第二章 交叉开发

2.1 为什么要使用交叉编译

1、首先我们知道计算机只识别二进制，所以我们编写的程序如果要在计算机上运行是需要进行编译的，编译就是编译器（例如gcc、g++）将程序翻译成二进制的过程

2、在ubuntu系统下使用gcc或者g++编译器生成的可执行程序只能运行在ubuntu系统下，并且只能运行在x86架构的CPU上。

我们可以使用file 命令查看编译生成的可执行文件。

```
msb@msb:/mnt/hgfs/untitled3$ file main
main: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]
=89561ab9172374aef7b91679736c286e32cd3269, for GNU/Linux 3.2.0, not stripped
```

因为不同的CPU所支持的指令集不同，所以在ubuntu系统下使用gcc 或者 g++编译生成的可执行程序不能运行在ARM平台上。

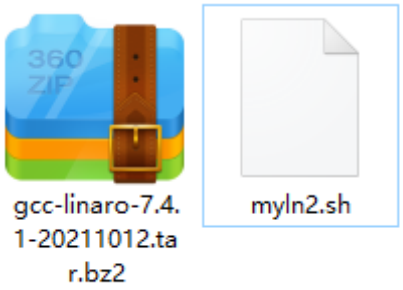
3、如果程序需要运行在ARM平台上，我们需要在ubuntu系统下使用专门的编译器对源代码进行编译，然后再将编译好的可执行程序拷贝到ARM平台上运行，这个过程我们就叫做交叉编译。

4、问题：直接在ARM平台上使用专用的编译器编译这样是不是也可以呢？还省去了需要将程序从PC机器上拷贝到开发板上？

> 答：理论上是可以的！！但是因为ARM平台的资源（性能）与PC机相差太多，如果在ARM平台上运行编程程序速度较慢

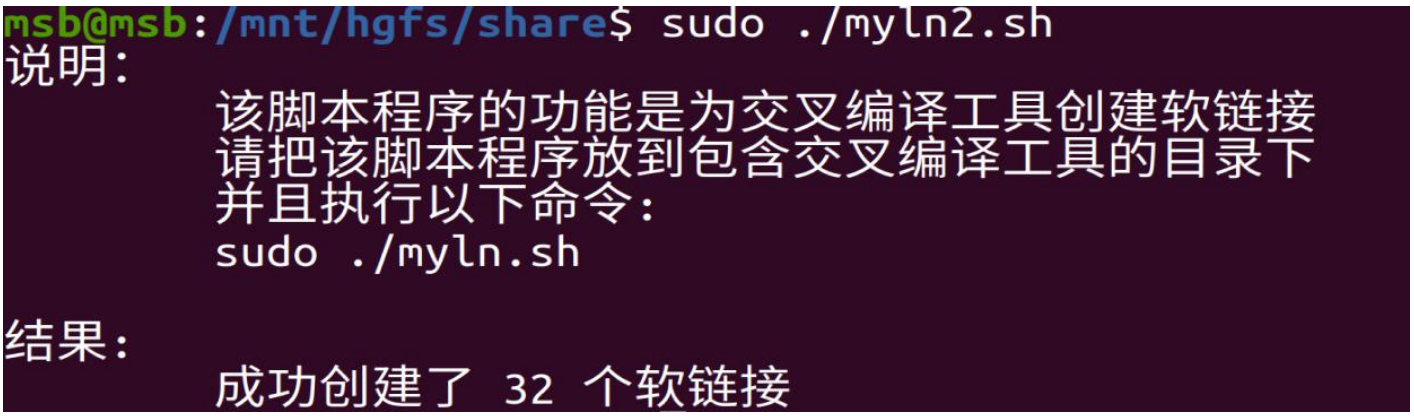
2.2 交叉编译器的安装

1、将“编译器/gcc-linaro-7.4.1-20211012.tar.bz2”和 “编译器/myln2.sh”这两个文件拷贝到vmware虚拟机的共享目录下



2、在虚拟机中进入到共享目录下，然后打开终端

3、在终端输入运行myln2.sh脚本文件，执行命令：sudo ./myln2.sh。注意：虚拟机密码为123456



4、经过一段时间后交叉编译器就安装完成了。

5、重启虚拟机

6、虚拟机重启后测试交叉编译器是否成功安装：在终端输入arm-linux-gcc，如果出现如下提示信息则说明安装成功

2.3 交叉编译器的使用

我们可以编写一个简单的程序hello.c

然后使用交叉编译器进行编译：

> arm-linux-gcc hello.c -o hello

我们使用file命令查看hello文件：file hello

```
msb@msb:~$ file hello
hello: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib
/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]=4a811eb31ef2b877dee1edc698f89ba4e6fcbf4f,
with debug_info, not stripped
```

发现hello为支持ARM体系结构的可执行程序。

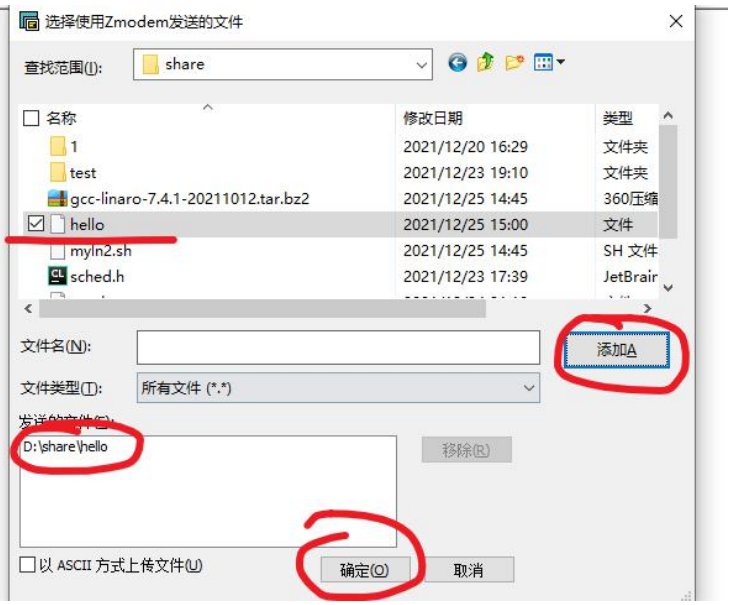
下面我们就可以将编译好的hello文件下载到开发板上运行了:

1、将hello拷贝到虚拟机的共享目录下：cp hello /mnt/hgfs/share

```
msb@msb:~$ cp hello /mnt/hgfs/share/
```

2、在开发板中输入命令：rz -e 命令然后 选择共享目录下的hello文件进行下载

```
root@/ # cd /home/
root@/home # rz -e
rz waiting to receive.
¿ªÊ¼ zmodem ´«Êä¡£  °´ Ctrl+C È¡Ïû¡£
```



3、为hello添加可执行权限：chmod +x hello

```
root@/home # chmod +x hello
root@/home #
```

4、运行hello程序：./hello

```
root@/home # ./hello
hello
root@/home #
```

2.4 NFS网络文件系统搭建

2.4.1 NSF服务的安装

NFS 是 Network FileSystem 的简称，可以让不同的主机通过网络访问远端的 NFS 服务器共享出来的文件。这样，将主机当作 NFS 服务器，我们就可以在开发板上通过网络访问主机的文件。在嵌入式中使用 NFS 使得应用程序的开发十分方便，如在 Ubuntu 环境搭建时已安装，可跳过此节。

具体的使用方法如下：

1、安装服务

> sudo apt-get install nfs-kernel-server

2、在虚拟机/home/msb目录下创建目录 nfs_share

> mkdir /home/msb/nfs_share

3、创建共享目录，修改 /etc/exports 文件末尾

> sudo vi /etc/exports
>
> 或者
>
> sudo gedit /etc/exports

在文件末尾加入下面一行：

> /home/msb/nfs_share *(rw,sync,no_root_squash)

这样，就将主机的/home/msb/nfs_share目录当作了共享目录，也可自己配置共享目录。

4、启动 NFS 服务器,重启 nfs-kernel-server 服务：

> sudo /etc/init.d/nfs-kernel-server restart

2.4.2 挂载 NFS 服务器

我们可以使用之前配置好的 NFS 服务器，在开发板上挂载主机的 NFS 服务器后，就可以开发板上操作主机上的文件，如复制文件、运行程序等。这种方式十分方便于调试，具体实现方法如下：

1、将ubuntu虚拟机的IP地址设置为静态地址：192.168.1.76

2、确保通过网线将开发板与 PC 机连接好，并在 PC 机上开启了 NFS 服务

3、设置开发板的 IP 与 PC 机的 IP 在同一网段，如:

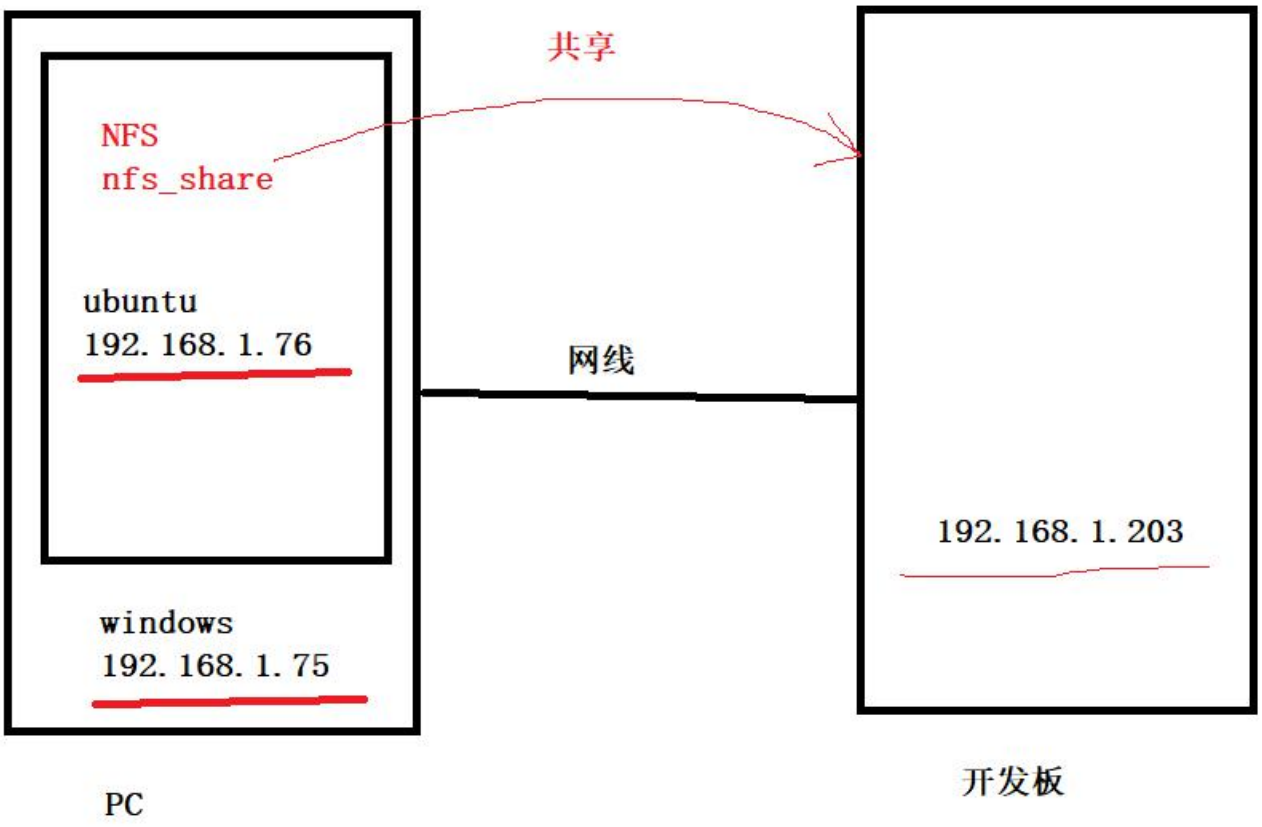
> PC 机 IP ： 192.168.1.75
>
> 目标板 IP：192.168.1.203
>
> Network Marsk: 255.255.255.0
>
> Broadcast IP: 192.168.1.255

4、测试网络连接

在开发板上 ping 主机，在超级终端运行如下命令：

> #ping 192.168.1.75

在主机上以同样的方式 ping 开发板，如果主机和开发板可以互相 ping 通，则说明网络连接正常。



5、挂载主机 NFS 服务器(在开发板上操作)

在超级终端输入以下命令：

> mount -o nolock 192.168.1.76:/home/msb/nfs_share /mnt/
>
> cd /mnt/

挂载正确后，可以在开发板的/mnt 目录下看到主机的根目录