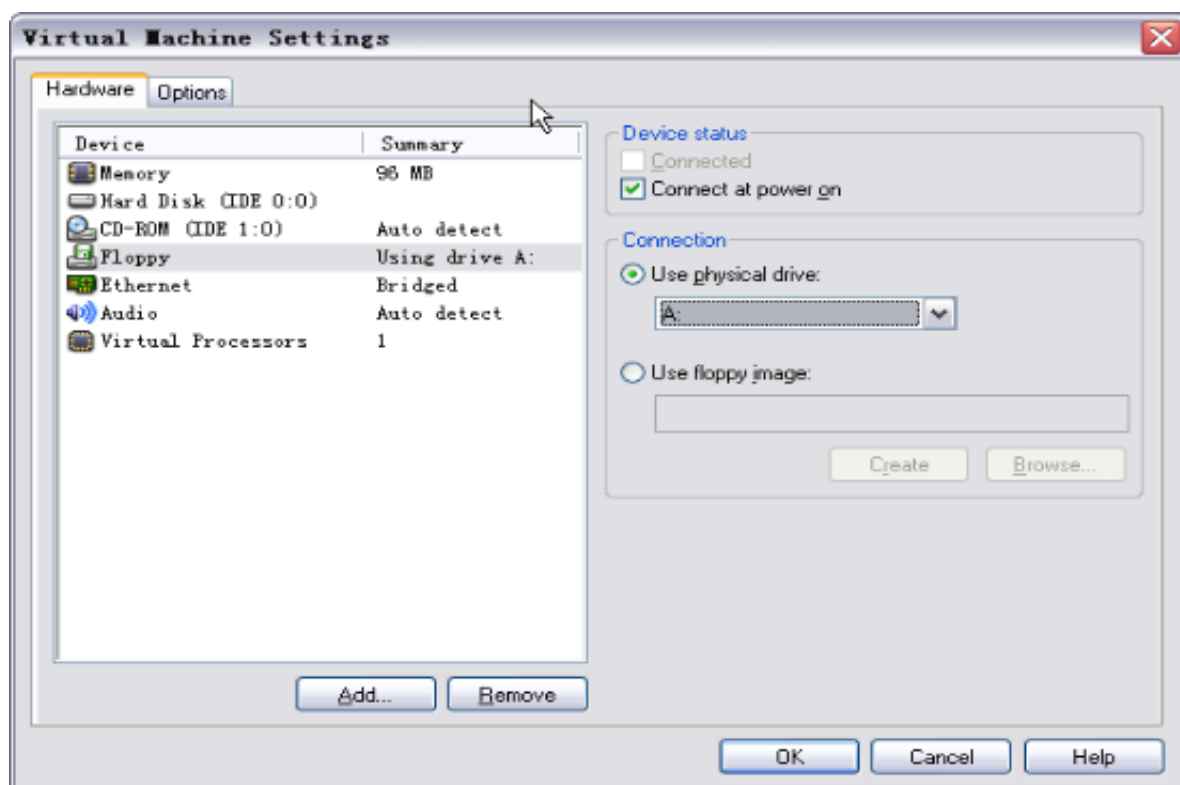
此步骤中注意在操作系统中选择“other”，如图：



由于目标机最终通过软盘启动，因此要求你的电脑具有软驱。很遗憾，当年日常使用的软盘如今成了古董，很少再有电脑配备软驱。因此，我们再来制造一个假冒伪劣产品，虚拟一个软驱。又一个天才的工具软件RamDiskNT为我们提供了这一便利，图演示了用RamDiskNT虚拟一个1.44M软盘的方法。



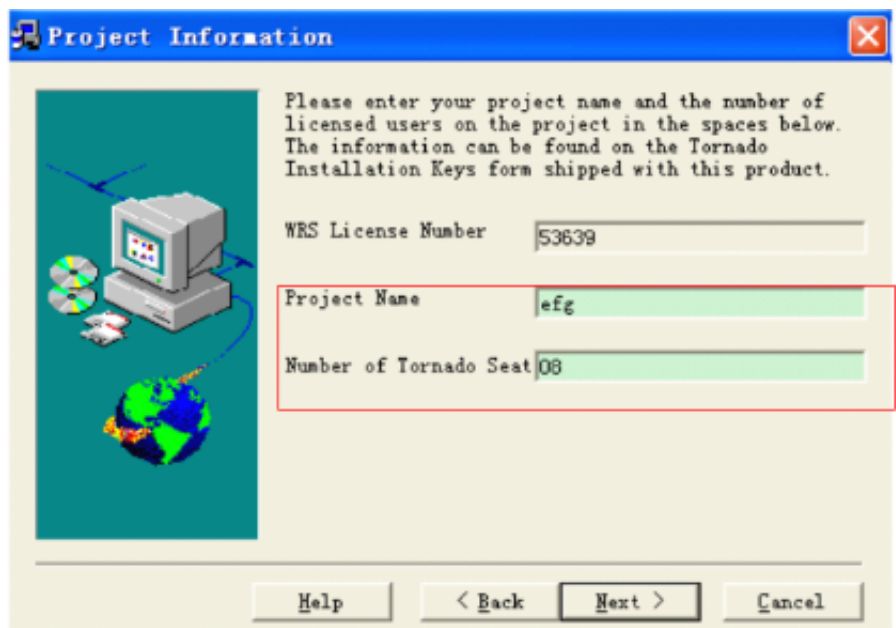
仅仅虚拟一个软驱是不够的，把这个软驱添加到我们建立的虚拟机中才算修成正果，下图演示了添加软驱后的虚拟机硬件设置。



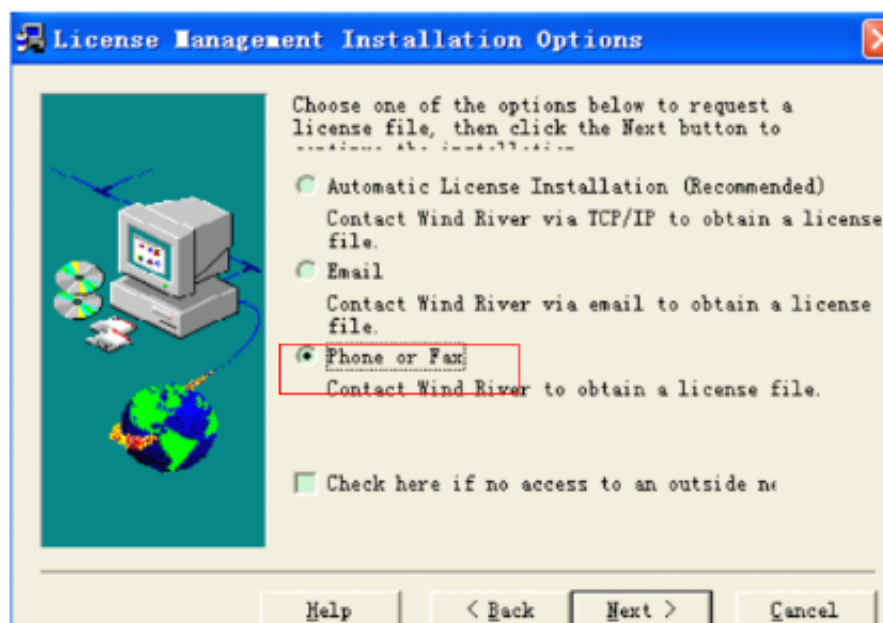
三. 安装Tornado

安装Tornado 2.2需使用公司提供的WindRiver Tornado 2.2破解安装包。该安装包内包含CD1、CD2两个文件夹和一个vxworks_end.exe可执行文件。

CD1文件夹中为Tornado2.2安装文件； CD2中为BSP安装包； vxworks_end.exe文件后面会用到。 Tornado 2.2的安装如同Windows其它应用程序， 运行Setup开始安装。 这里先安装CD1中的Tornado 2.2程序， 再安装CD2中的BSP驱动。 在使用CD1安装Tornado 2.2时候， 有两个步骤需要注意， 如下图所示



在安装Tornado 2.2时候， 如果安装到了如上图所示步骤， 则其中的Project Name和Number of TornadoSeat是可以随意填写的。不过要注意Number of Tornado Seat栏只能填写数字。



当安装到如上图步骤时， 请选择Phone or Fax单选按钮。其它均使用默认设置即可。在安装CD2中的BSP驱动时候， 如同CD1中的安装步骤， 不过更为简单。只需使用CD2文件夹中的序列号， 一路Next即可完成安装。 按照上面步骤完成安装后需要对软件进行破解才能正常使用。需要的破解文件包含在CD1中的CRACK文件夹， 名为license.dat。破解步骤如下： （1）复制license.dat文件到C:\Tornado2.2\flexlm文件夹下， 需要首先在Tornado 2.2文件夹中创建flexlm文件夹 （2）新建变量名为LMLICENSEFILE， 变量值为C:\Tornado2.2\flexlm\license.dat的环境变量。

如果不出意外，安装和破解Tornado2.2的所有工作就完成了。注意Tornado2.2和Win7不兼容，即使能通过网上一些方法成功安装，但是也可能在后期使用的时候出现各种意想不到的问题。

四. 构造BSP

• 下载网卡驱动

VMware为运行于其上的操作系统提供虚拟网卡支持，该网卡类型即为AMD的PC-NET。实际上，在Tornado开发包中已经包含了该类型网卡的驱动程序，但经过测试，对于VMware无法正常工作，所以需要从AMD的网站下载最新的驱动程序。下载得到的是一个可执行的安装程序，运行该程序将得到一个压缩包和一个帮助文件，按照该帮助的要求，将压缩包直接释放到Tornado目录下。如果提示是否允许覆盖文件，则选择允许。此后按如下步骤完成驱动程序的编译和替换：

1) 打开一个控制台窗口，运行批处理程序：

```
C:\tornado2.2\host\x86-win32\bin\torVars.bat;
该批处理文件用于设置环境变量，其内容为：
rem Command line build environments
set WIND_HOST_TYPE=x86-win32
set WIND_BASE=C:\Tornado2.2
set PATH=%WIND_BASE%\host%\WIND_HOST_TYPE%\bin;%PATH%`
如果没有设置成功，可以手工设置这些环境变量。
```

2) 定位到C:\tornado2.2\target\src\drv\end目录，运行：

```
make CPU=PENTIUM tool=gnu ln97xend.o
```

3) 定位到C:\tornado2.2\target\lib\pentium\PENTIUM\common目录，并将上一步生成的文件ln97xend.o复制到此目录下。备份此目录下的文件libdrv.a；

4) 运行命令arpentium -d libdrv.a ln97xEnd.o，删除libdrv.a中原有的ln97xEnd模块；

5) 运行命令arpentium -ra iOlicomEnd.o libdrv.a ln97xEnd.o将我们刚刚创建的新ln97xEnd模块添加进去。

• 修改配置文件

我们要修改编译VxWorks的配置头文件Config.h中定义的一些参数，使编译出来的系统引导程序和VxWorks的映像符合我们的要求；同时还要修改sysLn97xEnd.c这个文件，以使系统的网络功能正常运行。

1) 定位目录到C:\tornado2.2\target\config\pcPentium并打开该目录下Config.h文件；

2) 查找到定义DEFAULTBOOTLINE宏的地方，修改预处理条件CPU == PENTIUM分支下的定义如下：


```
#define DEFAULT_BOOT_LINE \
"lnPci(0,0)your_host_name:d:\\vxWorks h=193.168.80.169 e=193.168.80.254 u=targ
et pw=vxworks tn=target"
```

其中：

lnPci(0,0)指定了使用第0个网卡和第0个处理器，lnPci这个标识会因为使用的驱动程序不同而有所不同，但这里用lnPci就可以了；

host指定你的主机的名字，使用host就可以；

d:\\vxWorks指定了VxWorks映象下载的完整路径；

h=193.168.80.169是宿主机的IP地址，可以根据实际情况修改；

e=193.168.80.254是目标机的IP地址，可以根据实际情况修改；

u=target指定了FTP服务器的用户名，这个FTP就是用来下载VxWorks映象的，后面还会提到；

pw=target是用户名对应的口令；

tn=target指定目标机的名字，任意指定即可。

3) 指定网卡驱动程序

首先查找“Network driver options”这段文字，之后你可以看到在该注释后面定义了一系列的有关网卡驱动

的宏定义。注意保证INCLUDE_END和INCLUDE_LN_97X_END这两个宏处于定义状态（define），其他的宏都处于未定义状态（undef），即：

```
1  /* Network driver options */
2  #define INCLUDE_END          /* Enhanced Network Driver Support */
3  #undef INCLUDE_DEC21X40_END /* (END) DEC 21x4x PCI interface */
4  #undef INCLUDE_EL_3C90X_END /* (END) 3Com Fast EtherLink XL PCI */
5  #undef INCLUDE_ELII_3C509_END /* (END) 3Com EtherLink III interface */
6  #undef INCLUDE_ENE_END      /* (END) Eagle/Novell NE2000 interface */
7  #undef INCLUDE_FEI_END      /* (END) Intel 8255[7/8/9] PCI interface */
8  #undef INCLUDE_GEI8254X_END /* (END) Intel 82543/82544 PCI interface */
9  #define INCLUDE_LN_97X_END  /* (END) AMD 79C97x PCI interface */
10 #undef INCLUDE_ULTRA_END    /* (END) SMC Elite16 Ultra interface */
11
12 #undef INCLUDE_BSD          /* BSD / Netif Driver Support (Deprecated) */
13 #undef INCLUDE_EEX          /* (BSD) Intel EtherExpress interface */
14 #undef INCLUDE_EEX32        /* (BSD) Intel EtherExpress flash 32 */
15 #undef INCLUDE_ELC          /* (BSD) SMC Elite16 interface */
16 #undef INCLUDE_ESMC         /* (BSD) SMC 91c9x Ethernet interface */
```

4) 缺省情况下,VxWorks系统是不接受外部输入设备（如键盘）的输入，也不向外部输出设备（如显示器）输出数据。为了便于调试，我们必须改变它的这种缺省状态。我们查找定位宏

INCLUDE_PCCONSOLE，然后保证其处于定义状态（define）即可，即：

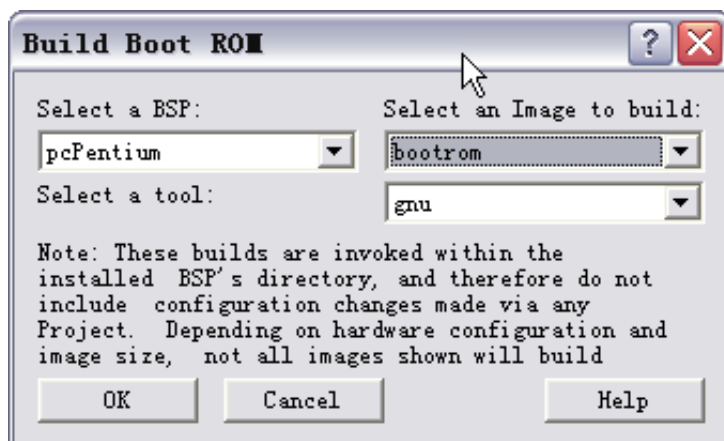
```
1  #define INCLUDE_PC_CONSOLE /* PC keyboard and VGA console */
2  #ifdef INCLUDE_PC_CONSOLE
3  #   define PC_CONSOLE      (0) /* console number */
4  #   define N_VIRTUAL_CONSOLES (2) /* shell / application */
5  #endif /* INCLUDE_PC_CONSOLE */
```

(5) 修改sysLn97xEnd.c 打开C:\tornado2.2\target\config\pcPentium目录下的sysLn97xEnd.c文件，先定位到“memory-mapped IO base”这段文字，然后将其前面的参数由pciRsrc[endUnit].bar[1]修改为NONE，保存即可，即：

```
1  sprintf (paramStr, paramTemplate,
2      endUnit,                /* END unit number */
3      NONE,                   /* memory-mapped IO base */
4      pciRsrc[endUnit].bar[0], /* IO address space base */
5      PCI2DRAM_BASE_ADRS,     /* host PCI mem. base */
6      pciRsrc[endUnit].irqvec, /* IRQ vector */
7      pciRsrc[endUnit].irq,    /* IRQ number */
8      LN97X_CSR3_VALUE,       /* csr3 register value */
9      LN97X_OFFSET_VALUE,     /* offset */
10     LN97X_RSVD_FLAGS,        /* flags (reserved) */
11     &ln97xStrDesc[typeIdx]   /* device description */
12 );
```

• 编译Bootrom

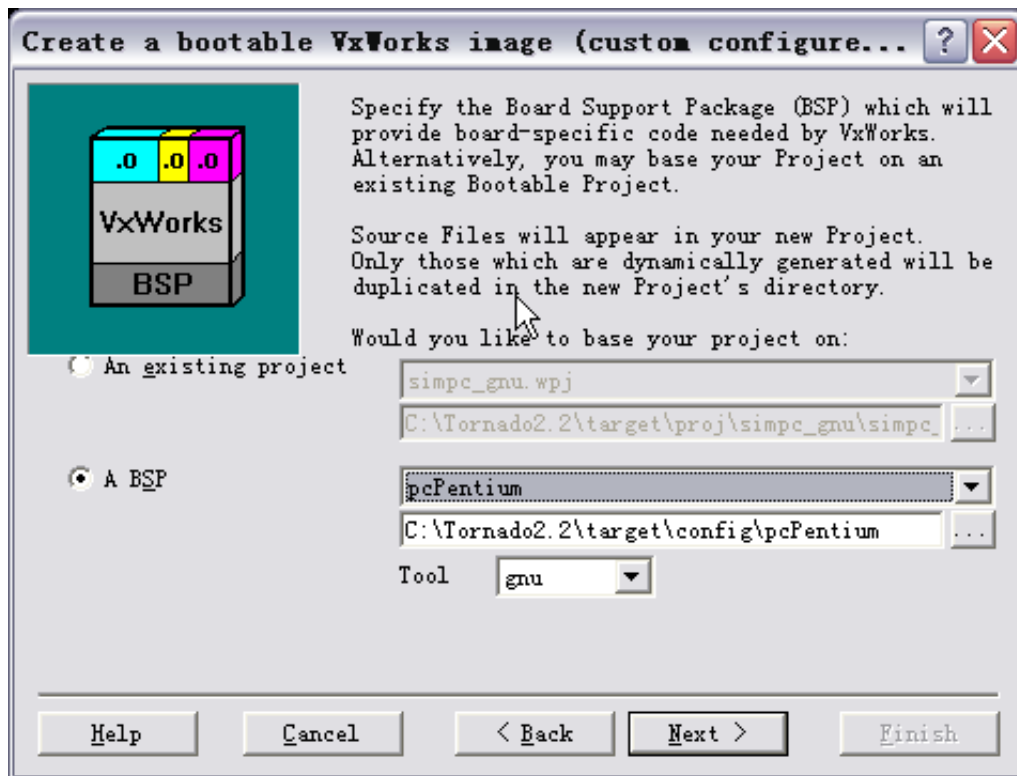
打开Tornado开发工具，在Build菜单下选择Build Boot ROM，选择BSP为pcPentium，image类型为bootrom，工具为gnu，如下图。点击OK，Bootrom就会编译。



• 编译VxWorks映像

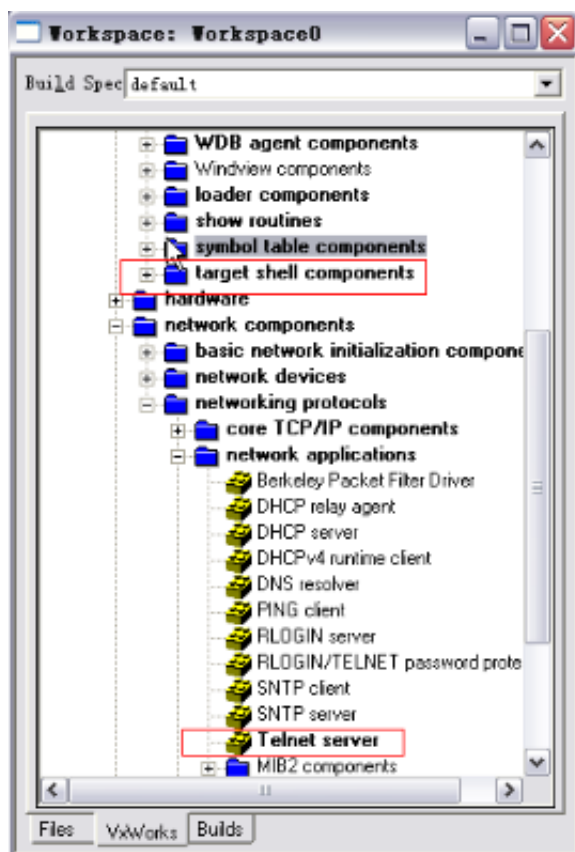
编译生成bootrom后，还要创建一个VxWorks映像（image），也就是VxWorks操作系统本身的代码，步骤为：

1) 创建一个“bootable VxWorks image”的工程，BSP 选择pcPentium；



2) 选择需要的VxWorks组件

如下图，在这个例子中我们需要包括两个重要的组件：Telnet server 和 Target shell。前者使我们可以通Telnet协议登录到VxWorks操作系统中；后者则可以让我们通过命令行控制VxWorks系统。另外，需要把所有C++相关的选项都包含进去。



完成选择后，即可开始编译程序。到此我们已经生成了VxWorks的系统引导程序和运行时的代码映像。这里还要提醒读者，在你每次修改完系统的配置信息（如：config.h）后，都要重新创建一个工程来编译VxWorks映像，以免出现代码不一致的问题。将生成的名为“vxworks”的文件(在你创建的工程的目录下)复制到D:\VxWorks目录下。这个路径是由上面我们所设置的DEFAULTBOOTLINE宏中的路径参数决定的，必须保持二者一致。

- **创建引导软盘**

现在开始制作VxWorks系统引导磁盘，用于引导装载VxWorks运行映像。定位目录到C:\tornado2.2\target\config\pcPentium，插入您已经格式化好的软盘，然后运行：

```
mkboot a: bootrom
```

该命令将在软盘上建立VxWorks系统引导分区，并将引导程序复制到软盘上。

五. 建立调试环境

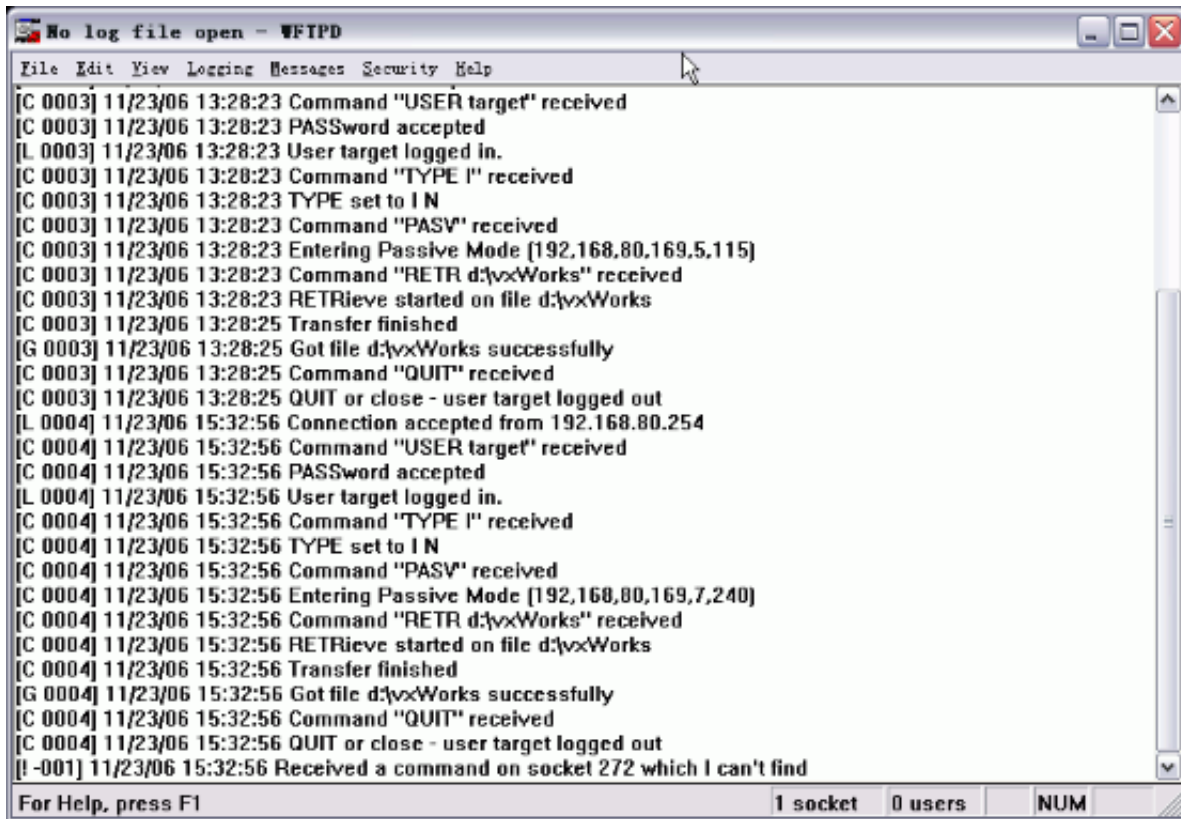
- **配置FTP服务器**

这里的FTP服务器用于在系统成功引导后，下载VxWorks的运行时映像。我们这里使用Tornado开发环境自带的FTP服务器。如下图，打开Tornado FTP Server，选择“Security”菜单下的“Users/Rights”子菜单，创建User Name为“target”，修改“Home Directory”为D:\VxWorks目录（此路径由上面的DEFAULTBOOTLINE参数决定），同时修改口令为“target”，最后点击“Done”按钮完成修改。



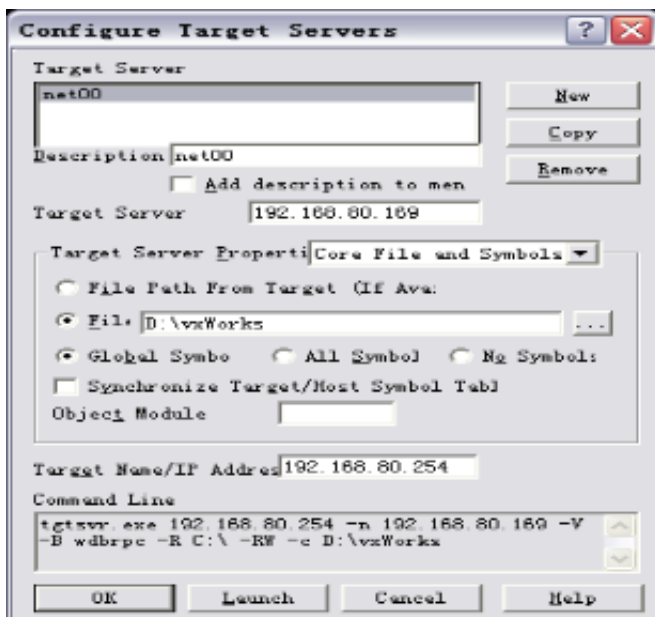
为了便于调试，我们还要打开FTP Server的日志功能。选择“Logging”菜单下的“Logging Options”子菜单，其中除了“Winsock Calls”外，让其他选项全都处于开启状态。保持FTP Server窗口处于打开状态（这样FTP服务器就处于运行状态）。

虚拟机从软盘启动，接下来会下载服务器D:\vxworks映像，服务器的FTP会有相关的log，如下图：



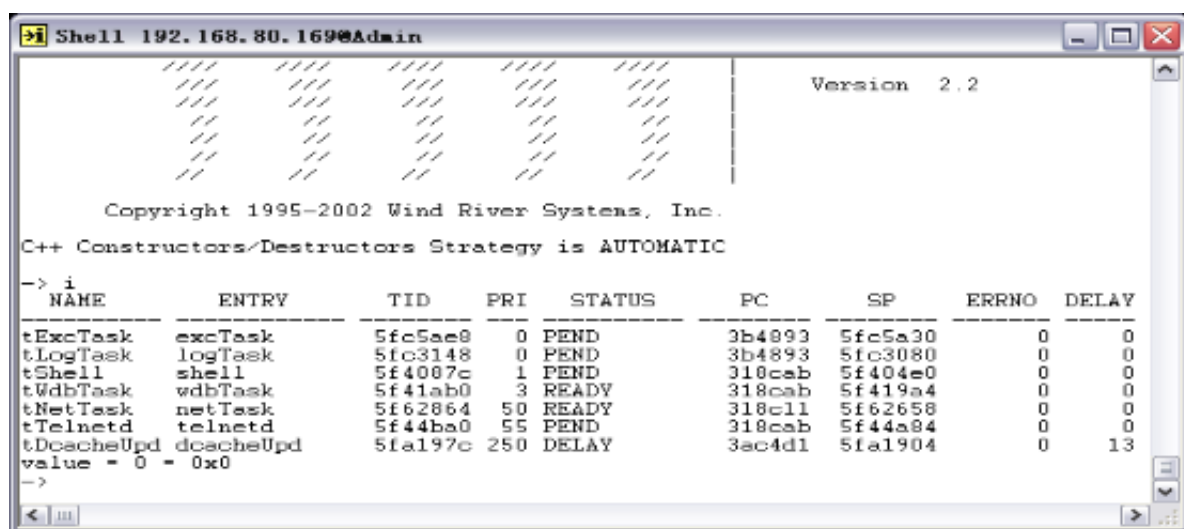
• 配置target server

如下图，打开您的Tornado开发环境，选择“Tools->Target Server->Configure”菜单，在“Description”中任意填写一个名字，这里是“net00”；在“Available Back”中选择“wdbRPC”，并在下面的IP地址框中填写目标机的IP地址，这里是“192.168.80.254”（由DEFAULTBOOTLINE参数决定）；将“Target Server Properties”下拉框更改至“Core File and Symbols”，并在“File Path”一项中选择您的映象的完整路径，这里是“D:\VxWorks”（由DEFAULTBOOTLINE参数决定）。



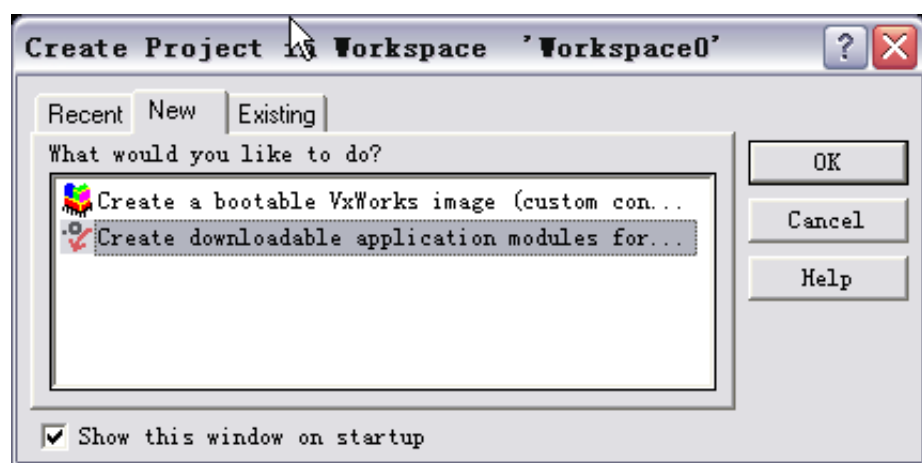
完成以上两项配置，如果目标机已经启动，点击“Launch”按钮，就可以启动Target Server了。再回到Tornado开发环境，在工具条上的Target Server下拉框列表中选择“192.168.80.169@host”。这时您会发

现工具条中一些原先处于“禁用”状态的工具按钮，现在都已经处于“激活”状态了。试一试，我们打开 TShell，并运行“i”命令，效果如下图：



六. 编写简单程序

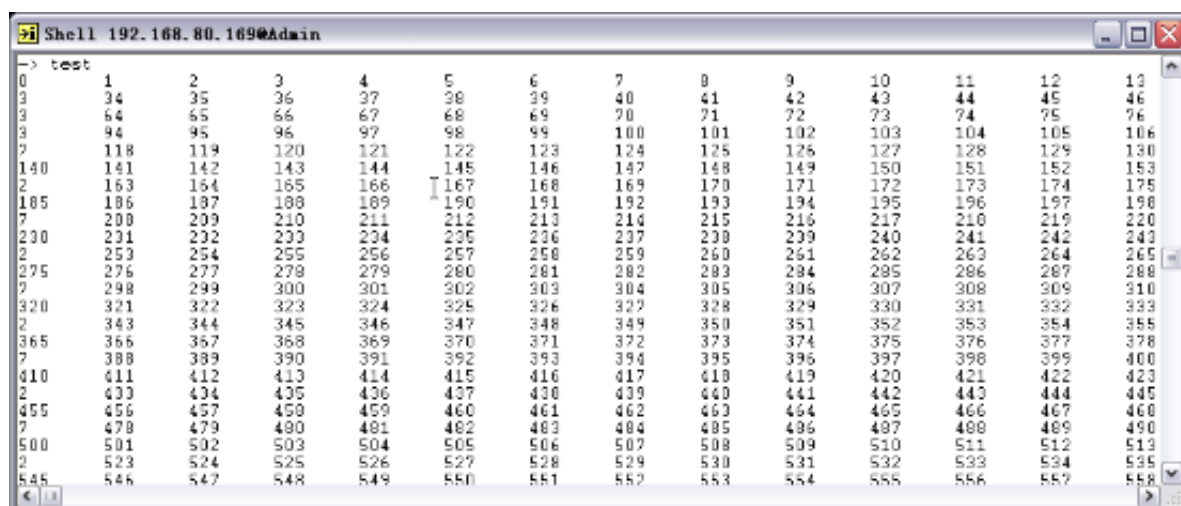
现在，编写应用程序的所有条件已经完全具备了。我们来编写一个简单的应用。如下图，建立一个名为“test”的可下载的应用程序模块：



这个时候工程中还一个文件都没有，我们编写一个简单的程序：编译生成了test.out，下载到目标板，在 TShell中运行test函数，效果如图17。

```
1  #include <stdio.h>
2  void test(void)
3  {
4      int i = 0;
5      while(1)
6      {
7          printf("%d\t",i);
8          i++;
9          if(i==10000)
10         {
11             break;
12         }
13     }
14 }
```

下面我们以Run Task模式运行上述的test()函数，并设置在入口点即断点：



此后，我们可以像使用任何其它的集成开发环境一样在我们编写的应用程序中进行所有如下的调试行为：

- 设置断点
- 单步运行
- 察看变量
- 察看内存
- 察看寄存器

接下来，开动吧！