VxWorks6.6 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。

如上图，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，主体内容包括四个方面：

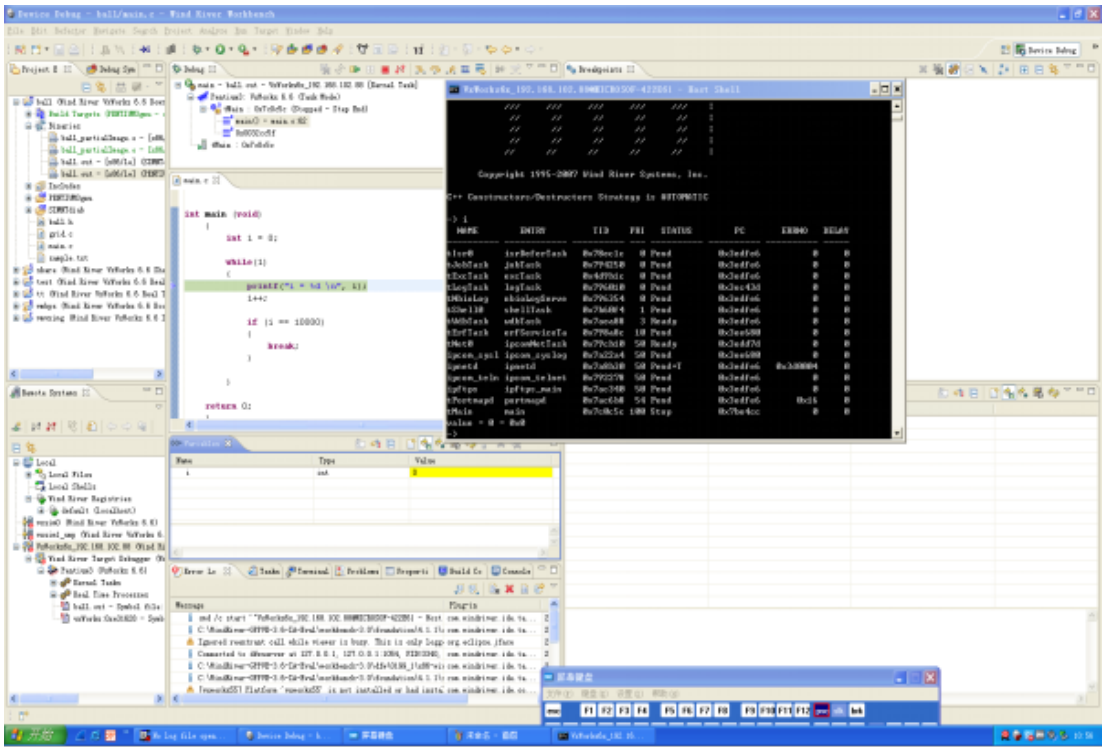
- 利用VMware等软件模拟真实的目标机；
- 构建VMware虚拟PC上VxWorks BSP，建立Bootrom和OS映像；
- 安装Workbench 3.0 开发环境
- 修改Workbench 3.0相关设置，连接宿主机与目标机，建立调试通道；

本文工作的最终目标为：

1）VxWorks在VMware启动成功并顺利运行

```
dding 5845 symbols for standalone.


                VxWorks

Copyright 1984-2007  Wind River Systems, Inc.

            CPU: PC PENTIUM3
    Runtime Name: VxWorks
 Runtime Version: 6.6
     BSP version: 2.0/9
         Created: Dec 14 2015, 16:14:19
ED&R Policy Mode: Deployed
   WDB Comm Type: WDB_COMM_END
            WDB: Ready.


-> _
```

2）可在Workbench 上针对目标板编译程序并进行调试



二. 安装VMWARE

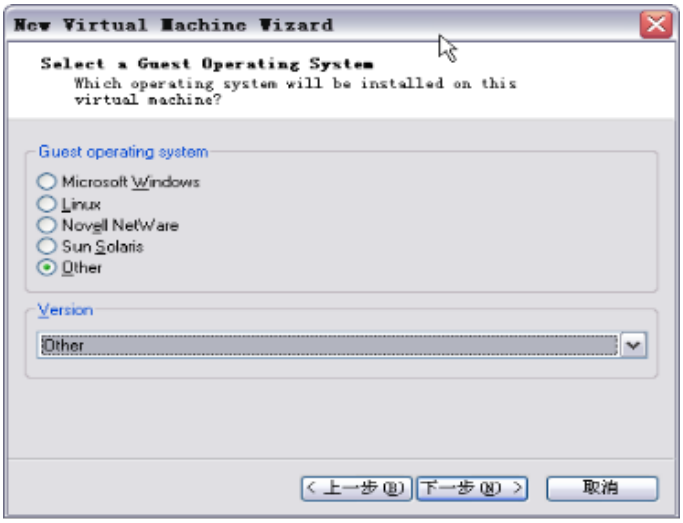
PC也具有目标机的所有特点，实际上，我们可以把PC作为嵌入式系统的目标机，从而构造如下图所示的开发模型：

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。

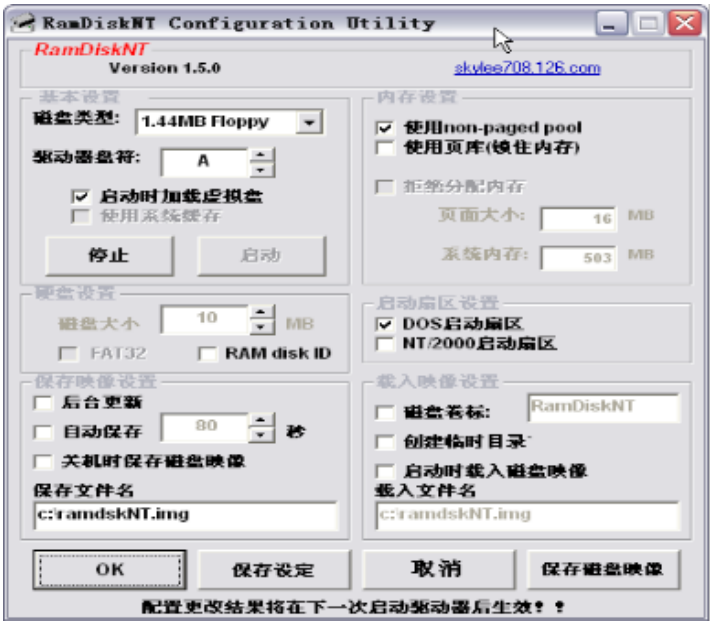
下面我们讲解用VMware来建立一台虚拟PC的步骤：

1）下载并安装VMware；

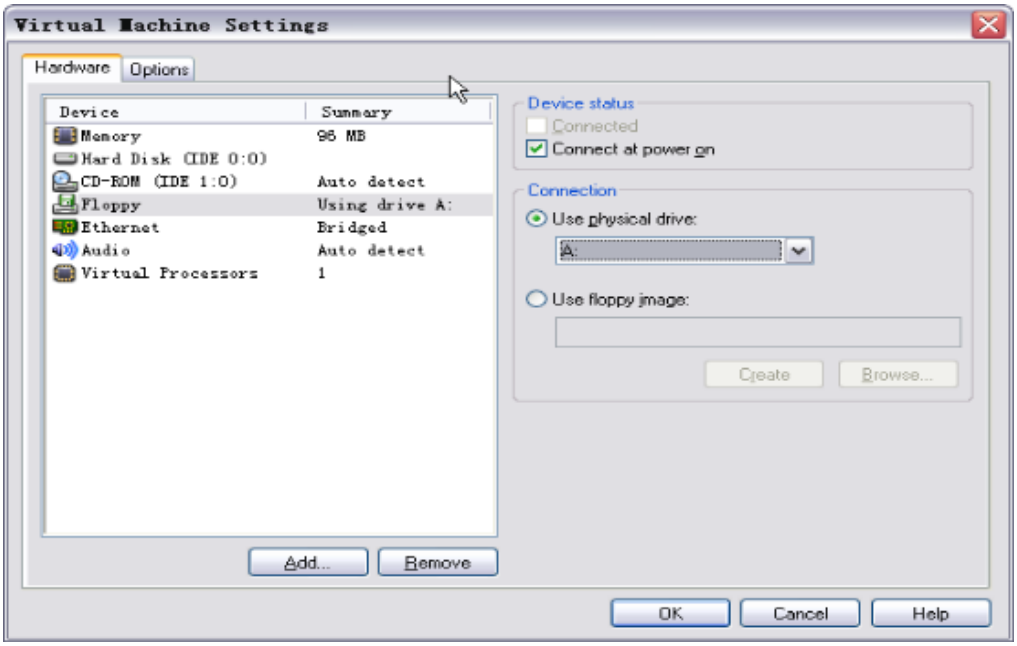
2）使用VMware向导建立一个针对VxWorks的虚拟机；此步骤中注意在操作系统中选择“other”



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，下图演示了用RamDiskNT虚拟一个1.44M软盘的方法。



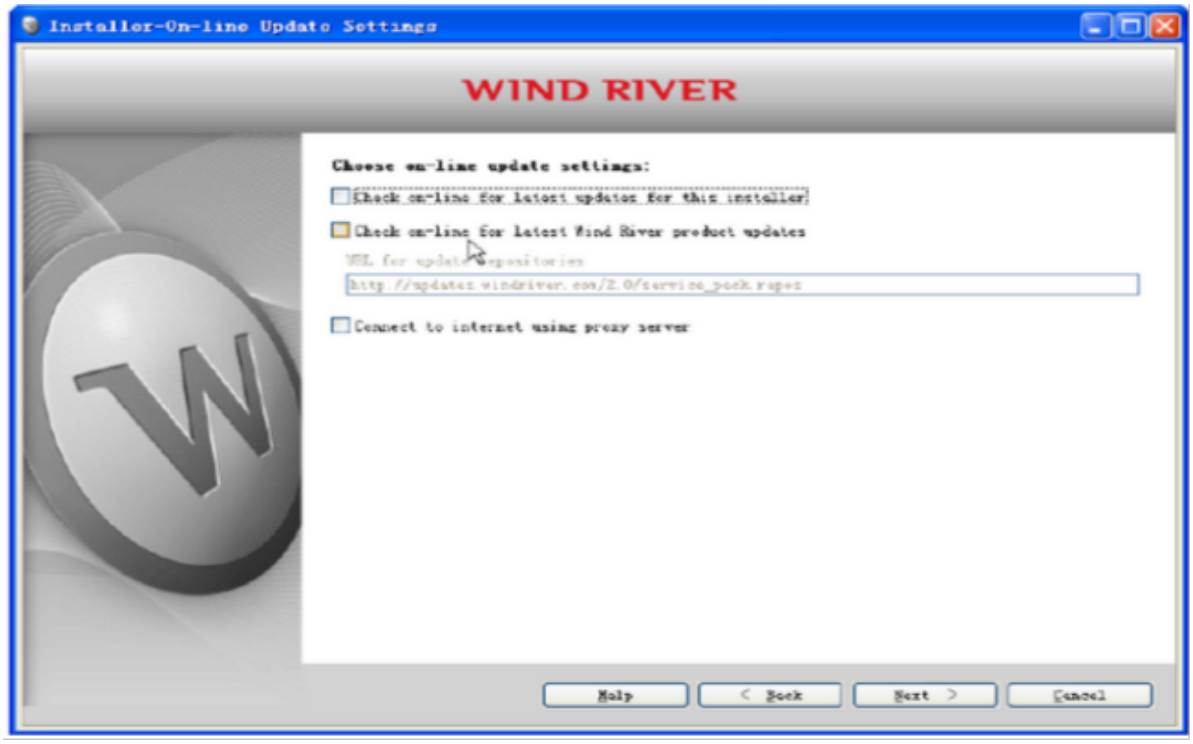
仅仅虚拟一个软驱是不够的，把这个软驱添加到我们建立的虚拟机中才算修成正果，图7演示了添加软驱后的虚拟机硬件设置。



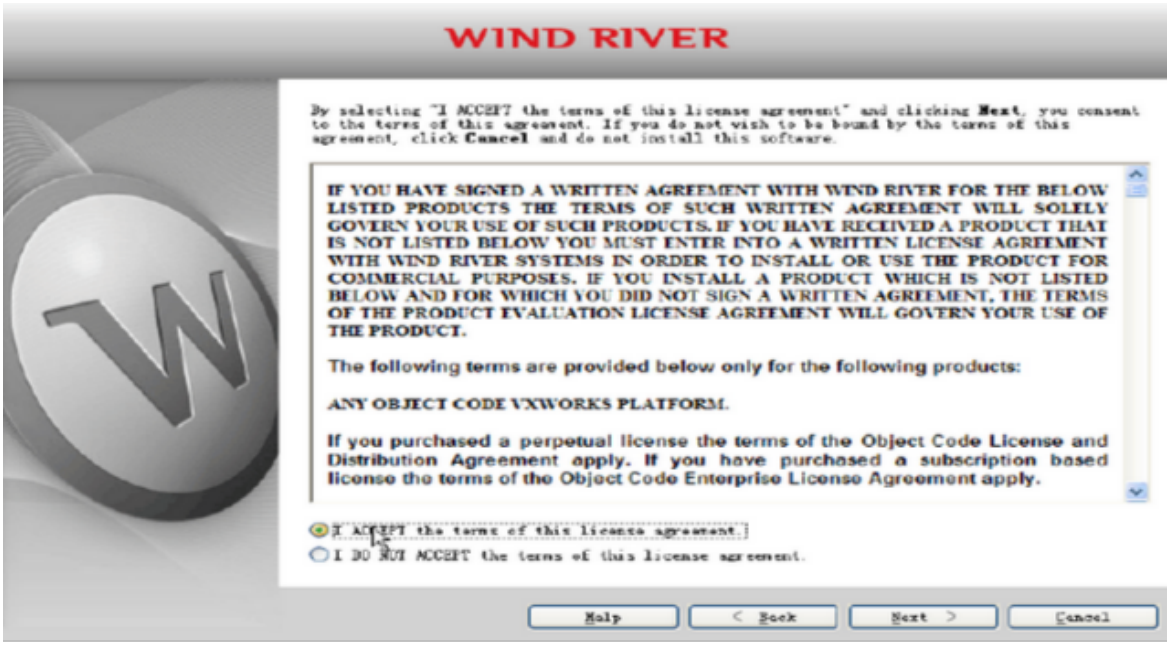
三. 安装Workbench

Workbench 3安装过程比较简容易如下操作:

“Next”前面两项默认是勾选，取消勾选，然后”Next”

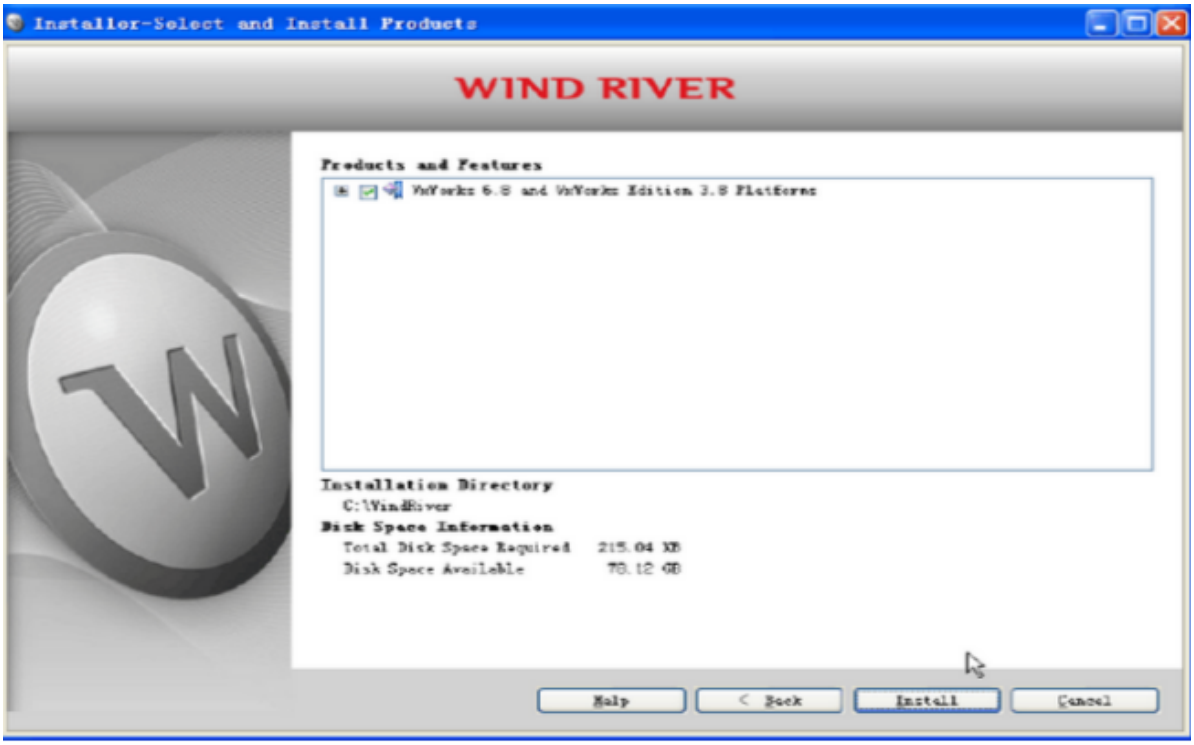


勾选”ACCEPT”项，然后”Next”

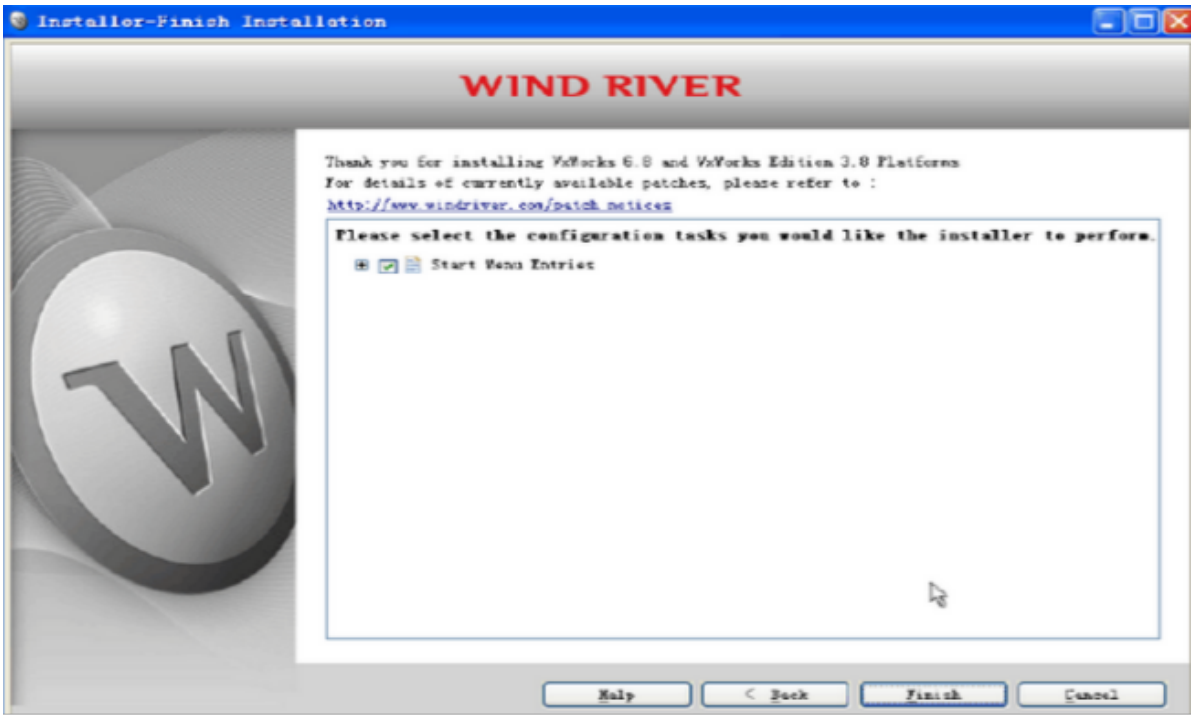


勾选“Inter”选项，然后“Next”

按“Install”



按“Finish”即可，至此完成Workbench安装。



四. 构造BSP

- **修改配置文件**

首先，我们复制一份纯净未做修改的 D:\WindRiver-GPPVE-3.6-IA-Eval\vxworks-6.6\target\config\下的 pcPentium3 文件夹，改名为 Pentium3（名称随便你起），再将该改名后文件夹放到config目录下，最好不要带空格或什么特殊符号。 修改编译 bootrom、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 bootrom 和 VxWorks 的映象符合我们的要求，然后修改config.h。

定位到目录 D:\WindRiver-GPPVE-3.6-IA-Eval\vxworks-6.6\target\config\Pentium3并打开 该目录下 Config.h 文件;

1) 定位到”INCLUDE*CPU*PROBE”,更改成如下：

```
#undef  INCLUDE_CPU_PROBE
#ifndef INCLUDE_CPU_PROBE
#   undef  CPU
#   define CPU      PENTIUM3
```

2) 查找到定义 DEFAULT*BOOT*LINE 宏的地方，修改预处理条件 CPU == PENTIUM3 分支下的定义如下：

```
"lnPci(0,0)host: vxWorks h=172.18.101.121 e=172.18.101.124:ffffff00 u=target pw=target tn=target"
```

这里的host地址得修改成你的主机地址，target地址只需要在同一个网段内就可以了 。

3) 定位到INCLUDE*ATA*，*在前面一句添加#undef INCLUDE*FD,如下所示：

```
#undef  INCLUDE_FD              /* include floppy disk driver */
#undef  INCLUDE_ATA             /* include IDE/EIDE(ATA) hard disk

#undef  INCLUDE_LPT             /* include parallel port driver */

#undef  INCLUDE_TFFS            /* include TrueFFS driver for Flash
#undef  INCLUDE_PCMCIA          /* include PCMCIA driver */
```

4) 定位到/* Network driver options: VxBus drivers */ 作如下修改：

```
/* Network driver options: VxBus drivers */

#undef  INCLUDE_AM79C97X_VXB_END
#undef  INCLUDE_AN983_VXB_END
#undef  INCLUDE_FEI8255X_VXB_END
#undef  INCLUDE_GEI825XX_VXB_END
#undef  INCLUDE_MVYUKONII_VXB_END
#undef  INCLUDE_MVYUKON_VXB_END
#undef  INCLUDE_NS8381X_VXB_END
#undef  INCLUDE_RTL8139_VXB_END
#undef  INCLUDE_RTL8169_VXB_END
#undef  INCLUDE_TC3C905_VXB_END
#undef  INCLUDE_NE2000_VXB_END
```

5) 定位到INCLUDE*LN*97X_END，将其定义上，修改如下：

```
#define INCLUDE_END             /* Enhanced Network Driver Support

#undef  INCLUDE_DEC21X40_END    /* (END) DEC 21x4x PCI interface */
#undef  INCLUDE_EL_3C90X_END    /* (END) 3Com Fast EtherLink XL PCI
#undef  INCLUDE_ELT_3C509_END   /* (END) 3Com EtherLink III interfa
#undef  INCLUDE_ENE_END         /* (END) Eagle/Novell NE2000 interf
#undef  INCLUDE_ULTRA_END       /* (END) SMC Elite16 Ultra interfac
#undef  INCLUDE_GEI8254X_END    /* (END) Intel 82543/82544 PCI inte
#define INCLUDE_LN_97X_END      /* (END) AMD 79C97x PCI interface *
```

6) 定位到INCLUDE*PC*CONSOLE，将其设成定义的，如下所示

```
#define INCLUDE_PC_CONSOLE              /* PC keyboard and VGA co
#ifdef INCLUDE_PC_CONSOLE
#   define PC_CONSOLE           (0)     /* console number */
#   define N_VIRTUAL_CONSOLES   (2)     /* shell / application */
#   ifdef INCLUDE_VXBUS
```

7) 定位到#if (SYS*WARM*TYPE == SYS*WARM*BIOS)，修改如下

```
#if (SYS_WARM_TYPE == SYS_WARM_BIOS)            /* non-volatile RAM
#   define NV_RAM_SIZE             (NONE)
#else
#   define NV_RAM_SIZE             (0x1000)
#endif

#ifdef NV_RAM_SIZE
#   undef  NV_RAM_SIZE
#   define NV_RAM_SIZE             (NONE)
#endif
```

修改 ConfigNet.h

8) 定位到/ *max number of END ipAttachments we can have* / 在上面添加如下内容：

```
/* Am79C97x (lnPci) driver defines */
#ifdef INCLUDE_LN_97X_END
#define LN_97X_LOAD_FUNC      sysLn97xEndLoad
#define LN_97X_BUFF_LOAN      TRUE #define LN_97X_LOAD_STR""
IMPORT END_OBJ * LN_97X_LOAD_FUNC (char *, void *);
#endif /* INCLUDE_LN_97X_END */
```

如下图所示

```
#endif /* INCLUDE_ENE_END */

/* Am79C97x (lnPci) driver defines */

#ifdef INCLUDE_LN_97X_END

#define LN_97X_LOAD_FUNC      sysLn97xEndLoad
#define LN_97X_BUFF_LOAN      TRUE
#define LN_97X_LOAD_STR       ""

IMPORT END_OBJ * LN_97X_LOAD_FUNC (char *, void *);

#endif /* INCLUDE_LN_97X_END */

/* max number of END ipAttachments we can have */

#ifndef IP_MAX_UNITS
#   define IP_MAX_UNITS (NELEMENTS (endDevTbl) - 1)
#endif
```

9) 定位到/ *Atheros AR521X WLAN Support* / 在上面添加如下内容：

```
#ifdef INCLUDE_LN_97X_END
    {0, LN_97X_LOAD_FUNC, LN_97X_LOAD_STR, LN_97X_BUFF_LOAN,     NULL, FALSE},
#endif /* INCLUDE_LN_97X_END */
```

截图如下：

```
#ifdef INCLUDE_LN_97X_END
    {0, LN_97X_LOAD_FUNC, LN_97X_LOAD_STR, LN_97X_BUFF_LOAN,
    NULL, FALSE},
#endif /* INCLUDE_LN_97X_END */

/* Atheros AR521X WLAN Support */

#ifdef INCLUDE_AR521X_END
    {-1, END_TBL_END, NULL, 0, NULL, FALSE}, /* up to 4 Atheros
    {-1, END_TBL_END, NULL, 0, NULL, FALSE},
```

10) 将D:\WindRiver-GPPVE-3.6-IA-Eval\vxworks-6.6\target\src\drv\end目录下的ln97xEnd.c拷贝进你的Pentium3目录下，并作如下修改： 定位到

```
do
{
/* poll for suspend mode entry */
} while ((csrLockedRead (pDrvCtrl, CSR(5)) & CSR5_SPND) == 0);
```

添加宏如下所示：

```
#if !defined(VMWARE_HACK)
    do  {
        ;  /* poll for suspend mode entry */
        }
    while ((csrLockedRead (pDrvCtrl, CSR(5)) & CSR5_SPND) == 0);
#endif
```

将5.5的Sysln97xEnd.c移植到你的Pentium3目录下，并作如下修改：

11) 定位到/ *map a 4Kb 32-bit non-prefetchable memory address decoder* / 添加如下宏：

```
#if !defined(VMWARE_HACK)
    /* map a 4Kb 32-bit non-prefetchable memory address decoder */

    if (sysMmuMapAdd ((void *)(memIo32 & PCI_DEV_MMU_MSK),
        PCI_DEV_ADRS_SIZE, VM_STATE_MASK_FOR_ALL, VM_STATE_FOR_PCI)
        {
        return (ERROR);
        }
#endif
```

12) Boot定位到ln97xPciResources[ln97XUnits].bar[0] = ioBase；添加如下宏：

```
#if !defined(VMWARE_HACK)
    /* map a 4Kb 32-bit non-prefetchable memory address decoder */

    if (sysMmuMapAdd ((void *)(memIo32 & PCI_DEV_MMU_MSK),
        PCI_DEV_ADRS_SIZE, VM_STATE_MASK_FOR_ALL, VM_STATE_FOR_PCI)
        {
        return (ERROR);
        }
#endif
```

修改makefile文件，作如下修改：

13) 定位到TOOL = diab 将diab修改为gnu

```
    /* update the board-specific resource table */

    ln97xPciResources[ln97XUnits].bar[0]     = ioBase;
#if defined(VMWARE_HACK)
    ln97xPciResources[ln97XUnits].bar[1]     = (UINT32)NONE;
#else
    ln97xPciResources[ln97XUnits].bar[1]     = memIo32;
#endif
    ln97xPciResources[ln97XUnits].irq        = irq;
    ln97xPciResources[ln97XUnits].irqvec     = INT_NUM_GET (irq);
```

14) 定位到EXTRA_DEFINE，作如下修改：

```
EXTRA_DEFINE        = -DVMWARE_HACK -DFAST_REBOOT

MACH_EXTRA          = ln97xEnd.o
```

修改sysnet.c

15)定位到# include "sysUltraEnd.c" 添加内容如下：

```
#ifdef  INCLUDE_END
#   include "sysDec21x40End.c"          /* dec21x40End support rou
#   include "sysElt3c509End.c"          /* elt3c509End support rou
#ifdef INCLUDE_ENE_END
#   include "sysNe2000End.c"            /* ne2000End support routi
#endif
#ifdef INCLUDE_LN_97X_END
#   include "sysLn97xEnd.c"             /* ln97xEnd support routin
#endif
#   include "sysUltraEnd.c"            /* ultraEnd support routin
#endif  /* INCLUDE_END */
```

16) 定位到LOCAL VEND*ID*DESC vendorIdEnet [] = ,修改内容如下：

```
LOCAL VEND_ID_DESC vendorIdEnet [] =
    {
    #if defined(INCLUDE_DEC21X40_END)
    {DEC_PCI_VENDOR_ID,      sysDec21x40PciInit},
    #endif /* INCLUDE_DEC21X40_END */

    #if defined(INCLUDE_LN_97X_END)
    {AMD_PCI_VENDOR_ID,      sysLan97xPciInit},
    #endif /* INCLUDE_LN_97X_END */

    {0xffffffff, NULL}        /* last entry */
    };
```

17) 将安装目录下的wrenv.exe拷贝进Pentium3目录下，删除如下文件：

```
bootrom;
bin;
bootrom.pxe;
vxWorks;
vxWorks.st;
vxWorks.sym;
```

并添加两个文件如下：

torVars.bat 内容为： `wrenv -p vxworks-6.6`

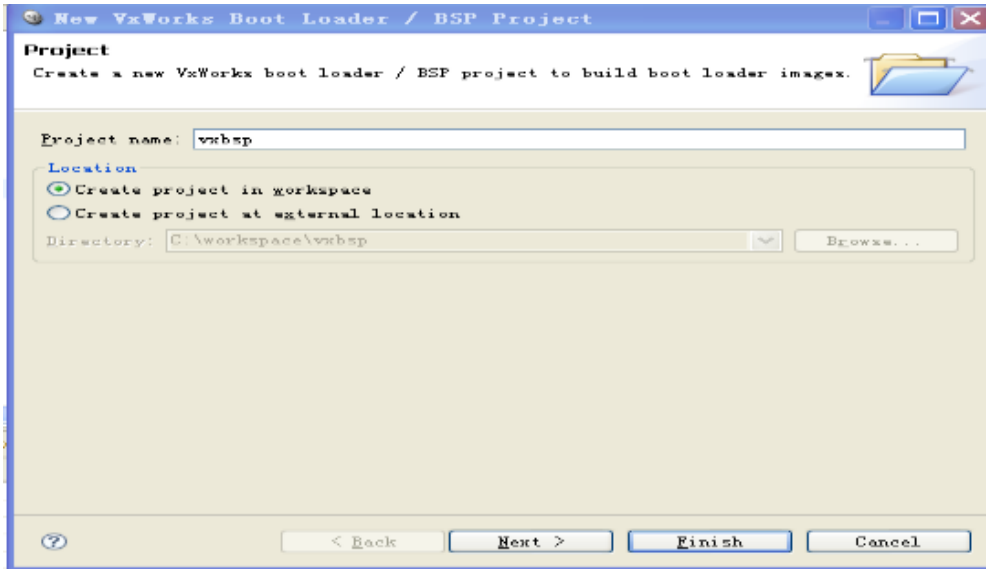
vxcopy.bat 内容为： `objcopypentium -O binary --gap-fill=0 %1 %2`

- **编译Bootrom**

往PATH环境变量添加路径，如下所示

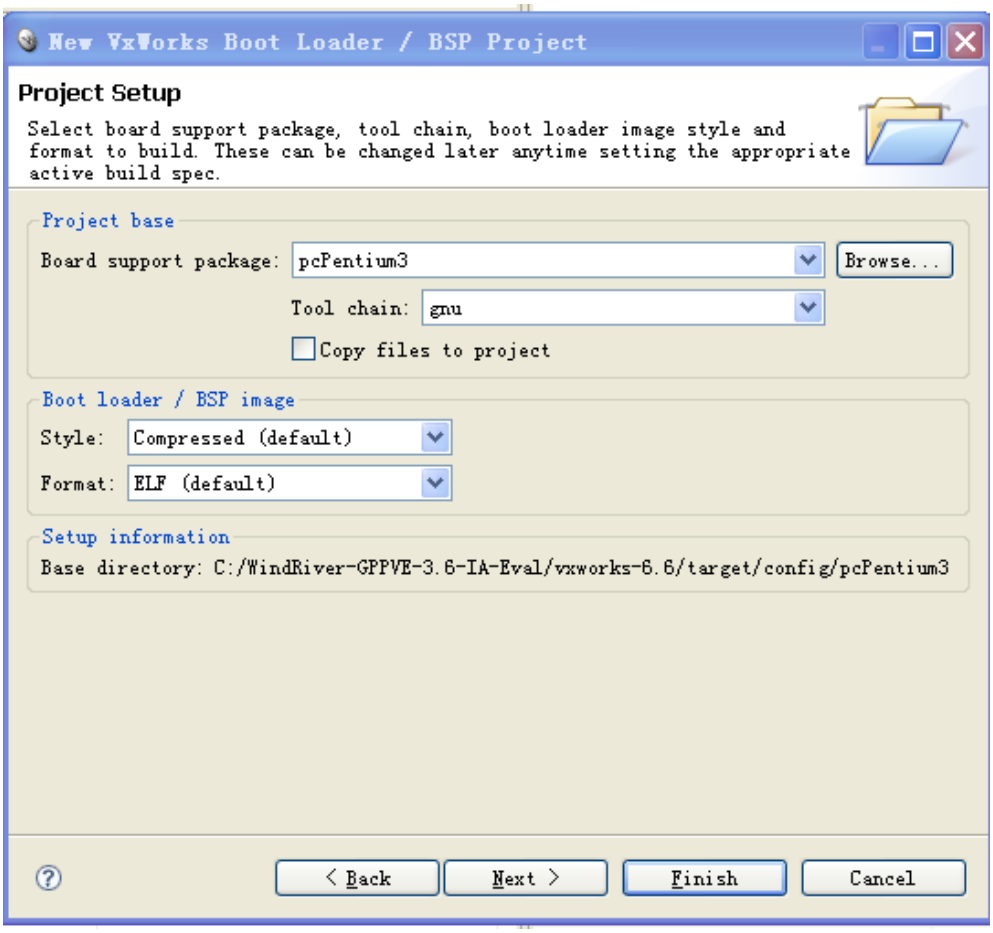
```
D:\WindRiver-GPPVE-3.6-IA-Eval\vxworks-6.6\host\x86-win32\bin;D:\WindRiver-GPPVE-3.6-IA-Eval
\workbench-3.0\x86-win32\bin
```

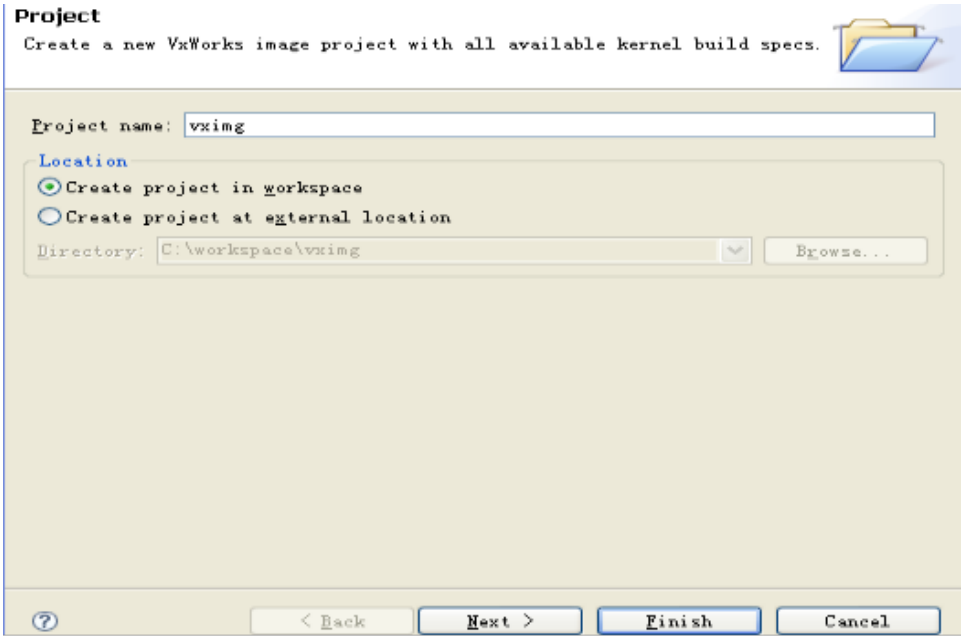
打开Wind River Workbench 3.0，File->New -> VxWorks BootLloader / BSP Project， 填写工程名vxbsp，并选择我们的Pentium3文件夹，及工具gnu,然后编译：



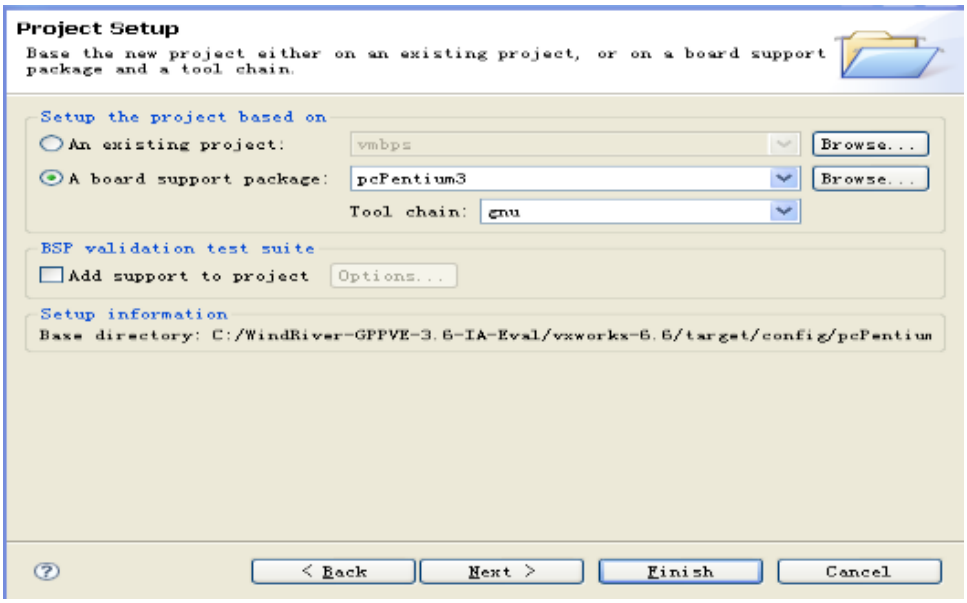
- **编译VxWorks映像**

打开Wind River Workbench 3.0，File->New-> VxWorks Image Project。

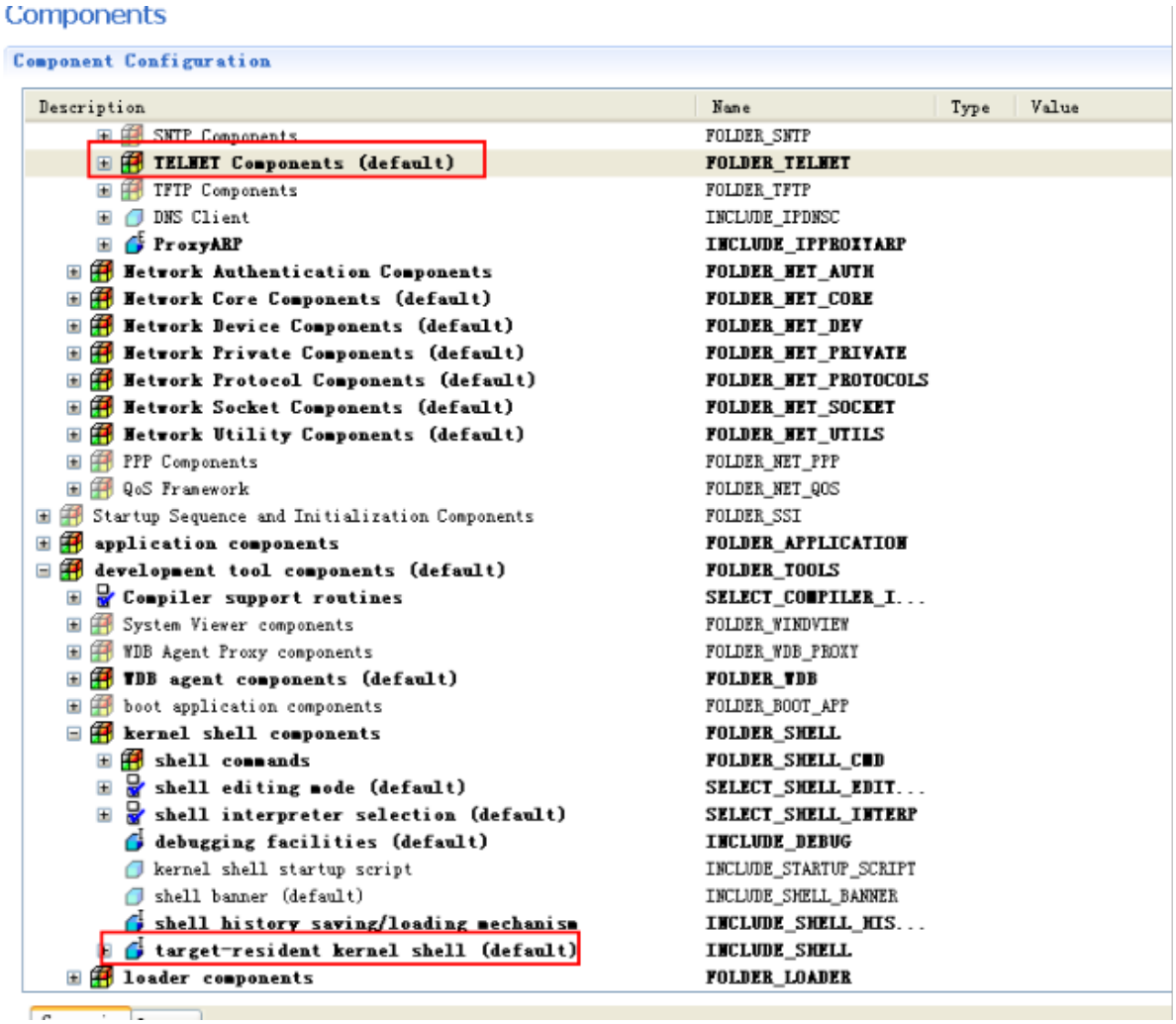




建立基于pcPentium BSP的VxWorks映像



选择需要的VxWorks组件



在这个例子中我们需要包括两个重要的组件：Telnet Components 和 Target shell。前者使我们可以通过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。

- **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运行映象。打开CMD,进去目录 Pentium3文件夹，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

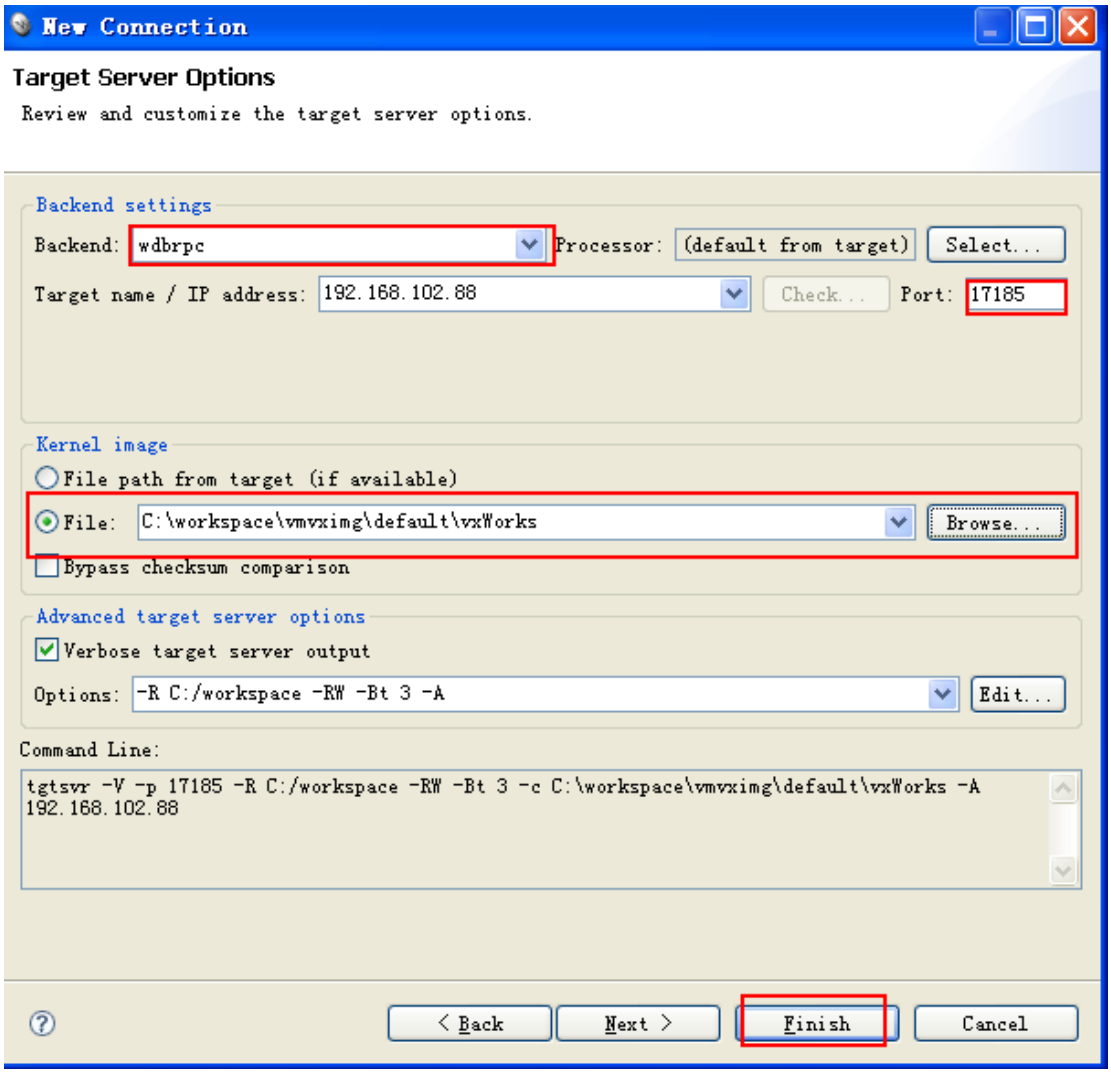
- **配置FTP服务器**

这里的FTP服务器用于在系统成功引导后，下载VxWorks的运行时映象。我们这里使用Workbench开发环境自带的FTP服务器。如下图

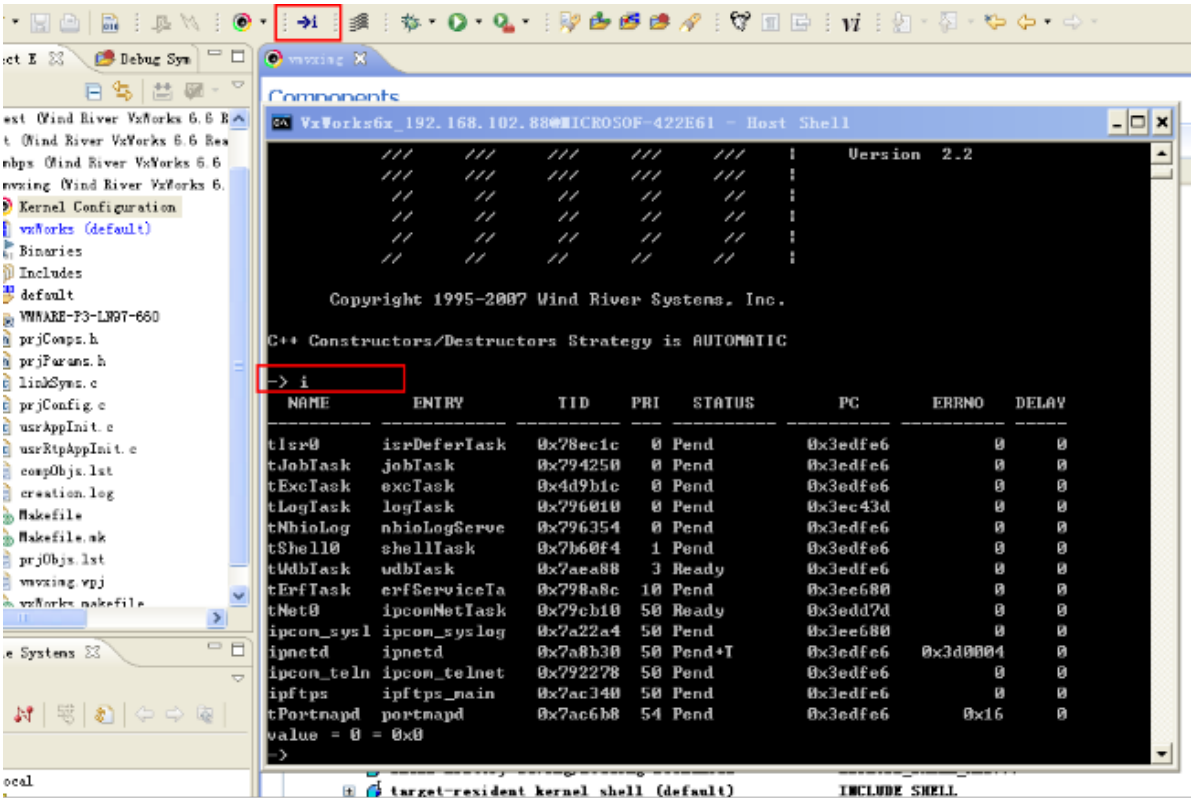
打开Tornado FTP Server，选择“Security”菜单下的“Users/Rights”子菜单，创建User Name为“target”，修改“Home Directory”为Pentium3目录（此路径由上面的DEFAULT*BOOT*LINE参数决定），同时修改口令为“target”，最后点击“Done”按钮完成修改。

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Pentium3目录映像，服务器的FTP会有相关的log。

- **配置target server**
- 打开您的Workbench开发环境，选择“Target->Connection...”菜单



需要说下的Target name/ IP address:为Vxworks目标的IP地址 我们来看调试器成功连接上Vxworks效果，点击工具栏中 >i 图标，弹出Vxworks SHELL，效果如下图：


Debug Sym
Kernel Configuration
vxWorks (default)
Binaries
Includes
default
VMWARE-P3-LW97-660
prjComps.h
prjParams.h
linkSyms.c
prjConfig.c
usrAppInit.c
usrRtpAppInit.c
compObjs.lst
creation.log
Makefile
Makefile.mk
prjObjs.lst
vmvxing.vpj
Components
VxWorks6x_192.168.102.88@MICROSOF-422E61 - Host Shell
Version 2.2
Copyright 1995-2007 Wind River Systems, Inc.
C++ Constructors/Destructors Strategy is AUTOMATIC
-> i
NAME ENTRY TID PRI STATUS PC ERRNO DELAY
tIsr0 isrDeferTask 0x78ec1c 0 Pend 0x3edfe6 0 0
tJobTask jobTask 0x794250 0 Pend 0x3edfe6 0 0
tExcTask excTask 0x4d9b1c 0 Pend 0x3edfe6 0 0
tLogTask logTask 0x796010 0 Pend 0x3ec43d 0 0
tNbioLog nbioLogServe 0x796354 0 Pend 0x3edfe6 0 0
tShell0 shellTask 0x7b60f4 1 Pend 0x3edfe6 0 0
tWdbTask wdbTask 0x7aea88 3 Ready 0x3edfe6 0 0
tErfTask erfServiceTa 0x798a8c 10 Pend 0x3ee680 0 0
tNet0 ipcomNetTask 0x79cb10 50 Ready 0x3edd7d 0 0
ipcom_sysl ipcom_syslog 0x7a22a4 50 Pend 0x3ee680 0 0
ipnetd ipnetd 0x7a8b30 50 Pend+T 0x3edfe6 0x3d0004 0
ipcom_teln ipcom_telnet 0x792278 50 Pend 0x3edfe6 0 0
ipftps ipftps_main 0x7ac340 50 Pend 0x3edfe6 0 0
tPortmapd portmapd 0x7ac6b8 54 Pend 0x3edfe6 0x16 0
value = 0 = 0x0
->
target-resident kernel shell (default)
INCLUDE SHELL

接下来，开工吧!