**Linux 简单字符驱动实验**

**实验目的**

-掌握嵌入式linux下字符驱动程序的编译和运行

-掌握嵌入式linux字符驱动模块的动态加载，卸载；应用程序与驱动程序的通信调用机制

**实验内容**

1）编译内核

2）编译驱动以及应用程序

**实验步骤**

**编译内核**

1)我们系统的内核就在 /opt/cross-compiler/kernel-embv210 目录下。

可以打开一个终端，到那个目录直接桥这些名令就可以。

$ cd /opt/cross-compiler/kernel-embv210

$ make distclean

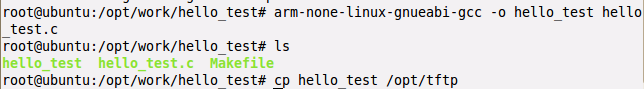
$ cp embv210.config .config

$ make zImage

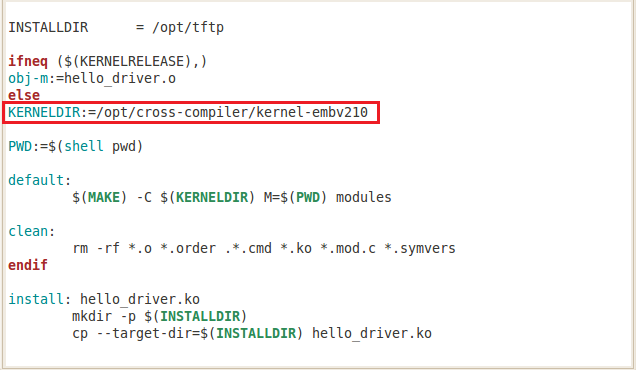
注意zImage 的大小写！

**编译驱动和应用程序**

1）把所提供的代码拷贝到 /opt/work/ 目录下。进入到hello\_test 目录。用 arm-none-linux-gnueabi-gcc 来编译本程序。然后把结果拷贝到 /opt/tftp 目录。如下图。

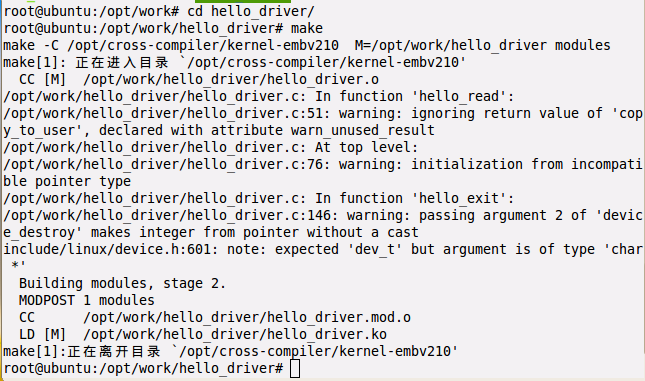


2）为了编译好hello\_driver,就是我们的驱动程序。我们将会使用Makefile的方法。先进入 /opt/work/hello\_driver, 然后用gedit 编辑器来查看Makefile 文件，保证它所指向的内核路径是否正确。内核路径就是 /opt/cross-compiler/kernel-embv210。编辑完之后保存并关闭



INSTALLDIR 就是他make install的时候考结果的目录。在本情况他会考到 /opt/tftp

2)在同样的目录，编译我们的程序。用 make 命令。会执行Makefile 文件中的default 部分。



3）现在用 make install 命令他会考到安装目录。因为我们不应该在本系统运行所以就该到 /opt/tftp 目录。它将会执行Makefile 文件的 install 部分。



结果已经考到 /opt/tftp 目录下。

4）现在桥make clean 来删除所有后来不用的文件。它会执行Makefile 文件中的clean 部分。

5)现在，确认宿主机和目标机是连好的，目标机是开机的，并在宿主机终端用minicom -s 命令打开串口软件。

6）用tftp服务把hello\_test 和hello\_driver.ko 文件考到目标机上去。

Tftp 就是 trivial file transfer protocol，也是一个文件传输协议。本协议没有scp协议安全，他不提供用户验证方案，所以用的时候最好在一个防火墙后面。它的用法如下。

tftp -[操作] [目标机ip地址] -r ./[目标机上文件名] -l ./[本机上文件名]

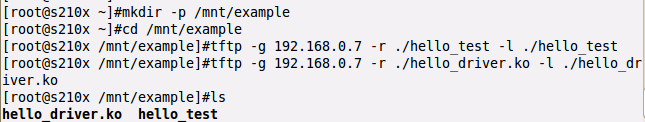
操作： 有两种操作，其中一次只能用一个。

g表示get

p表示put。

r 表示 remote, l 是 local的意思。

下面用本方案来传输我们的文件。从我们的目标机上。



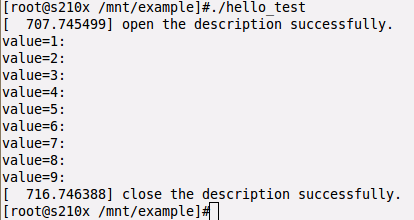
7）把文件的权限改一下



8）用insmod命令加载我们的驱动 hello\_driver.ko



9）执行hello\_test，结果如下图。



本结果是一下的工作方式。

read IOCTL

write IOCTL

应用程序

hello\_test

驱动程序

hello\_driver

设备

10)现在可以查看并卸载已经加载的驱动。

