实验 1　ubuntu 搭建交叉编译环境

作者：宋德锋(song_df@qq.com)

1　实验目的

1.1　如何在 ubuntu 系统中搭建交叉编译环境。

1.2　如何编译出可以在开发板上运行的应用程序。

1.3　如何将交叉编译出来的应用程序放到开发板上运行。

2　实验内容

2.1　在 ubuntu 系统中搭建交叉编译环境。

2.2　交叉编译 HelloC4Arm。

2.3　让编译出的 HelloC4Arm 在开发板上运行。

3　实验设备

3.1　运行有 ubuntu 的 PC 机或者虚拟机。本文以 VMware8.0 中运行 Ubuntu11.10 为例。

3.2　汇文 2440 开发板。

3.3　USB 转串口线。

3.4　网络传输线。

4　所需资料

4.1　UTU2440-F-V4.5-T35B.iso

5　实验步骤

5.1　将 UTU2440-F-V4.5-T35B.iso 作为 VMware 的虚拟光盘挂载：

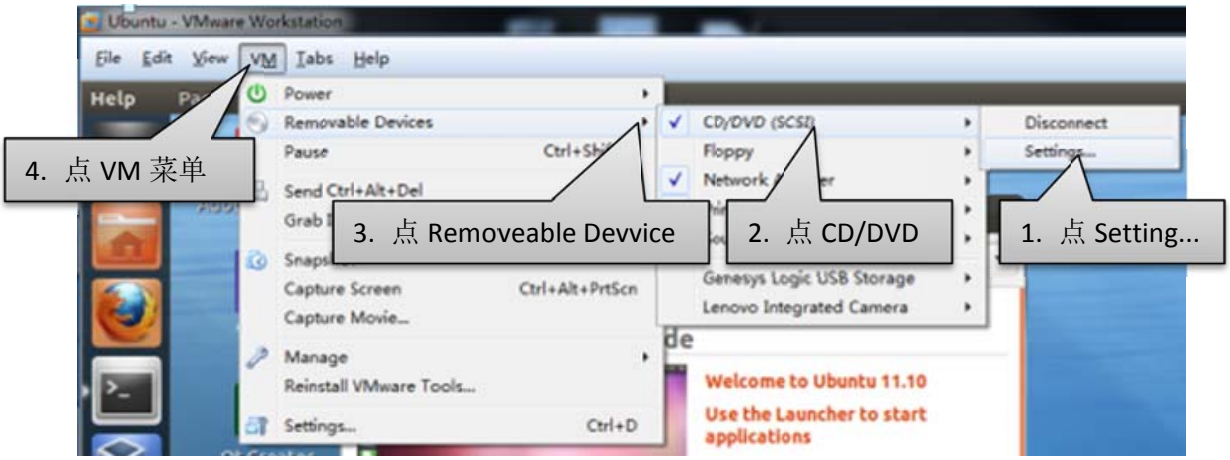


图 1-1 VMware 挂载 ISO 文件为虚拟光盘

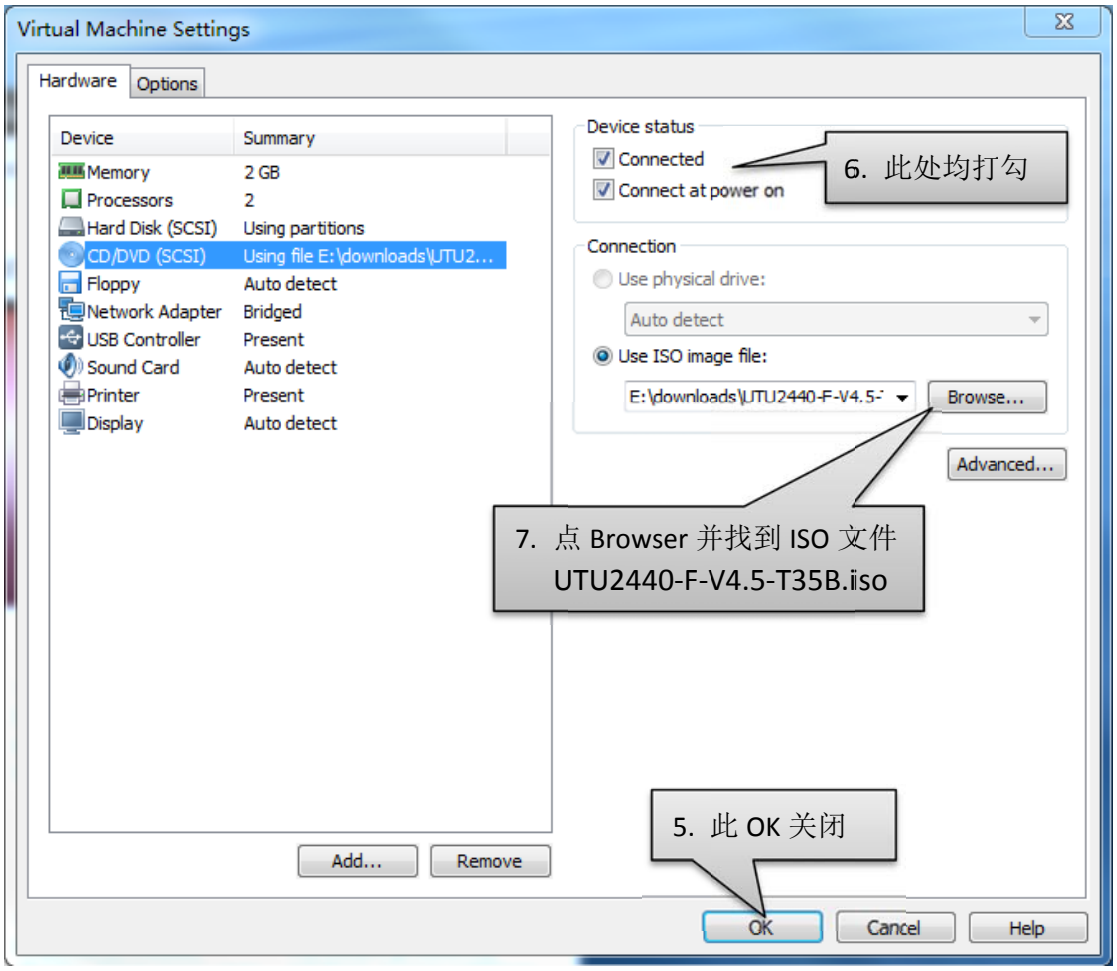
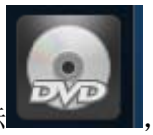


图 1-2 VMware CD/DVD 设置窗口

设置完成 ISO 文件后，会在 Ubuntu 桌面出现光盘图标  ，点开此图标，可以看到光盘内容，其实这个 ISO 文件会被挂载在/media 目录中，在命令行输入命令可以看到所挂载的光盘：

```
$ ls -l /media
dr-x------ 1 bryan bryan 2048 2009-07-28 15:15 UTU2440-F-V4.5_090725
```

命令 1-1 列出所挂载 ISO 位置

5.2 解压安装交叉编译器

```
$ sudo tar xjvf /media/UTU2440-F-V4.5_090725/UTU2440-F-V4.5_090725/utuLinux_for2440/arm-linux-gcc/arm-linux-gcc-3.4.1.tar.bz2 -C /
$ ls /usr/local/arm/3.4.1/bin/
arm-linux-addr2line  arm-linux-cpp      arm-linux-gcov
arm-linux-ranlib
arm-linux-ar         arm-linux-g++      arm-linux-ld
arm-linux-readelf
arm-linux-as         arm-linux-gcc      arm-linux-nm
arm-linux-size
arm-linux-c++        arm-linux-gcc-3.4.1  arm-linux-objcopy
arm-linux-strings
arm-linux-c++filt    arm-linux-gccbug   arm-linux-objdump
arm-linux-strip
```

命令 1-2

5.3 将交叉编译器路径添加到系统工作目录：

刚刚解压得到的 arm-linux-gcc 交叉编译器并不在系统工作目录中，为了后面使用的方便，以及项目的交叉编译的需求，要将这个工作目录设置到系统工作 PATH 中。

使用文本编辑器打开~/.bashrc，在文本最后添加如下内容：

```
PATH=/usr/local/arm/3.4.1/bin:$PATH
```

5.4 重新打开新的终端窗口，运行测试命令，将会得到交叉编译器的版本信息：

```
$ arm-linux-gcc --version
arm-linux-gcc (GCC) 3.4.1
Copyright (C) 2004 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

命令 1-3

5.5 编写 HelloC4Arm.c 文件，内容如下：

```
1    #include <stdlib.h>
2    #include <stdio.h>
3    int main(void)
4    {
5          printf(“hello c 4 arm\n”);
6          return 0;
7    }
```

代码 1-1 HelloC4Arm.c

5.6 编辑好后使用如下命令进行编译：

```
$    arm-linux-gcc -o HelloC4Arm HelloC4Arm.c
$    ls
HelloC4Arm HelloC4Arm.c
```

命令 1-4 交叉编译 HelloC4Arm.c

5.7 将编译好的可执行文件 HelloC4Arm 传送到开发板运行

要运行命令 1-4 生成的可执行文件 HelloC4Arm，必须要在开发板上运行，因为这个可执行文件的机器指令只能被 ARM 识别，如果直接在 PC 上执行这个它会出现如下的错误：

```
$    ./HelloC4Arm
bash: ./HelloC4Arm: cannot execute binary file
```

要将文件传送到开发板，可以使用网络传递，在开发板的 Linux 正常起动的情况下，开发板会启动一个 FTP 服务器程序，这时可以在 Ubuntu 上运行 FTP 命令，把可执行文件发送到开发板，也可以使用 FTP 的客户端进行发送。如果发送成功，会在开发板的 /var/ftp/pub 中看到上传的文件，然后就可以执行它。如果没有可执行权限，可以加上可执行权限。

6 实验作业

安实验步骤练习，并掌握目的所提内容。

实验 2 ubuntu 搭建 NFS 环境

1 实验目的

1.1 如何在 ubuntu 系统中搭建 NFS 环境。

1.2 配置开发板可以运程挂到 NFS 启动。

1.3 学会手动和自动挂载运程共享。

2 实验内容

2.1 在 ubuntu 系统中搭建 NFS 环境。

2.2 配置出开发板可以使用的文件系统目录。

2.3 挂载使用网络共享。

3 实验设备

3.1 运行有 ubuntu 的 PC 机或者虚拟机。本文以 VMware8.0 中运行 Ubuntu11.10 为例，并且网络已经配置到桥接方式。

3.2 汇文 2440 开发板。

3.3 USB 转串口线。

3.4 网络传输线。

4 所需资料

4.1 UTU2440-F-V4.5-T35B.iso

5 实验步骤

5.1 在 ubuntu 上安装 NFS 服务：

```
$    sudo apt-get install nfs-kernel-server
```

命令 2-1 安装 NFS 服务

如果 NFS 服务安装成功，会在/etc 下生成配置脚本文件 exports，exports 文件可以用来配置 NFS 服务共享目录。

5.2 在自己 ubuntu 的工作目录中创建目录 utuLinux2.6.24，然后把这个目录设置为共享目录：

```
$    cd
$    mkdir utuLinux2.6.24
$    sudo gedit /etc/exports
```

命令 2-2 修改 NFS 配置脚本 exports

在打开的 exports 文档的最后添加一行配置信息：

```
/home/bryan/utuLinux2.6.24      *(rw,sync,no_root_squash)
```

代码 2-1 设置目录共享的 exports 配置脚本行

注意这里的红色部分，各自要根据自己的用户名写出自己的工作路径。

这里的 *(rw,sync,no_root_squash) 就是共享配置：

* 表示所有客户机都可以挂载上述共享目录。

rw 表示挂接此目录的客户机对该目录有读写的权限，

no_root_squash 表示如果客户端以 root 用户连接时，root 权限不会被削减，如果换成 root_squash，则当客户端以 root 用户连接时，客户端对这个目录的操作会用 nobody 的替代。

5.3 在改变了 exports 配置文件后，如果要让配置生效，则要把 NFS 服务重新启动，操作命令：

```
$    sudo service nfs-kerner-server restart
```

命令 2-3 重启 NFS 服务

5.4 要测试刚刚的配置是否成功，可以在 ubuntu 上直接进行共享目录的绑定测试：

```
$    sudo mount -t nfs localhost:/home/bryan/utuLinux2.6.24 /mnt
$    df
```

命令 2-4 对设置的共享目录进行挂载测试

此命令执行成功没有任何提示，但这时候的/mnt 目录已经和/home/bryan/utuLinux2.6.24 目录同步了，在其中一个目录中的变化，在另一个目录中马上可以看到。

5.5 同样也可以让开发板上的 Linux 来绑定这个共享目录，只要在开发板启动完成后的 busybox 命令行中输入如下命令：

```
$    mount -t nfs 192.168.1.200:/home/bryan/utuLinux2.6.24 /mnt
```

命令 2-5 在开发板上挂载共享目录

如果执行成功，此时开发板 Linux 系统中的根目录下的/mnt 目录和 ubuntu 上被设置为共享的目录/home/bryan/utuLinux2.6.24 保持同步。

5.6 如果要提供一个可以让开发板作为远程文件系统的目录，则要在 /home/bryan/utuLinux2.6.24 中创建一个开发板可以使用的 linux 文件系统目录结构及所要使用的系统配置和命令文件系统，我们可以使用 utu2440-nogui.tar.gz 文件，此文件中已经创建好汇文开发板可以使用的文件系统，按照如下命令进行文件释放：

```
$    cd /home/bryan/utuLinux2.6.24
$    mkdir s3c2440_recover && cd s3c2440_recover
$    tar xzvf ~/utu2440-nogui.tar.gz
$    ls
$
```

命令 2-6 解包生成开发板的远程文件系统目录

正确执行完成后，会在/home/bryan/utuLinux2.6.24/s3c2440_recover 目录中出现 linux 根文件系统。这个文件系统就是提供给开发板启动的运程文件系统。

5.7 要让开发板自动挂载刚刚设置的文件系统，要修改开发板的 bootloader 的启动参数 bootargs，这个参数用来告诉 linux 内核启动时的一些重要参数，如果要让开发板自动挂载远程文件系统，则要在 bootloader 的命令模式下进行如下设置：

```
bootloader=>>> setenv bootargs 'console=ttySAC0
root=/dev/nfs
nfsroot=192.168.1.200:/home/bryan/utuLinux2.6.24/s3c2440
_recover
ip=192.168.1.168:192.168.1.200:192.168.1.1:255.255.255.0
:www.huiwen.com:eth0:off'
bootloader=>>> saveenv
```

命令 2-7 设置并保存开发板自动挂载远程文件系统的参数

6 常见问题：

6.1 Q：挂载根文件时，提示“Warning: unable to open an initial console.”，最终启动失败。

A：通过分析知道内核启动时需要成功打开/dev/console 设备，否则给出"Warning: unable to open an initial console."的错误提示。所以要为目标机配置 console 设备文件节点，配置方法是在 ubuntu 所设置的共享文件目录中的 dev/目录中创建出名为 console 的设备节点即可解决此问题，所以执行命令如下：

```
$ cd /home/bryan/utuLinux2.6.24/s3c2440_recover/dev
$ sudo mknod console c 5 1
$ ls -l /home/bryan/utuLinux2.6.24/s3c2440_recover/dev/console
$
```

命令 2-8 为远程文件系统创建 console 设备节点

6.2 Q: 开发板 Linux 在启运过程中报如下的错误，然后停止：

```
IP-Config: Complete:
      device=eth0, addr=192.168.1.168, mask=255.255.255.0,
gw=192.168.1.1,
     host=www, domain=, nis-domain=yctek.com,
     bootserver=192.168.1.200, rootserver=192.168.1.200, rootpath=
Looking up port of RPC 100003/2 on 192.168.1.200
rpcbind: server 192.168.1.200 not responding, timed out
Root-NFS: Unable to get nfsd port number from server, using default
Looking up port of RPC 100005/1 on 192.168.1.200
rpcbind: server 192.168.1.200 not responding, timed out
Root-NFS: Unable to get mountd port number from server, using default
mount: server 192.168.1.200 not responding, timed out
Root-NFS: Server returned error -5 while mounting
/home/bryan/utuLinux2.6.24/s3c2440_recover_qtopia
VFS: Unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device "nfs" or unknown-block(2,0)
Please append a correct "root=" boot option; here are the available
partitions:
1f00          384 mtdblock0 (driver?)
1f01         2048 mtdblock1 (driver?)
1f02        63088 mtdblock2 (driver?)
Kernel panic - not syncing: VFS: Unable to mount root fs on
unknown-block(2,0)
```

出现这种问题是 NFS 服务器连接不上，检查 IP 设置及网络连接是否畅通，还有 NFS 服务是否正常开启。如果要检查网络是否连通，可以在 bootloader 里使用 ping 命令测试网络连通情况。如果是 NFS 服务问题，可以重启一下 NFS 服务。

```
IP-Config: Complete:
      device=eth0, addr=192.168.1.168, mask=255.255.255.0, gw=192.168.1.1,
     host=www, domain=, nis-domain=yctek.com,
     bootserver=192.168.1.200, rootserver=192.168.1.200, rootpath=
Looking up port of RPC 100003/2 on 192.168.1.200
Looking up port of RPC 100005/1 on 192.168.1.200
Root-NFS: Server returned error -13 while mounting
/home/bryan/utulinux2.6.24/s3c2440_recover_qtopia
VFS: Unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device "nfs" or unknown-block(2,0)
Please append a correct "root=" boot option; here are the available partitions:
1f00          384 mtdblock0 (driver?)
1f01         2048 mtdblock1 (driver?)
1f02        63088 mtdblock2 (driver?)
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(2,0)
```

出现这种情况是因为在 bootloader 里的 bootargs 的设置中存在路径设置错误，即开发板所设置的使用路径在 NFS 里不存在或者没有设置为共享。检查 exports 路径设置和 bootargs 路径设置。