实验 14：Linux 内核编译实验

一、实验目的

1．掌握配置和编译 Linux 内核的方法。
2．掌握 Linux 内核的编译过程。
3．熟悉 Linux 系统一些基本内核配置。
4．熟悉内核编译的常见命令。

二、实验内容

1．利用编译进内核的方法配置内核。
2．利用动态加载方法配置内核。
3．编译 Linux 内核镜像。
4．编译 Linux 内核模块。

三、实验设备

1．硬件：PC 机；dm365 系统教学试验箱；串口线；网线。
2．软件：PC 机操作系统；Windows 下超级终端 putty。
3．环境：ubuntu 系统版本 12.04；内核版本 kernel-for-mceb。

四、预备知识

1、概述

1.1 什么是 Linux 内核

内核是操作系统的核心部分，为应用程序提供安全访问硬件资源的功能。直接操作计算机硬件是很复杂的，内核通过硬件抽象的方法屏蔽了硬件的复杂性和多样性。通过硬件抽象的方法，内核向应用程序提供统一和简洁的接口，应用程序设计复杂度降低。实际上，内核可以看成是一个资源管理器，内核管理计算机中所有的硬件资源和软件资源。

1.2 Linux 内核版本

LInux 内核版本采用两个分割的“.”点，形式如”X.Y.Z”来表示。其中 X 表示主版本号，Y 表示次版本号，Z 表示补丁号。奇数代表不稳定版本，偶数代表稳定版本。Linux 内核的官方网站为 http:www.kernel.org。该站点提供各种版本的代码和补丁，用户可以根据需要自由下载。试验箱所用的内核是基于 2.6.18 版本改进来的，并命名为 kernel-for-mecb。考虑到后续实验如内核的移植等需要，实验采用 kernel-for-mceb 内核。

2、原理

2.1 内核配置和编译

内核编译主要分成配置和编译两部分。其中配置是关键，许多问题都是出在配置上。Linux 内核编译配置提供多种方法。如：

```
#make menuconfig  //基于图形工具界面
#make config  //基于文本命令行工具，不推荐使用
#make xconfig  //基于 X11 图形工具界面
```

由于对 Linux 还处在初学阶段，所以选择了简单的配置内核方法，即 make menuconfig。在终端输入 make menuconfig，等待几秒后，终端变成图形化的内核配置界面。进行配置时，大部分选项使用其缺省值，只有一小部分需要根据不同的需要选择硬件介绍。同时内核还提供动态加载的方式，为动态修改内核提供了灵活性。

2.2 内核编译系统

Linux 内核的复杂性，使其需要一个强大的工程管理工具。在 Linux 中，提供了 Makefile 机制。Makefile 是整个工程的编译规则。一个工程中源文件不计其数，按其类型、功能、模块被放在不同的目录中。Makefile 定义了一系列的规则来指定哪些文件需要先编译，那些文件需要后编译，哪些文件需要重新编译甚至进行更复杂的操作。Makefile 带来的直接好处就是自动化编译，一当写好，只要一个 make 命令，整个工程自动编译，极大提高效率。

内核编译时候通过 Makefile 规则将不同的文件进行整合。系统中主要有五种不同类型的文件，其类型和作用如下表所示：

表 1 编译的文件类型与作用表

文件类型	作用
Makefile	顶层 Makefile 文件
.config	内核配置文件
arch/$(ARCH)/Makefile	机器体系 Makefile 文件
scripts/Makefile.*	所有内核 Makefile 共用规则
Kbuild Makefile	其他 makefile 文件

表中.config 即是内核配置的文本文件。它记录了了文件的配置选项，可直接对其进行修改，只是较为繁琐，故不推荐使用。事实上，使用其他方式配置的文件最终都会保存到.config 中，换言之，内核配置就是围绕着.congfig 文件进行的。

内核编译的时候，顶层的 Makefile 文件在开始编译子目录下的代码之前，设置编译环境和需要用到的变量。顶层 Makefile 文件包含着通用部分，arch/$(ARCH)/Makefile 包含架构体系所需设置，其中也会设置一些变量和少量的目标。实验文档结尾有部分编译源码供参考。

五、实验步骤

以 root 用户名登录服务器 Linux，进入学生的学习目录，将“Linux 嵌入式实验/内核编译实验”目录下的 arm-linux-2.6.tar.gz 包通过解压缩到学习目录下，会建立一个 arm-linux-2.6 文件夹。

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）：

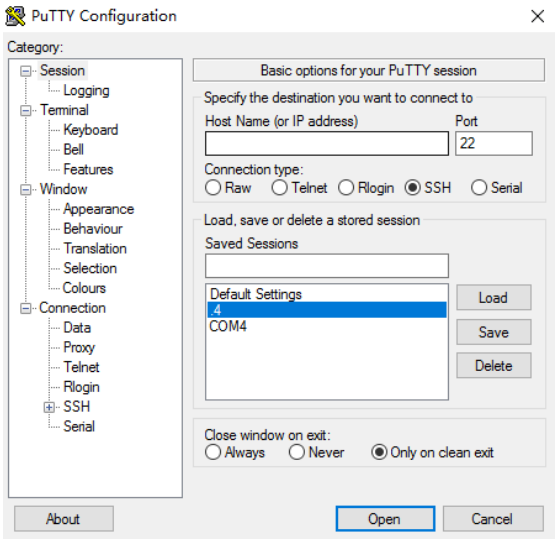

图 1 打开 putty 连接

(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：

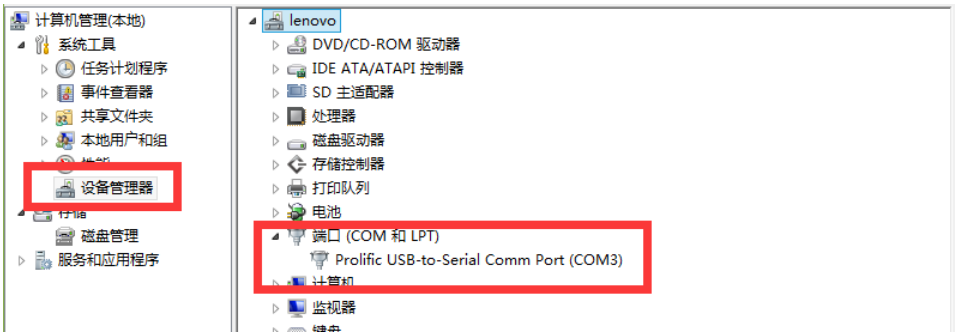

图 2 端口号查询

(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：

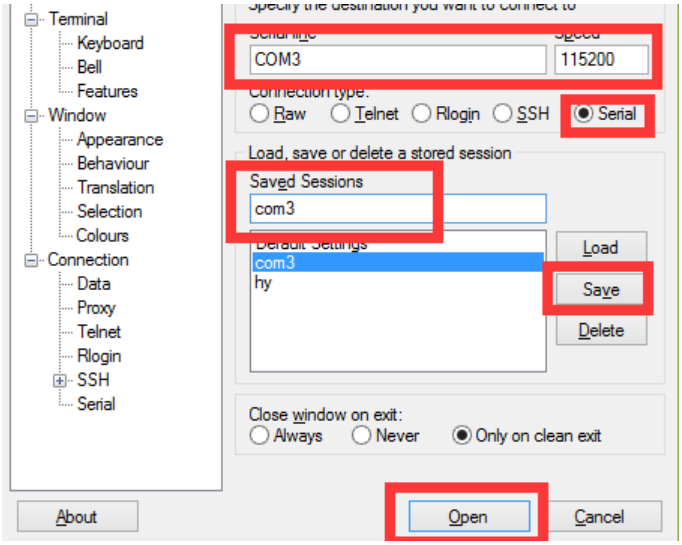

图 3 putty 串口连接配置

(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：在服务器窗口操作，进入内核所在目录

实验箱的内核基于 2.6.18 改进来的 kernel-for-mceb。

步骤 3：进入内核进行配置

在配置之前首先输入命令 sudo -s，并输入密码，登录超级用户，之后再次输入命令 vim /etc/profile 检查交叉编译路径是否正确。之后退出，输入命令 source /etc/profile ，使环境变量生效。

进入内核目录，执行命令为 cd kernel-for-mceb。如果不是第一次编译内核，那么请先

运行：make mrproper 清除以前的配置，回到默认配置。然后继续进行内核配置，执行命令为 make menuconfig。出现窗口如图 7-53 所示：

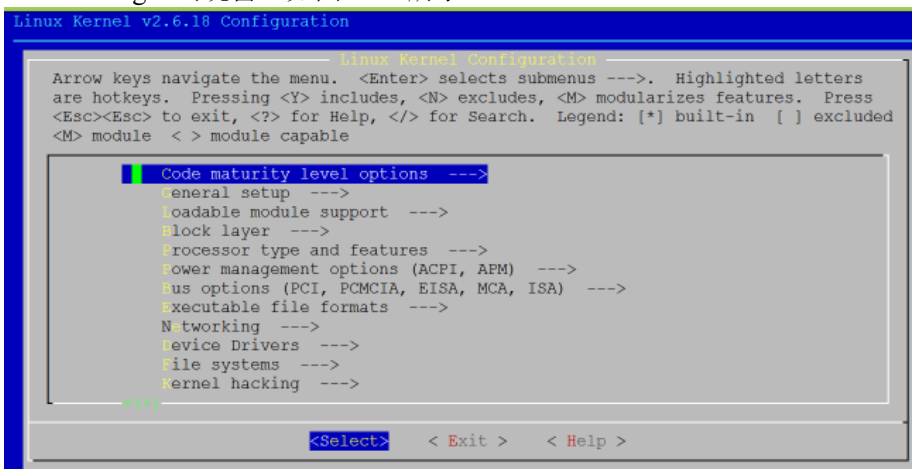

图 7-53　配置内核

编译内核时候往往根据自己的需要来编译自己所需要的。下面是一些常见的驱动选项。

表 1　驱动选项表

选项	说明
MemoryTechnology Devices(MTD)	配置存储设备，需要该选项使 Linux 可以读取闪存卡等（Flash、Card）存储器
Parallel port support	配置并口。如果不使用，可不选
Block devices	块设备支持
ATA/ATAPI/MFM/RLL support	IDE 硬盘和 ATAPI 光驱，纯 SCSI 系统且不使用这些接口，可以不选
SCSI device support	SCSI 仿真设备支持
Multi-devicesupport(RAID and LVM)	多设备支持（RAID 和 LAM）
Fusion MPT device support	MPT 设备支持
IEEE 1394(FireWire)support	IEEE 1394（火线）
I2O device support	I2O 设备支持。如果有 I2O 界面，必须选中。是由于智能 I/O 系统的标准接口
Network device support	内核在没有网络支持选项的情况下甚至无法编译。是必选项。
ISDN subsystem	综合数字业务网
Input device support	输入设备，包括鼠标、键盘等
Character devices	字符设备，包括虚拟终端、串口、并口等设备
I2C support	用于监控电压、风扇转速、温度等。

Hardware Monitoring support	需要 I2C 的支持
Misc devices	杂项设备
Multimedia Capabilities Port drivers	多媒体功能接口驱动
Multimedia devices	多媒体设备
Graphics support	图形设备/显卡支持
Sound	声卡
USB support	USB 接口支持配置
MMC/SD Card support	MMC/SD 卡支持

步骤 4：配置选择

内核定制，选择自己需要的功能。按键盘空格键进行选择，*表示选定直接编译进内核，M 表示选定模块编译为动态加载模块。在这里，以让内核支持 ntfs 文件系统为例。

(1) 在 make menuconfig 命令打开的窗口中选择到 Files systems。如图 7-54 所示：

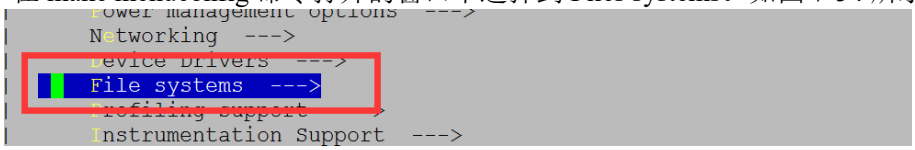

图 7-54 文件系统配置

(2) 敲回车后，继续选择能支持 ntfs 文件系统类型的选项。如图 7-55 所示：

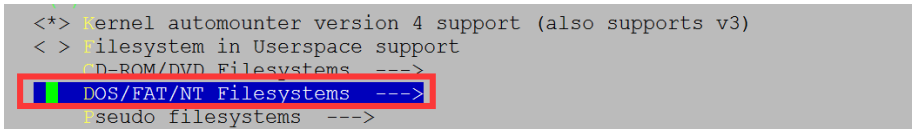

图 7-55 文件系统选项

(3) 继续回车键，最后选择我们需要的 ntfs 文件系统类型。如图 7-56 所示：

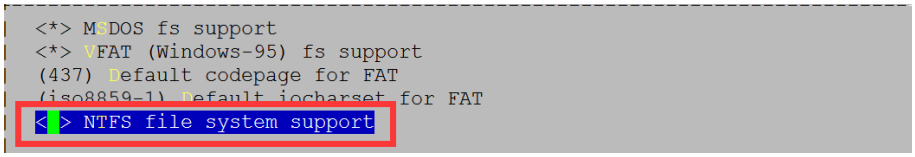

图 7-56 ntfs 文件系统选项

(4) 按空格选择编译进内核。并在键盘上按左右键移动光标到退出键按钮，按回车键不断退出。如图 7-57 所示：

图 7-57 配置为编译进内核

(5) 不断回车键后，出现是否保存界面，选择 yes 保存配置，回车键退出，如图 7-58 所示：

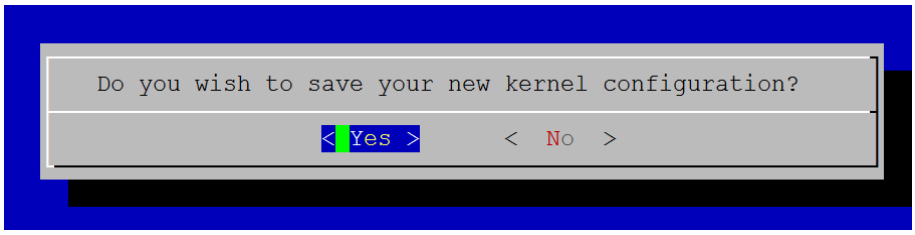

图 7-58 配置退出保存界面

步骤 5：编译内核镜像

编译内核时候一般需要 root 权限，对于特定用户加 sudo 命令即可。

在内核目录下使用命令 make uImage。回车键后内核开始编译，等到出现 `Image arch/arm/boot/uImage is ready` 表示编译结束。编译好后在目录 arch/arm/boot/下生成一个 uImage 二进制文件。如下所示：

```
Load Address: 80008000
Entry Point: 80008000
  Image  arch/arm/boot/uImag is ready
root@ubuntu:~/kernel-for-mceb$
```

这就是利用编译进内核的方法编译生成的内核镜像，可以移植到实验板子上。对于一般的要求，利用编译进内核的方法就足够了。不过对于许多实验和工程的要求，为了减轻内核的负担，往往是需要利用动态加载的方法。下面继续实验来熟悉动态模块编译的方法。

步骤 6：将 ntfs 文件系统配置成模块方式

按照步骤 4 中的方法，将 NTFS file system support 前面*改变成为 M，即将 ntfs 文件系统配置成为模块加载形式。如图 7-60 所示：

图 7-60 内核配置为模块方式

步骤 7：编译模块

编译好一个驱动程序，执行 make modelus 命令，将会生成一个 ntfs.ko 模块，位于 fs/ntfs/目录下。如下所示：

```
LD [M]    fs/nls/nls_ascii.ko
CC        fs/nls/nls_utf8.mod.o
LD [M]    fs/nls/nls_utf8.ko
CC        fs/nls/nls_ntfs.mod.o
LD [M]    fs/nls/nls_ntfs.ko
CC        lib//libcrc32c.mod.o
LD [M]    lib//libcrc32c.mod.ko
```

步骤 8：重新编译内核镜像

由于编译生成的模块需要被加载到内核中才能使用，而开始编译生成的内核是一个可以支持 ntfs 文件系统的内核。可以先执行 make clean，清除刚才生成的镜像，然后执行 make uImage 生成一个新内核镜像。即新生成的内核不支持 ntfs 文件系统，从而可以将 ntfs 文件模块添加进去，使内核可以支持 ntfs 文件系统。这就是模块编译的方法。关于查看、加载以及卸载模块的方法可以参考其他实验。

以上介绍的就是内核编译方法。以添加支持 ntfs 文件系统为例，介绍了两种方法：一种是直接编译进内核中，另一种是编译成模块加载到内核中。

```
附录：部分 makefile 源码解释
#版本基本信息
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 18
EXTRAVERSION =-plc
NAME=Avast! A bilge rat!
######################################################################
MAKEFLAGS += --no-print-directory     #不要再屏幕上打印"Entering directory..."，始终被自动的传递给所有的子 make。
######################################################################
ifdef V     #V=1。
ifeq ("$(origin V)", "command line")   #函数 origin 指示变量是哪里来的。
  KBUILD_VERBOSE = $(V)     #把 V 的值作为 KBUILD_VERBOSE 的值。
 Endif
endif
  ifndef KBUILD_VERBOSE   #即默认我们是不回显的，回显即在命令执行前显示要执行的命令。
 KBUILD_VERBOSE = 0
endif
######################################################################
ifdef   SUBDIRS
  KBUILD_EXTMOD ?= $(SUBDIRS)      #条件操作命令。
endif
Ifdef   M   #M 用来指定外在模块目录。
    ifeq ("$(origin M)", "command line")
       KBUILD_EXTMOD := $(M)
    endif
 Endif
######################################################################
ifeq ($(KBUILD_SRC),)   #变量，是否进入下一层。
######################################################################
ifdef O   #把输出文件放在不同的文件夹内。
  ifeq ("$(origin O)", "command line")
     KBUILD_OUTPUT := $(O)     #用于指定我们的输出文件的输出目录。
  endif
Endif
######################################################################
PHONY := _all   #默认目标是全部。
```

```
_all:
ifneq ($(KBUILD_OUTPUT),)   #检测输出目录。
#####################################################################
saved-output := $(KBUILD_OUTPUT)
KBUILD_OUTPUT := $(shell cd $(KBUILD_OUTPUT) && /bin/pwd)   #测试目录是否存在，存在则赋
给 K BUILD_OUTPUT。
$(if $(KBUILD_OUTPUT),, \     #这里的为空即表示输出目录不存在
     $(error output directory "$(saved-output)" does not exist)) )   #使用了 error 函数
PHONY += $(MAKECMDGOALS)   #将任何在命令行中给出的目标放入变量。
$(filter-out _all,$(MAKECMDGOALS)) _all: )   #表示要生成的目标。
     $(if $(KBUILD_VERBOSE:1=),@)$(MAKE) -C $(KBUILD_OUTPUT) \
     KBUILD_SRC=$(CURDIR) \
KBUILD_EXTMOD="$(KBUILD_EXTMOD)" -f $(CURDIR)/Makefile $@
#$@表示取消回显的意思。
#####################################################################
skip-makefile := 1 #跳转目录所用
endif   #KBUILD_OUTPUT 结束处。
endif   #KBUILD_SRC 结束处
#####################################################################
#以下设置编译连接的默认程序,都是变量赋值操作。
AS          = $(CROSS_COMPILE)as
LD          = $(CROSS_COMPILE)ld
CC          = $(CROSS_COMPILE)gcc
CPP         = $(CC) -E
AR          = $(CROSS_COMPILE)ar
NM          = $(CROSS_COMPILE)nm
STRIP           = $(CROSS_COMPILE)strip
OBJCOPY         = $(CROSS_COMPILE)objcopy
OBJDUMP         = $(CROSS_COMPILE)objdump
AWK         = awk
GENKSYMS        = scripts/genksyms/genksyms
DEPMOD          = depmod
KALLSYMS        = scripts/kallsyms
PERL            = perl
CHECK           = sparse
CHECKFLAGS          := -D__linux__ -Dlinux -D__STDC__ -Dunix -D__unix__ -Wbitwise $(CF)
MODFLAGS        = -DMODULE
CFLAGS_MODULE     = $(MODFLAGS)
AFLAGS_MODULE     = $(MODFLAGS)
LDFLAGS_MODULE    = -r
CFLAGS_KERNEL     =
AFLAGS_KERNEL     =
#####################################################################
```