实验 12：Mount 挂载实验@

一、 实验目的

1. 掌握配置 NFS 服务的方法。
2. 掌握 mount 挂载 usb/sd 的方法。

二、实验内容

1。配置服务器端 NFS 服务。
2．实现 nfs 挂载
3．mount 挂载 usb 和 sd 设备。

三、实验设备

1. 硬件：ARM9 系统教学实验系统；串口线；网口线 ；U 盘；SD 卡；服务器。
2. 软件：Linux 操作系统；远程登录系统 putty。

四、预备知识

1.概述

NFS（Network File System）即网络文件系统，是 FreeBSD 支持的文件系统中的一种，它允许网络中的计算机之间通过 TCP/IP 网络共享资源。在 NFS 的应用中，本地 NFS 的客户端应用可以透明地读写位于远端 NFS 服务器上的文件，就像访问本地文件一样。

mount 是 Linux 下的一个命令，可以将 Windows 分区作为 Linux 的一个“文件”挂载到 Linux 的一个空文件夹下，从而将 Windows 的分区和/mnt 目录联系起来，因此只要访问这个文件夹，就相当于访问该分区。

mount 命令指示操作系统使文件系统在指定位置（安装点）可用。此外，可以用 mount 命令构建由目录和安装文件（file mounts）组成的其他文件树。mount 命令通过在 Directory 参数指定的目录上使用 Device/Node: Directory 来安装表示为设备的文件系统。mount 命令完成以后，指定的目录变为新安装文件系统的根目录。

只有 root 权限的用户或系统组成员和对安装点有写权限的用户能进行文件或目录安装（directory mounts）。文件或目录可以是符号链接。mount 命令使用真实的用户标识，而不是有效的用户标识，来确定用户是否有相应的访问权限。假定系统组成员对安装点或在/etc/filesystems 文件中指定的安装有写入权限，他们能发出设备安装（device mounts）。有 root 权限的用户能发出任意的 mount 命令。

如果用户属于系统组并且有相应的存取权限，则能安装设备。安装设备时，mount 命令使用 Device 参数作为块设备名，Directory 参数作为文件系统所要安装的目录。

2.基本原理

在 Linux 系统中，设备在上层都被映射为设备文件，比如 IDE 硬盘被映射为设备文件/dev/hda1，U 盘被映射为设备文件/dev/sda1。如果用户直接访问这些设备文件，则得到的一堆二进制代码。所以，为了方便用户的使用，Linux 规定，必须将该设备文件挂载到某一目录下(常用的是/mnt 目录)，用户对该目录(比如/mnt 目录)的操作(读/写)就是对设备文件的操作，也就是对设备的操作。当然，在实际应用中，常在/mnt 目录下新建一子目录，比如

hdisk(IDE 硬盘)、usb (U 盘)，然后将设备文件挂载到该子目录下。

五、实验说明

本实验中假设 Linux 系统中默认安装了 NFS 服务程序，如果没有安装，则需要在安装 Linux 系统时选择默认安装 NFS 服务。假设系统中默认安装了 usb/sd 驱动，系统能自动检测 usb/sd 设备。

六、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）：

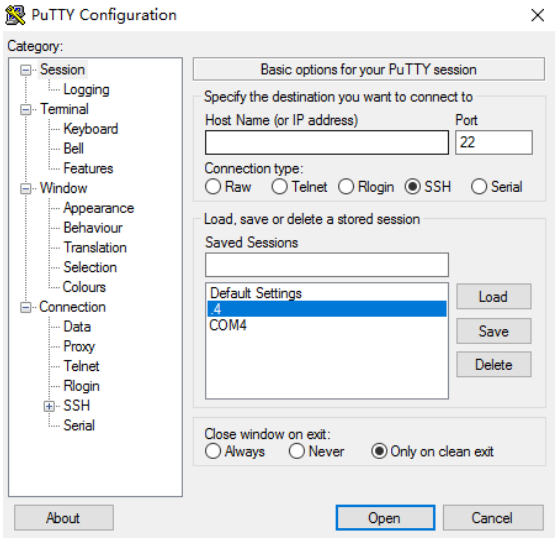

图 1 打开 putty 连接

(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：

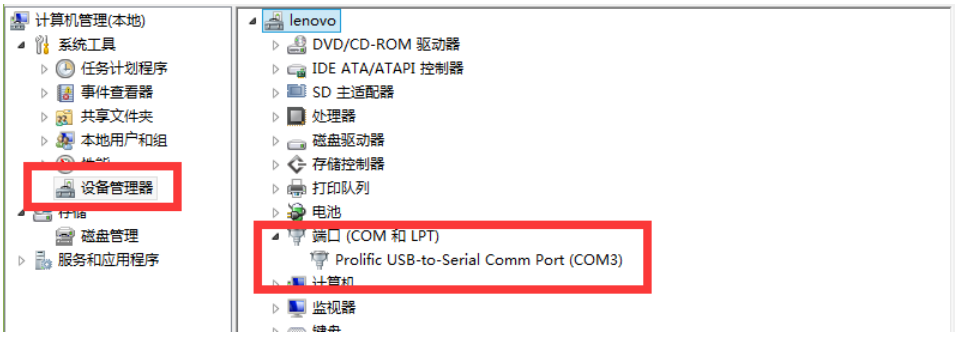

图 2 端口号查询

(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：

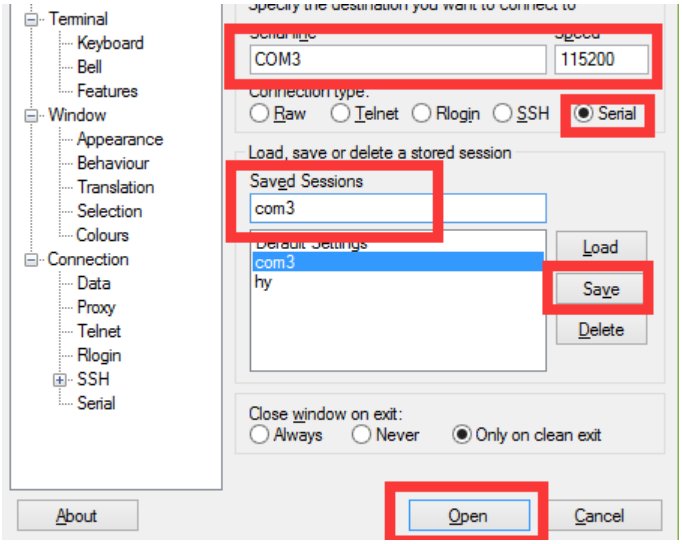

图 3 putty 串口连接配置

(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：配置 nfs 服务器设置

1. 进入 Linux 服务器系统的/etc 目录，命令如下：

```
$ cd /etc/
```

2. 编辑/etc/exports 的文件，sudo 命令是进入 root 权限，这里需要输入登录密码，命令如下：

```
$ sudo vi exports
[sudo] password for st1:
```

进入如下所示 exports 文件，在 exports 文件中添加一行：

```
/home/挂载目录  192.168.*.*(rw,sync,no_root_squash)
```

具体过程如下所示：

```
*/*tc/exports：the access control list for filesystems which may be exported
*                    to NFS clients.    See exports(5).
*
*Examples for NFSv2 and NFSv3：
*/srv/homes                hostname1(rw,sync,no_subtree_check)    hostname2(rv,sync,no_
subtree_check)
*
*Examples for NFSv4
*/srv/nfs4                    gss/krb5i(rw, sync, fsid=0, crossmnt,no_subtree_check)
*/srv/nfs4 /homes                gss/krb5i(rw, sync, no_subtree_check)
*
/home/st1/nfs 192.168.1.*    (rw, sync, no_root_squash)
```

从最后一行中可以看出，本文为系统添加了一行,192.168.1 根据具体的网关而言。如果虚拟机在 192.168.0.1 网关下则改为,192.168.0.*

```
/home/xxx/nfs 192.168.1.*      (rw sync no_root_squash)
```

至此，NFS 服务器的配置已经完成，接下来启动 NFS 服务。如果已启动，则跳过此步骤。

启动 NFS 的命令如下：

```
$ sudo ./etc/rc.d/init.d/nfs start
```

如果之前已启动 NFS，更改后可用以下命令：

```
$ sudo /etc/rc.d/init.d/nfs restart
```

步骤 3：文件夹挂载

挂载

服务器端的 NFS 服务配置完成以后，启动实验板，在串口调试工具中开始挂载文件夹，在 mount 之前，必须先配置。加上 ifconfig eth0 192.168.1.***命令，修改实验板系统 IP。（192.168.1.***表示的是实验箱的具体 ip 地址，注意实验箱的 ip 地址要和被挂载的服务器处在同一网段）

mount 过程如下所示：(此步骤在实验箱上进行)

```
[root]# mount   -t   nfs   -o   nolock   192.168.1.65:/home/st1/nfs   /mnt/mtd/
[root]#
```

（注：st1 是用户名，每个人创建 ubuntu 时的用户名都不一样）

为了验证挂载是否成功，输入 df 命令查看，结果如下所示：

```
[root]#df
Filesystem              1K-blocks      Used      Availabled    Use%    Mounted   on
/dev/sda1               806368         45520     719884        6%      /
/dev/mtdblock1          193241632      102773502 80652000      56%     /mnt/mtd
192.168.1.65：/home/st1/nfs
                        193241632      102773502 80652000      56%     /mnt/mtd
```

从上可以看出已经将服务器上的/home/st1/nfs（192.168.1.65:/home/st1/nfs）目录挂载到了实验箱文件系统的/mnt/mtd 目录下。也就是说此时实验箱可以通过/mnt/mtd 目录直接访问服务器上的/home/st1/nfs 目录。可以在服务器端进入/home/st1/nfs 目录和在实验箱中进入/mnt/mtd 目录对比里面的内容，可以发现内容是一样的，并且在任意端向目录中创建新文件，在另一端均可见。

2. 卸载

为了将/192.168.1.65:/home/st1/nfs 目录与/mnt/mtd 目录卸载分开，首先退到 root 目录下（cd / 请注意卸载命令发生在实验箱端，且一定要在卸载挂载前退出挂载目录，否则会报错，报错内容为设备忙），需要使用 umount 命令（umount 被挂载目录），如下所示：

```
[root]# umount   /mnt/mtd
[root]#df
Filesystem              1K-blocks      Used      Availabled    Use%    Mounted   on
/dev/sda1               806368         45520     719884        6%      /
/dev/mtblock1           806368         45520     719884        6%      /mnt/mtd
[root]#
```

步骤 4：usb 挂载

（1）将 U 盘插入实验板的 USB 接口处，实验板中的串口调试工具出现以下信息提示：

```
[root]# [ 149.340000] usb 1-1.3:new high speed USB device using musb_hdrc and address 4
```

（2）使用 fdisk-l 查看盘符信息，如下所示：

```
[root]# fdisk   -1
Disk /dev/sda: 4057 MB, 4057989120 bytes
91 heads, 45 sectors/track, 1935 cylinders
Units = cylinders of 4095 * 512 = 2096640 bytes

   Device Boot      Start         End      Blocks   Id System
/dev/sda1               1        1936     3962852    b Win95 FAT32
```

```
[root]#
```

（3）创建一个/mnt/usb 文件夹，如下所示：

```
[root]# mkdir   /mnt/usb
[root]#
```

（4）把 sda1 盘符 mount 到/mnt/usb 文件夹上，如下所示：

```
[root]# mount/dev/sda1 /mnt/usb/
[root]#
```

（5）进入/mnt/usb 文件夹，查看该文件夹中的内容，如下所示：

```
[root]#   cd /mnt/usb/
[root]#   ls
h264            disk.tar.gz     linuxrc         sbin        tmp
bin             etc             mnt             share       usr
data.h264       init            proc            shm         var
dev             lib             root            sys
[root]#
```

（6）卸载 U 盘，先退出到根目录下，再解除挂载，如下所示：

```
[root@zjut   usb]#   cd   /
[root@zjut ~]# umount   /mnt/usb
```

（7）解除挂载以后，可再次进入/mnt/usb 文件夹，输入 ls 查看，若文件夹内已经没有内容，说明解除挂载成功：

```
[root@zjut ~]# cd   /mnt/usb
[root@zjut   usb]#   ls
[root@zjut   usb]#
```

步骤 5：SD 卡挂载

将 SD 卡插入卡槽中，putty 上会跳出以下信息：

```
[root@zjut ~]# [    557.610000] mmcblk0: mmc0:aaaa SL32G 30183936KiB
[    557.620000]   mmcblk0: p1
```

跳出以下信息后，输入 mount，将会跳出挂载信息，我们可以找到以下信息，说明 SD 卡已经自动挂载到/mnt/mmc 文件夹中。

```
/dev/mmcblk0p1        on        /mnt/mmc        typevfat
 (rw,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso8859-1)
```

如果要重新挂载 SD 卡，先输入命令，解除 SD 卡的挂载，之后可进入/mnt/mmc/image 文件夹内查看，里面没有内容。

（注：此时假设 SD 卡上 存在 image 目录且该目录下有图片文件）

```
[root@zjut ~]#   umount   /mnt/mmc
```

之后可输入命令重新挂载

```
[root@zjut ~]# mount -t vfat /dev/mmcblk0p1 /mnt/mmc
```

再进入/mnt/mmc/image 文件夹中查看，里面就会显示出 SD 卡的内容了

```
2016-01-27-16-31-36_null.jpg   2016-03-31-16-34-26_null.jpg
2016-03-29-14-29-54_null.jpg   2016-03-31-16-55-27_null.jpg
2016-03-29-14-30-05_null.jpg   2016-03-31-16-55-39_null.jpg
```

```
2016-03-30-16-10-11_null.jpg    2016-03-31-17-03-04_null.jpg
2016-03-30-16-10-40_null.jpg    2016-03-31-17-03-08_null.jpg
2016-03-31-16-34-12_null.jpg    2016-04-01-10-00-44_null.jpg
2016-03-31-16-34-21_null.jpg    2016-05-26-08-51-18_null.jpg
```

我们再退出目录，输入命令解除挂载。并重新进入/mnt/mmc/image 文件夹中，看文件信息，里面没有文件，说明解除挂载成功。

```
[root@zjut ~]#    cd ~
[root@zjut ~]#    umount    /mnt/mmc
```

实验结束。