

# 实验五：Linux 交叉编译环境

## 一、实验目的

1. 理解交叉编译的原理和概念。
2. 掌握在 Linux 下搭建交叉编译平台的方法。
3. 掌握使用交叉编译平台编译源代码。

## 二、实验内容

1. 正确运行实验箱。
2. 通过串口线将实验箱和 PC 机连接。
3. 在 PC 机的 Linux 操作系统上搭建交叉编译平台，并编译程序。
4. 在实验箱上运行交叉编译程序结果。

## 三、实验设备

1. 硬件：PC 机；ARM9 系统教学实验系统；串口线；网线；NFS 服务器。
2. 软件：PC 机操作系统（Windows XP）；PuTTY 客户端；Ti 提供的交叉编译软件包 `mvltools5_0_0801921_update.tar.gz`。

## 四、预备知识

### 1. 交叉编译环境建立的原理

交叉编译是指，在某个主机平台上（比如 PC 上）建立交叉编译环境后，可在其他平台（如 ARM9 实验箱）上运行代码的过程。搭建交叉编译环境，即安装、配置交叉编译工具链。在该环境下编译出嵌入式 Linux 系统所需的操作系统、应用程序等，然后再上传到其他平台（如 ARM9 实验箱）上。

交叉编译工具链是为了编译、链接、处理和调试跨平台体系结构的程序代码。对于交叉开发的工具链来说，在文件名称上加了一个前缀，用来区别本地的工具链。例如，`arm_v5t_le` 表示是对 arm 的交叉编译工具链；`arm_v5t_le_gcc` 表示是

使用 gcc 的编译器。除了体系结构相关的编译选项以外，其使用方法与 Linux 主机上的 gcc 相同，所以 Linux 编程技术对于嵌入式同样适用。

gcc 和 arm-linux-gcc 的区别是什么呢？区别就是 gcc 是 linux 下的 C 语言编译器，编译出来的程序在本地执行，而 arm-linux-gcc 用来在 linux 下跨平台的 C 语言编译器，编译出来的程序在目标机(如 ARM9 实验箱)上执行，嵌入式开发应使用嵌入式交叉编译工具链。

## 2. NFS 服务器概述

NFS 是 Network File System 的缩写，即网络文件系统。一种使用于分散式文件系统的协定，由 Sun 公司开发，于 1984 年向外公布。功能是通过网络让不同的机器、不同的操作系统能够彼此分享个别的数据，让应用程序在客户端通过网络访问位于服务器磁盘中的数据，是在类 Unix 系统间实现磁盘文件共享的一种方法。

NFS 的基本原则是“容许不同的客户端及服务端通过一组 RPC 分享相同的文件系统”，它是独立于操作系统，容许不同硬件及操作系统的系统共同进行文件的分享。

NFS 在文件传送或信息传送过程中依赖于 RPC 协议。RPC(Remote Procedure Call) 远程过程调用能使客户端执行其他系统中程序的一种机制。NFS 本身是没有提供信息传输的协议和功能的，但 NFS 却能让我们通过网络进行资料的分享，这是因为 NFS 使用了一些其它的传输协议。而这些传输协议用到了 RPC 的功能，也可以说 NFS 本身就是使用 RPC 的一个程序，或者说 NFS 是一个 RPC SERVER。所以只要用到 NFS 的地方都要启动 RPC 服务，不论是 NFS SERVER 或者 NFS CLIENT，这样 SERVER 和 CLIENT 才能通过 RPC 来实现 PROGRAM PORT 的对应。可以这么理解 RPC 和 NFS 的关系：NFS 是一个文件系统，而 RPC 负责信息的传输。

## 五、实验步骤

前面的章节知识里，已经了解了交叉编译环境建立的原理，下面建立交叉编译环境。

**步骤 1：**登陆 PuTTY 服务器端（或者直接进入虚拟机终端）

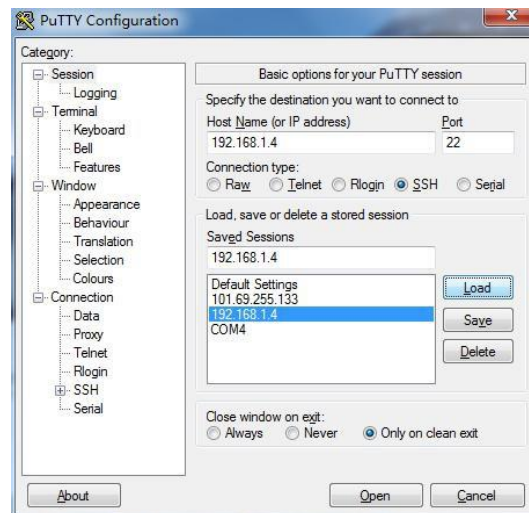


图 1 PuTTY 配置界面

如图 1，输入登陆名和登陆密码，即可登陆到服务器。

**步骤 2：**搭建交叉编译环境（先切换成管理员“sudo su”）

创建一个文件夹 mv\_pro\_5.0，进入文件夹 mv\_pro\_5.0，将/home/shiyan/2021（虚拟机是在/home/shiyan 目录）目录下的软件包 mvltools5\_0\_0801921\_update.tar 复制到当前目录 mv\_pro\_5.0 下（注意不能省略最后一条语句中的“.”，且前面有空格）：

```
#mkdir mv_pro_5.0
```

```
#cd mv_pro_5.0
```

```
服务器：#cp /home/shiyan/2021/mvltools5_0_0801921_update.tar.gz ./
```

```
虚拟机：#cp /home/shiyan/Desktop/shiyan/mvltools5_0_0801921_update.tar.gz ./
```

解压缩 mvltools5\_0\_0801921\_update.tar 软件包，解压缩后会出现 montavista 文件夹：

```
#tar zxvf mvltools5_0_0801921_update.tar.gz
```

配置系统环境变量，把交叉编译工具链的路径添加到环境变量 PATH 中去，使其可以在任何目录下使用，进入/etc/profile 文档：

```
#vim /etc/profile
```

点击插入键 i，在文件的最后一行添加：

```
export PATH=$PATH:/home/stX/mv_pro_5.0/montavista/pro/devkit/arm/v5t
```

```
_le/bin
```

（服务器）

```
export PATH=$PATH:/home/mv_pro_5.0/montavista/pro/devkit/arm/v5t
```

```
_le/bin
```

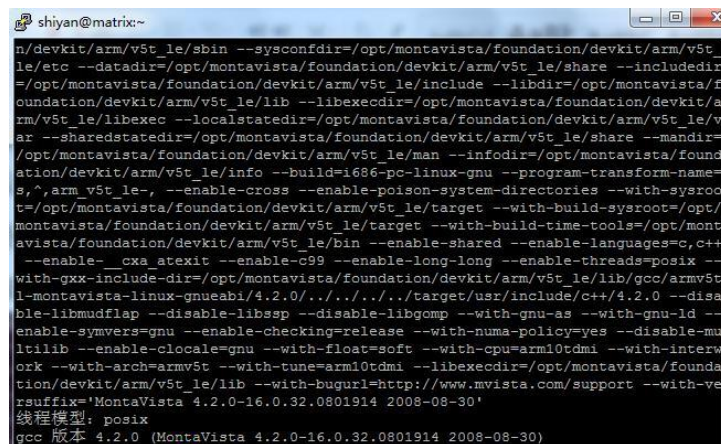
（虚拟机）

点击 esc 键退出输入，输入:wq!退出文档。使环境变量生效：

```
#source /etc/profile
```

检测交叉编译环境是否搭建成功：

在命令行中输入 `arm_v5t_le-gcc -v`，打印出图 2 中的版本信息，表示交叉编译环境搭建成功。



```
shiyao@matrix:~$ arm_v5t_le-gcc -v
Using built-in defaults.
Target: armv5t-le
Configured with:
  --sysconfdir=/opt/montavista/foundation/devkit/arm/v5t_le/etc
  --datadir=/opt/montavista/foundation/devkit/arm/v5t_le/share
  --includedir=/opt/montavista/foundation/devkit/arm/v5t_le/include
  --libdir=/opt/montavista/foundation/devkit/arm/v5t_le/lib
  --libexecdir=/opt/montavista/foundation/devkit/arm/v5t_le/libexec
  --localstatedir=/opt/montavista/foundation/devkit/arm/v5t_le/var
  --sharedstatedir=/opt/montavista/foundation/devkit/arm/v5t_le/share
  --mandir=/opt/montavista/foundation/devkit/arm/v5t_le/man
  --infodir=/opt/montavista/foundation/devkit/arm/v5t_le/info
  --build=i686-pc-linux-gnu
  --program-transform-name=s,arm_v5t_le-,
  --enable-cross
  --enable-poison-system-directories
  --with-sysroot=/opt/montavista/foundation/devkit/arm/v5t_le/target
  --with-build-sysroot=/opt/montavista/foundation/devkit/arm/v5t_le/target
  --with-build-time-tools=/opt/montavista/foundation/devkit/arm/v5t_le/bin
  --enable-shared
  --enable-languages=c,c++
  --enable-cxa_atexit
  --enable-c99
  --enable-long-long
  --enable-threads=posix
  --with-gxx-include-dir=/opt/montavista/foundation/devkit/arm/v5t_le/lib/gcc/armv5t-le-montavista-linux-gnueabi/4.2.0/../../../../target/usr/include/c++/4.2.0
  --disable-libmudflap
  --disable-libssp
  --disable-libgomp
  --with-gnu-as
  --with-gnu-ld
  --enable-symvers=gnu
  --enable-checking=release
  --with-numa-policy=yes
  --disable-mu
  --with-cpu=arm10tdmi
  --with-interwork
  --with-arch=armv5t
  --with-tune=arm10tdmi
  --libexecdir=/opt/montavista/foundation/devkit/arm/v5t_le/lib
  --with-bugurl=http://www.mvista.com/support
  --with-version-suffix='MontaVista 4.2.0-16.0.32.0801914 2008-08-30'
Thread model: posix
gcc version 4.2.0 (MontaVista 4.2.0-16.0.32.0801914 2008-08-30)
```

图 2 交叉编译环境检测

### 步骤 3：小程序测试

步骤 2 中搭建了交叉编译环境，接下来交叉编译一个小程序“hello world”，挂载到实验箱上，新建 vim 文件（vim helloworld.c），输入以下内容：

```
#include<stdio.h>

int main()
{
printf("hello world \n");

return 0;

}
```

点击esc 键退出输入，输入:wq!退出文档。

### （2）交叉编译

生成二进制可执行文件 `helloworld`，其中 `helloworld.c` 为交叉编译的程序，`-o` 表示输出，`helloworld` 表示生成的二进制可执行文件名：

```
# arm_v5t_le-gcc helloworld.c -o helloworld
```

在 PC 上运行生成的二进制可执行文件 `helloworld`：

```
#./helloworld
```

提示 `-bash:./ helloworld: cannot execute binary file`，即不能执行该二进制可执行文件。（这与下面在实验箱上的运行作对比。）

### （3）正确连接实验箱和 PC 机

将 PC 机与开发板通过 USB 转串口线正确连接，将开发板的电源线、网线正确连接，插上电源。

### （4）登陆 PuTTY 的 COM 口端

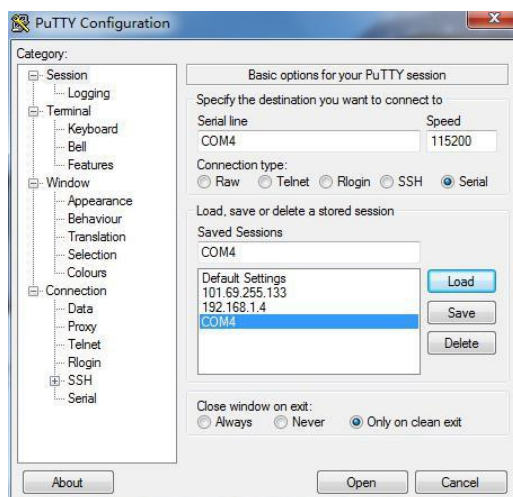


图 3 PuTTY 配置界面

打开设备开关，在 PuTTY 的 COM 口端进行操作，当实验板有打印消息时，按 `enter` 键使系统停止启动，输入启动参数：

保存启动参数：

```
#saveenv
```

### （5）通过实验箱运行交叉编译生成的可执行文件

在 PuTTY 的服务器端进行操作，将生成的二进制可执行文件 `helloworld` 由它所在的目录复制到文件系统所在目录的 `/opt/dm365` 下（eg:服务器文件系统所在目录为 `/home/shiyan/filesys_test`，就复制到 `/home/stX/filesys_test /opt/dm365` 目录）；

（虚拟机文件系统所在目录为 `/home/shiyan/share/filesys_test/`，就复制到 `/home/shiyan/share/filesys_test/opt/dm365` 目录）：

服务器：#sudo cp helloworld /home/stX/filesys\_test/opt/dm365

虚拟机：#sudo cp helloworld /home/shiyan/share/filesys\_test/opt/dm365

在 putty 端口端操作：

进入可执行文件所在目录/opt/dm365

```
#cd /opt/dm365
```

运行二进制可执行文件 helloworld:

```
#helloworld
```

显示结果如下所示：

```
hello world!
```