# 实验 11：Linux 交叉编译平台实验

## 一、实验目的

1. 理解交叉编译的原理和概念。
2. 掌握在 Linux 下建立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. 通过路由器将实验箱和 PC 机连接。
3. 在 Linux 操作系统的服务器上安装交叉编译环境，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；AMR9 系统教学实验系统；串口线；网线；服务器。
2. 软件：PC 机操作系统（Windows XP）；Linux 操作系统。

## 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。

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。

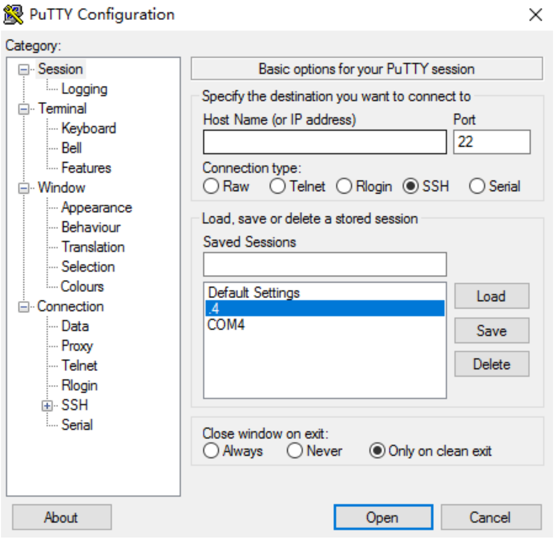
NFS 在文件传送或信息传送过程中依赖于 RPC 协议。RPC(Remote Procedure Call) 远程过程调用能使客户端执行其他系统中程序的一种机制。NFS 本身是没有提供信息传输的协议和功能的，但 NFS 却能让我们通过网络进行资料的分享，这是因为 NFS 使用了一些其它的传输协议。而这些传输协议用到了 RPC 的功能，也可以说 NFS 本身就是使用 RPC 的一个程序，或者说 NFS 是一个 RPC SERVER。所以只要用到 NFS 的地方都要启动 RPC 服务，不论是 NFS SERVER 或者 NFS CLIENT，这样 SERVER 和 CLIENT 才能通过 RPC 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

**步骤 1：** 硬件连接

(1) 连接好实验箱的网线、串口线和电源。

(2) 首先通过 putty 软件使用 ssh 通信方式登录到服务器，如下图所示（在 Hostname 栏输入服务器的 ip 地址）：



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
.4
COM4

Close window on exit: ○ Always ○ Never ● Only on clean exit

图 1 打开 putty 连接

(3) 查看串口号，右键我的电脑--->选择管理--->设备管理器--->端口，查看实验箱的串口号。如下图 2 所示：

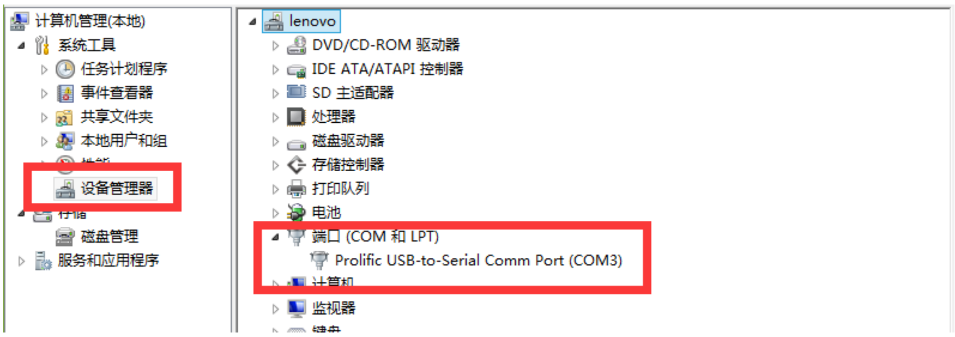
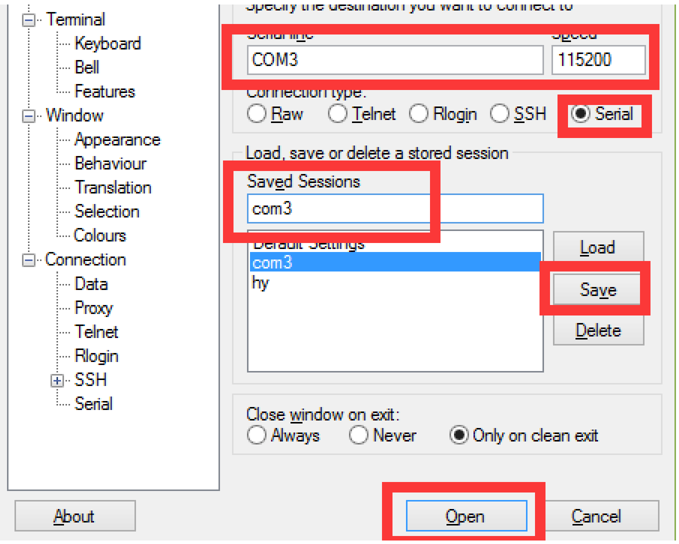


图 2 端口号查询

(4) 在 putty 软件端口栏输入(3)中查询到的串口，设置波特率为 115200，连接实验箱，如下图 3 所示：



Specify the destination you want to connect to

Serial line: COM3 Speed: 115200

Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: com3

Default Settings
com3
hy

Close window on exit:
○ Always ○ Never ● Only on clean exit

图 3 putty 串口连接配置

(5) 点击 putty 软件中的 Open 按键，进入连接页面，打开实验箱开关，在 5s 内，点击 Entet 键，然后输入挂载参数，再次点击 Enter 键，输入 boot 命令，按 Enter 键，开始进行挂载。具体信息如下所示：

```
DM365 EVM :>setenv    bootargs 'mem=110M console=ttyS0,115200n8 root=/dev/nfs rw
nfsroot=192.168.1.18:/home/shiyan/filesys_clwxl
ip=192.168.1.42:192.168.1.18:192.168.1.1:255.255.255.0::eth0:off eth=00:40:01:C1:56:78
video=davincifb:vid0=OFF:vid1=OFF:osd0=640x480x16,600K:osd1=0x0x0,0K dm365_imp.oper_mode=0
davinci_capture.device_type=1 davinci_enc_mngr.ch0_output=LCD'
DM365 EVM :>boot

Loading from NAND 1GiB 3,3V 8-bit, offset 0x400000
   Image Name:    Linux-2.6.18-plc_pro500-davinci_
   Image Type:    ARM Linux Kernel Image (uncompressed)
   Data Size:     1996144 Bytes =    1.9 MB
   Load Address: 80008000
   Entry Point:    80008000
## Booting kernel from Legacy Image at 80700000 ...
   Image Name:    Linux-2.6.18-plc_pro500-davinci_
   Image Type:    ARM Linux Kernel Image (uncompressed)
   Data Size:     1996144 Bytes =   1.9 MB
   Load Address: 80008000
   Entry Point:   80008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux.............................................................................................................................
done, booting the kernel.
[     0.000000] Linux version 2.6.18-plc_pro500-davinci_evm-arm_v5t_le-gfaa0b471-dirty
(zcy@punuo-Lenovo) (gcc version 4.2.0 (MontaVista 4.2.0-16.0.32.0801914 2008-08-30)) #1 PREEMPT
Mon Jun 27 15:31:35 CST 2016
[     0.000000] CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177
[     0.000000] Machine: DaVinci DM365 EVM
[     0.000000] Memory policy: ECC disabled, Data cache writeback
[     0.000000] DaVinci DM0365 variant 0x8
[     0.000000] PLL0: fixedrate: 24000000, commonrate: 121500000, vpssrate: 243000000
[     0.000000] PLL0: vencrate_sd: 27000000, ddrrate: 243000000 mmcsdrate: 121500000
[     0.000000] PLL1: armrate: 297000000, voicerate: 20482758, vencrate_hd: 74250000
[     0.000000] CPU0: D VIVT write-back cache
[     0.000000] CPU0: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
[     0.000000] CPU0: D cache: 8192 bytes, associativity 4, 32 byte lines, 64 sets
[     0.000000] Built 1 zonelists.   Total pages: 28160
[     0.000000] Kernel command line: mem=110M console=ttyS0,115200n8 root=/dev/nfs rw
nfsroot=192.168.1.18:/home/shiyan/filesys_clwxl
ip=192.168.1.42:192.168.1.18:192.168.1.1:255.255.255.0::eth0:off eth=00:40:01:C1:56:78
video=davincifb:vid0=OFF:vid1=OFF:osd0=640x480x16,600K:osd1=0x0x0,0K dm365_imp.oper_mode=0
davinci_capture.device_type=1 davinci_enc_mngr.ch0_output=LCD
[     0.000000] TI DaVinci EMAC: kernel boot params Ethernet address: 00:40:01:C1:56:78
[     0.000000] PID hash table entries: 512 (order: 9, 2048 bytes)
[     0.000000] Clock event device timer0_0 configured with caps set: 07
[     0.000000] Console: colour dummy device 80x30
[     0.000000] Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
[     0.000000] Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
[     0.010000] Memory: 110MB = 110MB total
[     0.010000] Memory: 107136KB available (3165K code, 692K data, 492K init)
[     0.220000] Security Framework v1.0.0 initialized
[     0.220000] Capability LSM initialized
[     0.220000] Mount-cache hash table entries: 512
[     0.220000] CPU: Testing write buffer coherency: ok
[     0.220000] NET: Registered protocol family 16
[     0.240000] DaVinci: 104 gpio irqs
[     0.250000] MUX: initialized GPIO20
[     2.250000] MUX: initialized I2C_SCL
[     2.250000] Pin GPIO20 already used for I2C_SCL.
[    2.250000] MUX: initialized GPIO30
[    2.250000] MUX: initialized GPIO31
[    2.250000] MUX: initialized GPIO32
[    2.250000] MUX: initialized GPIO33
[    2.250000] MUX: initialized GPIO35
[    2.250000] MUX: initialized GPIO37
[    2.250000] MUX: initialized GPIO38
[    2.250000] MUX: initialized GPIO39
[    2.250000] MUX: initialized GPIO40
[    2.250000] MUX: initialized GPIO41
[    2.250000] MUX: initialized GPIO51
[    2.250000] MUX: initialized GPIO55
[    2.250000] MUX: initialized GPIO58
[    2.250000] MUX: initialized GPIO80
[    2.250000] MUX: initialized GPIO93
[    2.250000] MUX: initialized GPIO28
[    2.250000] MUX: initialized GPIO29
[    4.450000] MUX: initialized UART1_RXD
[    4.450000] MUX: initialized UART1_TXD
[    4.450000] DM365 IPIPE initialized in Continuous mode
[    4.460000] Generic PHY: Registered new driver
[    4.460000] ch0 default output "LCD", mode "NTSC"
[    4.460000] VPBE Encoder Initialized
[    4.460000] Invalid id...
[    4.460000] Set output or mode failed, reset to encoder default...
[    4.460000] MUX: initialized VOUT_FIELD_G81
[    4.460000] LogicPD encoder initialized
[    4.460000] Avnetlcd encoder initialized
[    4.460000] dm365_afew_hw_init
[    4.460000] SCSI subsystem initialized
[    4.460000] usbcore: registered new driver usbfs
[    4.460000] usbcore: registered new driver hub
[    4.470000] NET: Registered protocol family 2
[    4.560000] IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
[    4.560000] TCP established hash table entries: 4096 (order: 2, 16384 bytes)
[    4.560000] TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
[    4.560000] TCP: Hash tables configured (established 4096 bind 2048)
[    4.560000] TCP reno registered
[    4.590000] yaffs Jun 27 2016 15:33:25 Installing.
[    4.600000] Initializing Cryptographic API
[    4.600000] io scheduler noop registered
[    4.600000] io scheduler anticipatory registered (default)
[    4.600000] io scheduler deadline registered
[    4.600000] io scheduler cfq registered
[    4.620000] Console: switching to colour frame buffer device 80x30
[    4.660000] davincifb davincifb.0: dm_osd0_fb: 640x480x16@0,0 with framebuffer size 600KB
[    4.670000] davincifb davincifb.0: dm_vid0_fb: 0x0x16@0,0 with framebuffer size 900KB
[    4.670000] davincifb davincifb.0: dm_vid1_fb: 0x0x16@0,0 with framebuffer size 900KB
[    4.720000] TI Davinci ADC v1.0
[    4.730000] DAVINCI-WDT: DaVinci Watchdog Timer: heartbeat 60 sec
[    4.730000] imp serializer initialized
[    4.730000] davinci_previewer initialized
[    4.730000] davinci_resizer initialized
[    4.730000] dm365_gpio initialized
[    4.730000] Serial: 8250/16550 driver $Revision: 1.90 $ 2 ports, IRQ sharing disabled
[    4.730000] serial8250.0: ttyS0 at MMIO map 0x1c20000 mem 0xfbc20000 (irq = 40) is a 16550A
[    4.750000] serial8250.0: ttyS1 at MMIO map 0x1d06000 mem 0xfbd06000 (irq = 41) is a 16550A
[    4.760000] RAMDISK driver initialized: 1 RAM disks of 32768K size 1024 blocksize
[    4.770000] PPP generic driver version 2.4.2
[    4.770000] PPP Deflate Compression module registered
[    4.780000] PPP BSD Compression module registered
[    4.790000] Davinci EMAC MII Bus: probed
[    4.790000] sjwedit --> EMAC: 00:40:01:C1:56:78.
[    4.800000] MAC address is 00:40:01:C1:56:78
[    4.800000] TI DaVinci EMAC Linux version updated 4.0
[    4.810000] Linux video capture interface: v2.00
[    4.820000] vpfe_init
[    4.820000] Pin VIN_CAM_WEN already used for GPIO93.
[    4.820000] starting ccdc_reset...<7>
[    4.830000] End of ccdc_reset...<5>vpfe_probe
[    4.830000] TVP5150 : nummber of channels = 1
[    4.840000] vpfe ccdc capture vpfe ccdc capture.1: vpif_register_decoder: decoder = TVP5150
[    4.850000] Trying to register davinci display video device.
[    4.860000] layer=c07eb600,layer->video_dev=c07eb760
[    4.860000] Trying to register davinci display video device.
[    4.870000] layer=c07eb400,layer->video_dev=c07eb560
[    4.870000] davinci_init:DaVinci V4L2 Display Driver V1.0 loaded
[    4.880000] vpfe ccdc capture vpfe ccdc capture.1: vpif_register_decoder: decoder = TVP7002
[    4.890000] af major#: 251, minor# 0
[    4.890000] AF Driver initialized
[    4.900000] aew major#: 250, minor# 0
[    4.900000] AEW Driver initialized
[    4.910000] i2c /dev entries driver
[    4.920000] nand_davinci nand_davinci.0: Using 4-bit hardware ECC
[    4.920000] NAND device: Manufacturer ID: 0xec, Chip ID: 0xd3 (Samsung NAND 1GiB 3,3V 8-bit)
[    4.940000] Creating 5 MTD partitions on "nand_davinci.0":
[    4.940000] 0x00000000-0x00780000 : "bootloader"
[    4.950000] 0x00780000-0x00800000 : "params"
[    4.950000] 0x00800000-0x00c00000 : "kernel"
[    4.960000] 0x00c00000-0x20c00000 : "filesystem"
[    4.970000] 0x20c00000-0x40000000 : "backup_filesys"
[    4.980000] nand_davinci nand_davinci.0: hardware revision: 2.3
[    4.990000] Pin SPI1_SCLK already used for GPIO28.
[    5.000000] dm_spi.0: davinci SPI Controller driver at 0xc7008000 (irq = 42) use_dma=0
[    5.000000] Initializing USB Mass Storage driver...
[    5.010000] usbcore: registered new driver usb-storage
[    5.010000] USB Mass Storage support registered.
[    5.020000] usbcore: registered new driver usbhid
[    5.030000] drivers/usb/input/hid-core.c: v2.6:USB HID core driver
[    5.030000] usbcore: registered new driver usbserial
[    5.040000] drivers/usb/serial/usb-serial.c: USB Serial support registered for generic
[    5.050000] usbcore: registered new driver usbserial_generic
[    5.050000] drivers/usb/serial/usb-serial.c: USB Serial Driver core
[    5.060000] drivers/usb/serial/usb-serial.c: USB Serial support registered for GSM modem (1-port)
[    5.070000] usbcore: registered new driver option
[    5.070000] drivers/usb/serial/option.c: USB Driver for GSM modems: v0.7.1
[    5.080000] drivers/usb/serial/usb-serial.c: USB Serial support registered for pl2303
[    5.090000] usbcore: registered new driver pl2303
[    5.100000] drivers/usb/serial/pl2303.c: Prolific PL2303 USB to serial adaptor driver
[    5.100000] musb_hdrc: version 6.0, cppi-dma, host, debug=0
[    5.130000] musb_hdrc musb_hdrc: No DMA interrupt line
[    5.130000] musb_hdrc: USB Host mode controller at c700a000 using DMA, IRQ 12
[    5.140000] musb_hdrc musb_hdrc: MUSB HDRC host driver
[    5.140000] musb_hdrc musb_hdrc: new USB bus registered, assigned bus number 1
[    5.150000] usb usb1: configuration #1 chosen from 1 choice
[    5.160000] hub 1-0:1.0: USB hub found
[    5.160000] hub 1-0:1.0: 1 port detected
[    5.280000] DaVinci DM365 Keypad Driver
[    5.280000] MUX: initialized KEYPAD
[    5.290000] input: dm365_keypad as /class/input/input0
[    5.300000] year:2000,mon:1,day:0,hour:80,min:0,sec:0
[    5.310000] davinci-mmc davinci-mmc.0: Supporting 4-bit mode
[    5.310000] davinci-mmc davinci-mmc.0: Using DMA mode
[    5.320000] Advanced Linux Sound Architecture Driver Version 1.0.12rc1 (Thu Jun 22 13:55:50 2006
UTC).
[    5.330000] ASoC version 0.13.1
[    5.330000] AIC3X Audio Codec 0.2
[    5.340000] asoc: aic3x <-> davinci-i2s mapping ok
[    5.440000] ALSA device list:
[    5.450000]   #0: DaVinci DM365 EVM (aic3x)
[    5.450000] IPv4 over IPv4 tunneling driver
[    5.460000] GRE over IPv4 tunneling driver
[    5.460000] TCP bic registered
[    5.470000] NET: Registered protocol family 1
[    5.470000] NET: Registered protocol family 17
[    5.620000] usb 1-1: new high speed USB device using musb_hdrc and address 2
[    5.760000] usb 1-1: configuration #1 chosen from 1 choice
[    5.760000] hub 1-1:1.0: USB hub found
[    5.770000] hub 1-1:1.0: 4 ports detected
[    5.930000] Bridge firewalling registered
[    5.930000] 802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
[    5.940000] All bugs added by David S. Miller <davem@redhat.com>
[    5.940000] drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
[    5.950000] Time: timer0_1 clocksource has been installed.
[       5.960000] Clock event device timer0_0 configured with caps set: 08
[       5.960000] Switched to high resolution mode on CPU 0
[       6.120000] usb 1-1.4: new full speed USB device using musb_hdrc and address 3
[       6.220000] usb 1-1.4: configuration #1 chosen from 1 choice
[       6.220000] pl2303 1-1.4:1.0: pl2303 converter detected
[       6.240000] usb 1-1.4: pl2303 converter now attached to ttyUSB0
[       7.480000] IP-Config: Complete:
[       7.480000]              device=eth0, addr=192.168.1.42, mask=255.255.255.0, gw=192.168.1.1,
[       7.490000]            host=192.168.1.42, domain=, nis-domain=(none),
[       7.490000]            bootserver=192.168.1.18, rootserver=192.168.1.18, rootpath=
[       7.500000] Looking up port of RPC 100003/2 on 192.168.1.18
[       9.520000] Looking up port of RPC 100005/1 on 192.168.1.18
[       9.540000] VFS: Mounted root (nfs filesystem).
[       9.540000] Freeing init memory: 492K
[      21.060000] usb 1-1.1: new high speed USB device using musb_hdrc and address 4
[      21.160000] usb 1-1.1: configuration #1 chosen from 1 choice

INIT: Entering runlevel: 3

Starting internet superserver: inetd.
mount: special device /dev/mmcblk0p1 does not exist
open wifi ra0
/********Start RTC********/
[      25.520000] rtusb init rt2870 --->
[      25.530000] usbcore: registered new driver rt2870
[      25.610000] minor is 63
[      25.610000] #################
[      25.610000] [egalax_i2c]:    /proc/egalax_dbg created
[      25.620000] [egalax_i2c]:    Driver init done!
[      25.630000] egalax_i2c_detect
[      25.630000] i2c_adapter->name=DaVinci I2C adapter
[      25.640000] #####################
[      25.640000] new_client->name=egalax_i2c
[      25.640000] egalax_i2c_probe with name = egalax_i2c, addr = 0x4
[      25.670000] [egalax_i2c]:    Start probe
[      25.680000] input: eGalax_Touch_Screen as /class/input/input1
[      25.690000] [egalax_i2c]:    Register input device done
[      25.700000] No IRQF_TRIGGER set_type function for IRQ 44 (AINTC)
[      25.700000] [egalax_i2c]:    INT wakeup touch controller done
[      25.720000] [egalax_i2c]:    I2C probe done
[      25.780000] Register dht11 driver
[      25.850000] Register sr04 driver
[      25.900000] ov5640_i2c: Unknown symbol scanmode
insmod: cannot insert '/modules/ov5640_i2c.ko': Unknown symbol in module (-1): No such file or directory
[      25.960000] year:2000,mon:1,day:0,hour:80,min:0,sec:0
[      26.200000] [egalax_i2c]:    INT with irq:44
[      26.210000] [egalax_i2c]:    egalax_i2c_wq run
[      26.220000] [egalax_i2c]:    I2C get vendor command packet
[    26.220000] [egalax_i2c]:   Get Device type=1
[    26.230000] [egalax_i2c]:   I2C get vendor command packet
[    26.240000] [egalax_i2c]:   I2C get vendor command packet
osd0: xres 640 yres 480 xres_v 640 yres_v 480 line_length1280


MontaVista(R) Linux(R) Professional Edition 5.0.0 (0801921)



zjut login: [    27.200000] [egalax_i2c]:   Close egalax_i2c_wq_loopback work
[    27.210000] [egalax_i2c]:   INT with irq:44
[    27.220000] [egalax_i2c]:   egalax_i2c_wq run
[    27.230000] [egalax_i2c]:   I2C get vendor command packet
[    28.960000] CMEMK module: built on Apr   7 2014 at 10:55:46
[    28.980000]    Reference Linux version 2.6.18
[    28.980000]    File
/home/plc/dvsdk/linuxutils_2_24_02/packages/ti/sdo/linuxutils/cmem/src/module/cmemk.c
[    29.110000] ioremap_nocache(0x87000000, 16777216)=0xcb000000
[    29.110000] allocated heap buffer 0xcb000000 of size 0x3f7000
[    29.130000] cmem initialized 9 pools between 0x87000000 and 0x88000000
[    29.130000] CMEM Range Overlaps Kernel Physical - allowing overlap
[    29.130000] CMEM phys_start (0x1000) overlaps kernel (0x80000000 -> 0x86e00000)
[    29.150000] ioremap_nocache(0x1000, 28672)=0xc7010000
[    29.150000] no remaining memory for heap, no heap created for memory block 1
[    29.160000] cmem initialized 1 pools between 0x1000 and 0x8000
[    29.240000] IRQK module: built on Apr   7 2014 at 11:01:18
[    29.240000]    Reference Linux version 2.6.18
[    29.250000]    File
/home/plc/dvsdk/linuxutils_2_24_02/packages/ti/sdo/linuxutils/irq/src/module/irqk.c
[    29.270000] irqk initialized
[    29.340000] EDMAK module: built on Apr   7 2014 at 10:58:36
[    29.360000]    Reference Linux version 2.6.18
[    29.370000]    File
/home/plc/dvsdk/linuxutils_2_24_02/packages/ti/sdo/linuxutils/edma/src/module/edmak.c
WCDMA Autodialog
[    34.480000] Starting ccdc_config_ycbcr...<7>
[    34.480000] starting ccdc_reset...<7>
[    34.490000] End of ccdc_reset...<7>
[    34.490000] Starting ccdc_setwin...<7>ipipe_set_resizer, resizer - A enabled
[    34.610000] DavinciDisplay DavinciDisplay.1: Before finishing with S_FMT:
[    34.610000] layer.pix_fmt.bytesperline = 640,
[    34.610000]   layer.pix_fmt.width = 640,
[    34.610000]   layer.pix_fmt.height = 480,
[    34.610000]   layer.pix_fmt.sizeimage =460800
[    34.640000] DavinciDisplay DavinciDisplay.1: pixfmt->width = 640,
[    34.640000]   layer->layer_info.config.line_length= 640
KeypadDriverPlugin::create#####################: optkeypad
```

```
keyboard input device ( "/dev/input/event0" ) is opened.
id= "0"
msqid= 0


MontaVista(R) Linux(R) Professional Edition 5.0.0 (0801921)
```

(6) 点击 Enter，输入用户名 root 登录实验箱，如下所示：

```
zjut login: root


Welcome to MontaVista(R) Linux(R) Professional Edition 5.0.0 (0801921).

login[737]: root login on 'console'

/******Set QT environment********/

[root@zjut ~]#
```

**步骤 2**：搭建交叉编译环境

在服务器窗口，创建一个文件夹 mv_pro_5.0，进入文件夹 mv_pro_5.0，将/home/shiyan/arm_v5 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
#cp    /home/shiyan/arm_v5/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
```

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/shiyan/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/bin
```

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` ，打印如下的版本信息，表示交叉编译环境搭建成功。

```
使用内建 specs。
目标：armv5tl-montavista-linux-gnueabi
配置为：../configure --host=i686-pc-linux-gnu --build=i686-pc-linux-gnu
--target=armv5tl-montavista-linux-gnueabi --prefix=/opt/montavista/foundation/devkit/arm/v5t_le
--exec-prefix=/opt/montavista/foundation/devkit/arm/v5t_le
```

```
--bindir=/opt/montavista/foundation/devkit/arm/v5t_le/bin
--sbindir=/opt/montavista/foundation/devkit/arm/v5t_le/sbin
--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/armv5tl-montavista-linux-gnu
eabi/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-multilib --enable-clocale=gnu --with-float=soft --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-versuffix='MontaVista 4.2.0-16.0.32.0801914
2008-08-30'
线程模型：posix
gcc  版本  4.2.0 (MontaVista 4.2.0-16.0.32.0801914 2008-08-30)
```

图 7-5　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，然后运行（helloworld.c 文件在“实验电子材料/Linux 嵌入式实验/交叉编译实验”目录下）。

helloworld.c 程序代码如下：

```
#include<stdio.h>
int main()
{
printf("hello world !\n");
return 0;
}
```

（1）查看 helloworld.c 程序

进入该程序 helloworld.c 的目录下：

进入目录：

```
#cd    /home/shiyan/
```

查看文件

```
#ls
helloworld.c
```

（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）通过实验箱运行交叉编译生成的可执行文件

完成实验箱的登录挂载以后

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:文件系统所在目录为/home/shiyan/filesys_XXX，就复制到/home/shiyan/filesys_XXX /opt/dm365 目录）：

```
#sudo    cp    helloworld    /home/shiyan/filesys_XXX /opt/dm365
```

在 PuTTY 的 COM 口端进行操作：

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!
```

**六、总结**

以上通过一个简单的 hello.c 程序，可以清楚地看到，交叉编译后的二进制文件可以在板子上运行，却不能运行在服务器上。这个例子可以让实验者很好地认清楚交叉编译的意义和功能。