



**立宇泰电子**

Liyutai Elec. CO., Ltd.

专注于做最好的嵌入式计算机系统供应商

# ARMSYS2410 开发套件Linux2.6.15 内核

## 使用说明

Rev 1.1

2007年7月10日

杭州立宇泰电子有限公司

HangZhou LiYuTai Elec.Co.,Ltd

<b>1 安装交叉编译器</b> .....	<b>4</b>
<b>2 准备工作</b> .....	<b>4</b>
2. 1 解压内核.....	4
2. 2 配置内核.....	4
2. 3 编译内核.....	8
2. 4 准备好根文件系统.....	8
2. 5 启动BOOTLOADER并设置启动参数.....	8
<b>3 固化内核</b> .....	<b>9</b>
<b>4. NFS方式MOUNT主机</b> .....	<b>13</b>
4. 1 在主机上启动NFS服务.....	13
4. 2 在目标板上进行设定.....	15
4. 3 启动LINUX.....	16
<b>5. NFS方式更新文件系统</b> .....	<b>16</b>
5. 1 更新文件系统.....	16
5. 2 从YAFFS文件系统启动.....	18

感谢使用立宇泰电子研制生产的 ARMSYS2410 系列产品！

本文将指导你如何安装 ARMSYS2410 的 Linux2.6.15 版内核。主要介绍了以下内容：

- 安装交叉编译器
- 解压并编译内核
- 利用 bootloader 下载固化内核
- 通过 nfs 方式建立 yaffs 根文件系统

跟随本文档进行操作之前，请在 PC 机上安装 Linux, 我们的测试在 Red Hat9.0 下进行。

- 如果您使用 linux2.4 请参考光盘中的另一篇文档——《ARMSYS2410 开发套件 Linux2.4.18 内核使用说明》。
- 如果您希望详细了解 u-boot 的使用，请查看另一篇文档——《如何使用 ARMSYS2410 配套的 u-boot》。

# 1 安装交叉编译器

(1) 首先安装交叉编译toolchain，安装包位于光盘中的\Linux \ toolchain\ cross-3.4.3.tgz 在/usr/local 目录下建立名为arm的目录，进入该目录，执行解包：

```
cd /usr/local/arm  
tar xjvf /mnt/cdrom/linux/toolchain/ cross-3.4.3.tgz
```

(2) 编辑/etc/profile，找到 pathmunge /usr/local/sbin 在其下面添加一行：

```
pathmunge /usr/local/arm/release-3.4.3/bin
```

这样，内核或其他应用程序的交叉编译均可用 arm-none-linux-gnueabi- 来指定使用该交叉编译器。

## 2 准备工作

### 2. 1 解压内核

在根目录下建立 armsys2410 目录，进入该目录，执行解包：

```
mkdir armsys2410  
cd armsys2410  
tar xzvf /mnt/cdrom/linux/kernel/kernel-armsys2410-2.6.15-xxxxxx.tgz
```

这里 xxxxxxxx 的内容一般是日期，根据版本的不同而不同。

解压完毕后在 armsys2410 下产生 linux-2.6.15 目录，就是 linux 的内核源码包。

### 2. 2 配置内核

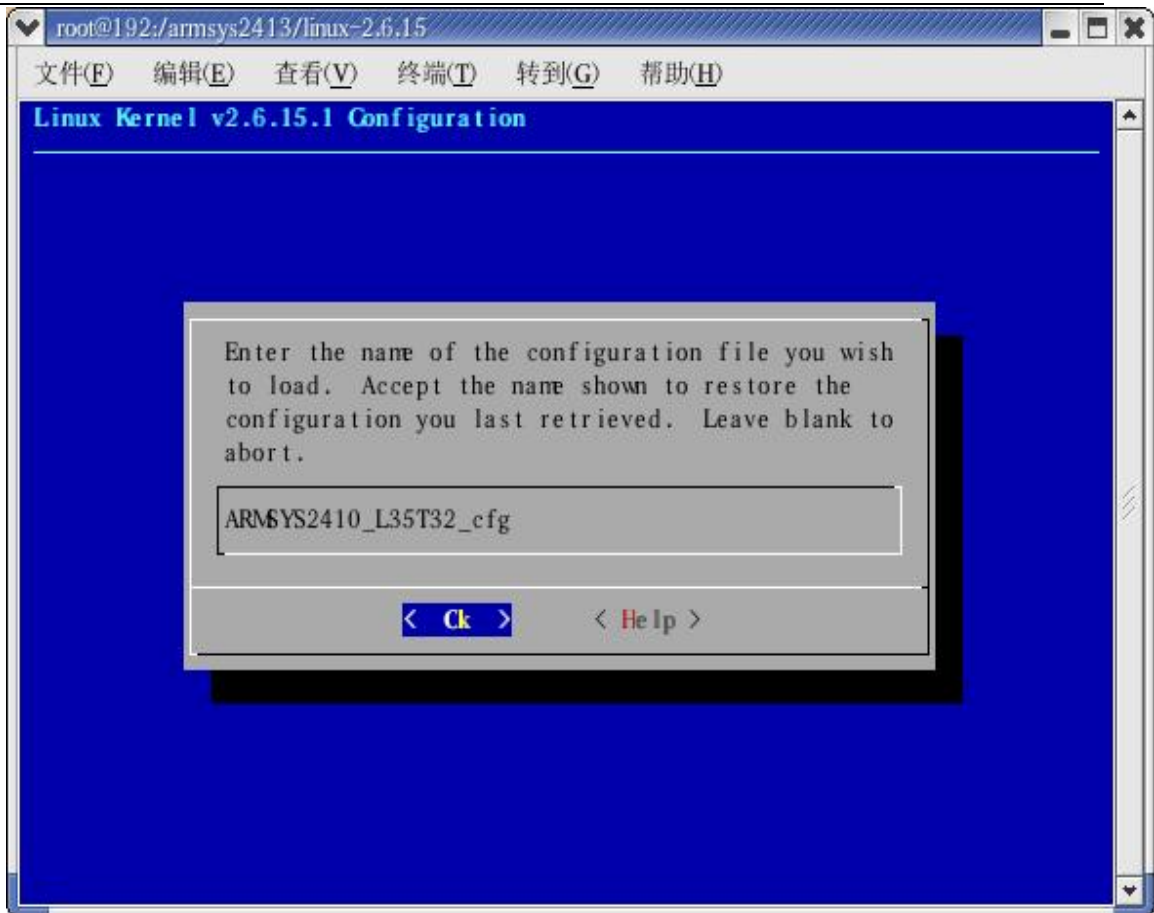
命令行如下：

```
cd /armsys2410/ linux-2.6.15  
make menuconfig
```

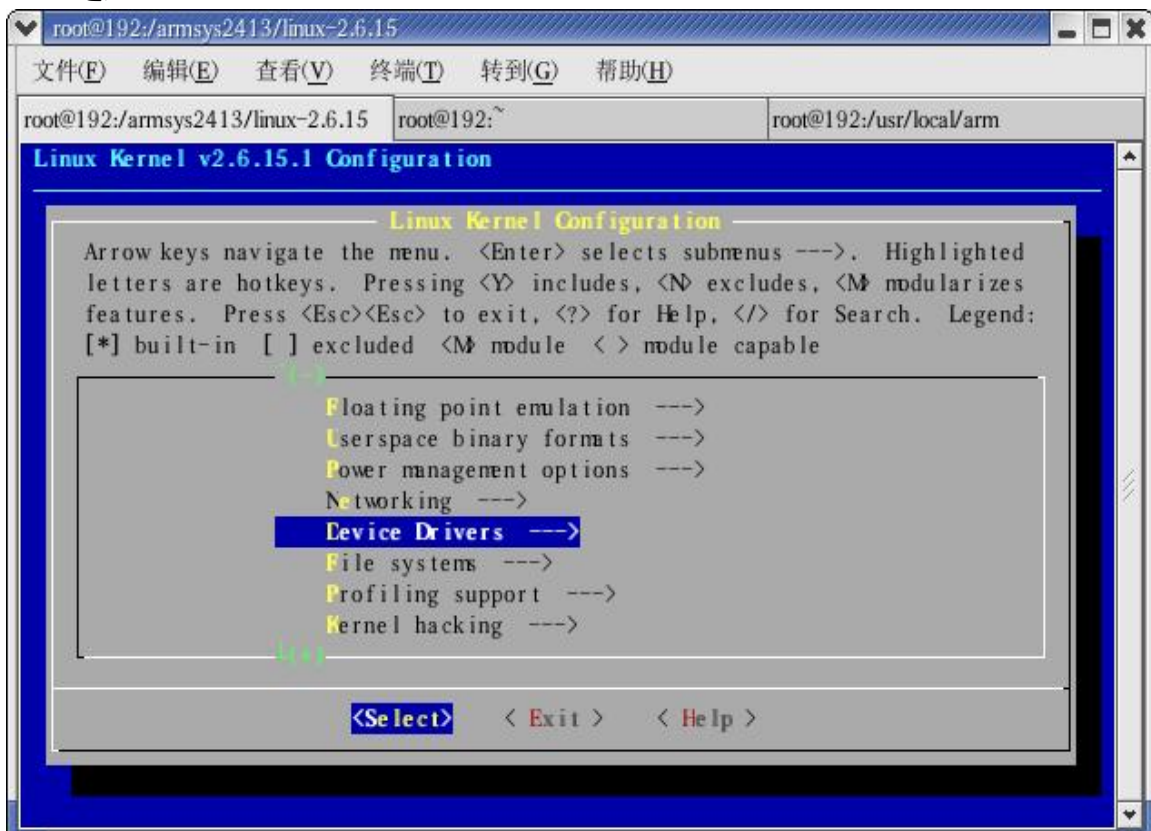
(1) 选中 Load an Alternate Configuration File，敲空格进入：



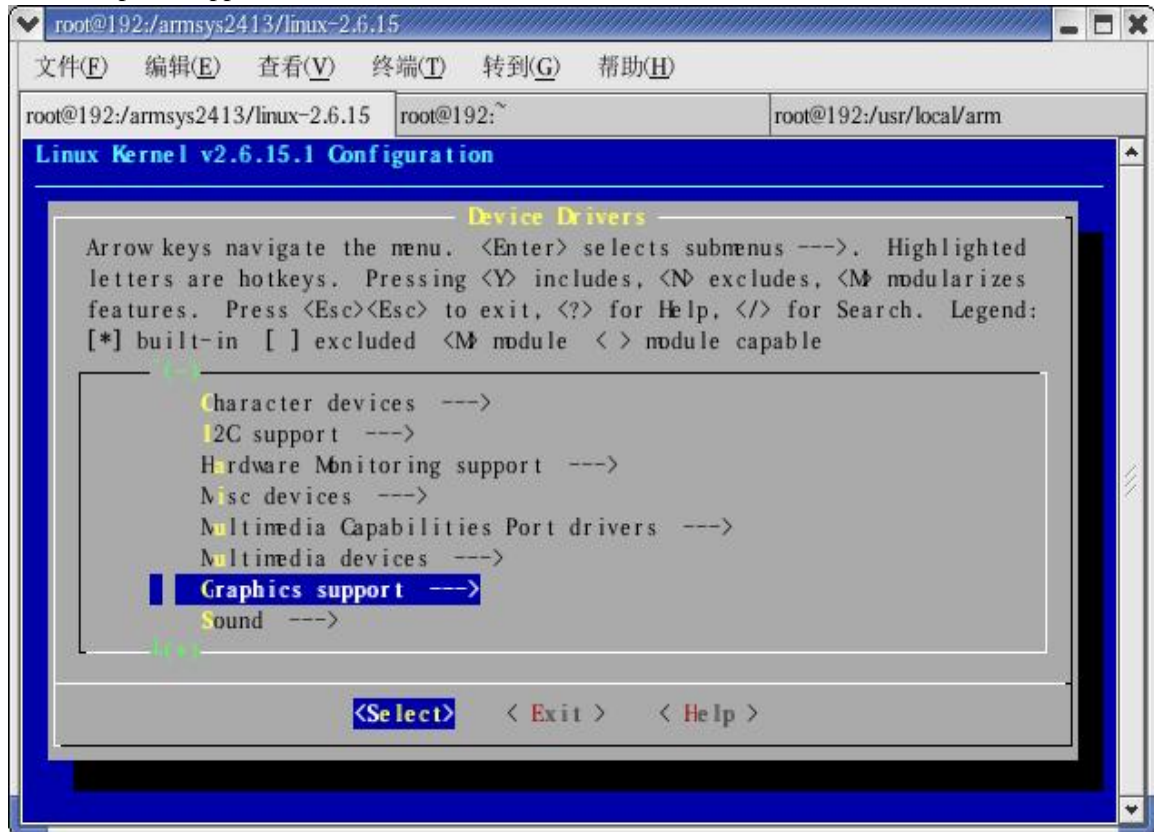
输入 `ARMSYS2410_L35T32_cfg`，这里 L35T32 表示你使用的液晶屏的型号。输入完成点击 OK。如果你使用的不是这款屏，则需要继续按照（2）的步骤重新选择液晶屏的型号。否则跳过第（2）步。



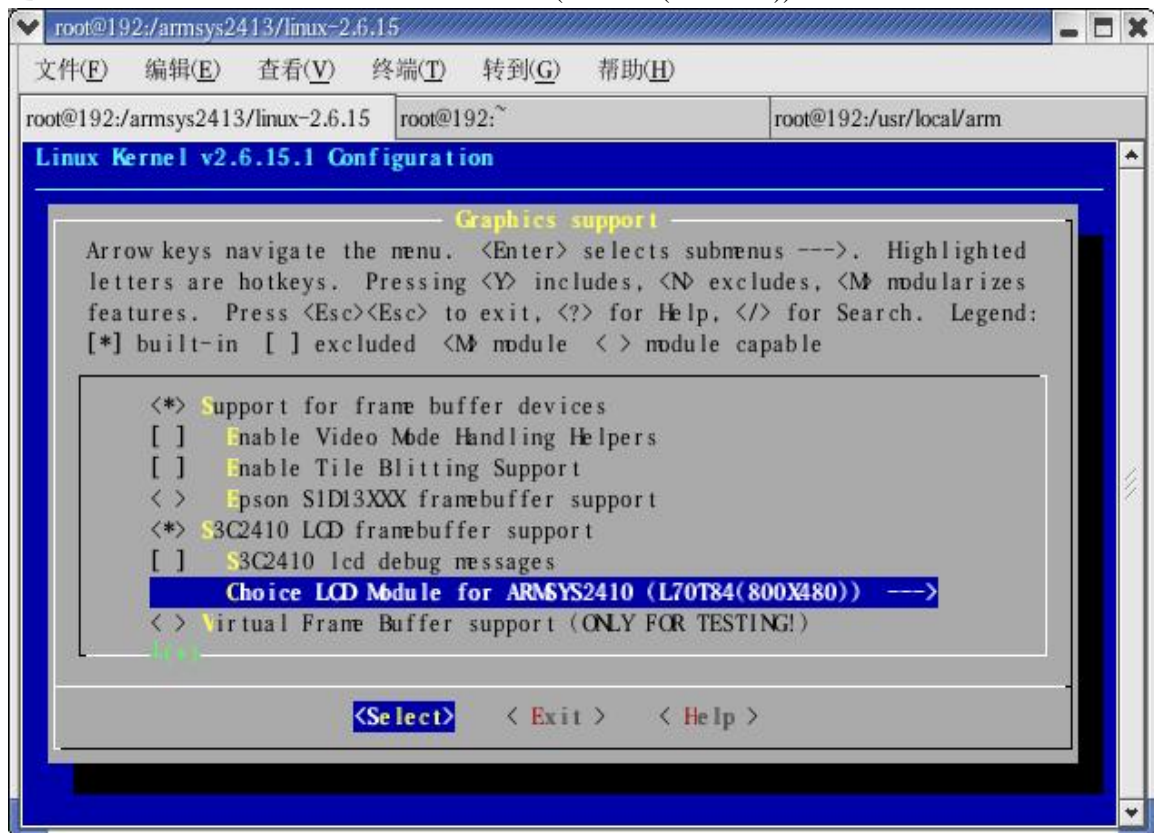
(2) 进入“Device Driver ->”：



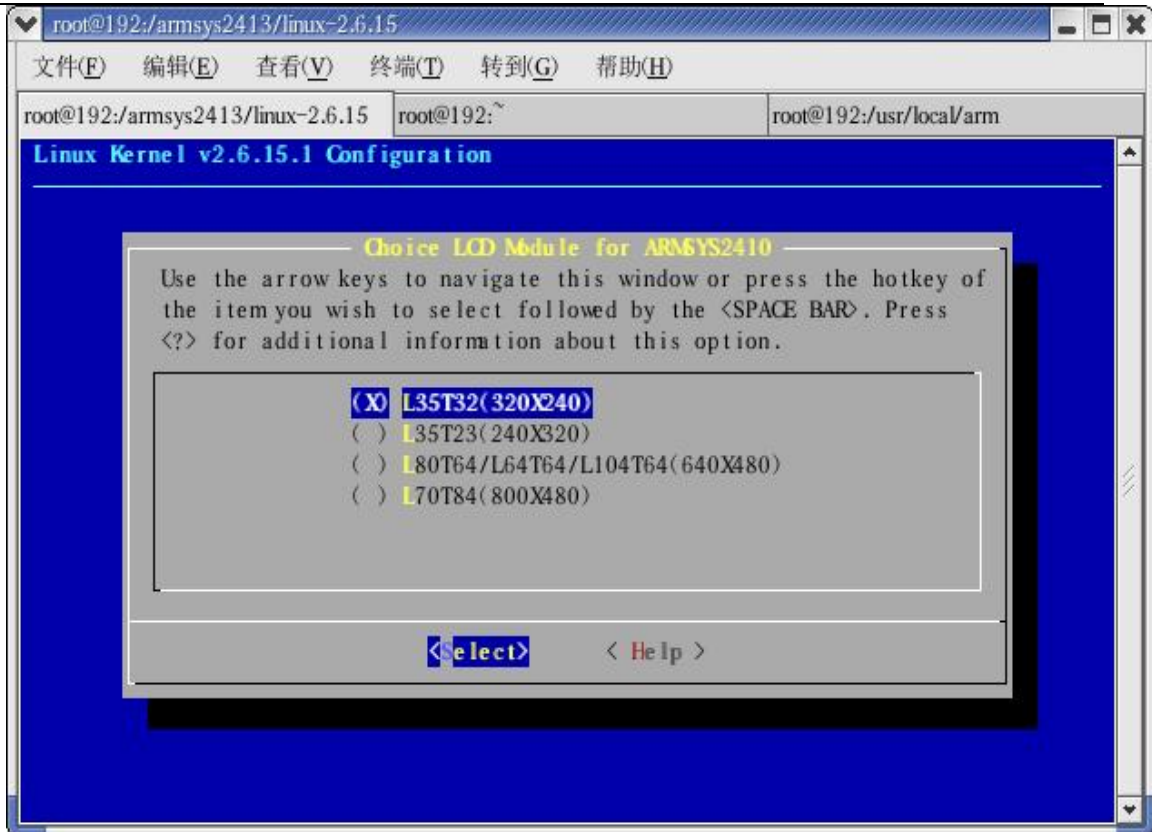
进入“Graphics support →”:



进入“Choice LCD Module for ARMSYS2410 (L70T84 (800X480)) -->”



在这里选中适合你的液晶屏的类型:



选中后退出配置。

这样就完成了针对 ARMSYS2410 的基本配置。

## 2. 3 编译内核

退出配置后，只要输入 make 即可。编译时间较长，大约等待 30 分钟。

编译结束后将在 kernel/arch/arm/boot 目录和/tftpboot/下得到 linux 内核压缩映像文件：zImage。

## 2. 4 准备好根文件系统

光盘资料中提供了文件系统 root\_armsys\_2.6\_xxxxxx，这里 xxxxxx 是发行的日期。

建立一个 root 文件夹，将该压缩包放入文件夹中，将它解压，得到根文件系统所需要目录。不要删除压缩包，将压缩包更名为 root\_armsys.tgz。在第 4 节我们将使用这个文件系统作为 nfs 文件系统，实现开发板的启动和 yaffs 文件系统的烧录。

## 2. 5 启动 Bootloader 并设置启动参数

目前我们提供两种 bootloader：u-boot 和 vivi。

- 为 u-boot 我们专门提供了文档《如何使用 ARMSYS2410 配套的 u-boot》，请到 linux/u-boot/目录下查看。
- 关于 Vivi Bootloader 请查看《ARMSYS2410 开发套件 Linux2.4.18 内核使用说明》文档中相关章节。



由于 u-boot 的下载速度更快，我们将以 u-boot 为 bootloader 介绍下面的步骤。

固化内核之前，必须重新设置 u-boot 的启动参数。

请查看 linux2.6\start\_param 文件，需要重新设置的参数有：

```
#####内核启动相关参数#####
setenv bootcmd nandr c 140000 30008000\; bootm

#####从 nfs 启动根文件系统相关参数#####
setenv linux_arg root=/dev/nfs nfsroot=192.168.253.1:/armsys2410/root
ip=192.168.253.9:192.168.253.1:192.168.253.100:255.255.255.0:armsys.hzlitai.com:eth0:off
console=ttySAC0

#####从 NAND 启动 yaffs 根文件系统相关参数#####
setenv linux_arg noinitrd root=/dev/mtdblock/2 init=/linuxrc console=ttySAC0,115200
```

下列参数在 u-boot 的命令行方式下设置即可。

### 3 固化内核

如果还没有成功编译内核，可以先使用光盘中提供的映像文件 zImage。将它拷贝到主机的 /tftpboot 目录下。连接好 ARMSYS2410 开发板的串口、网口。

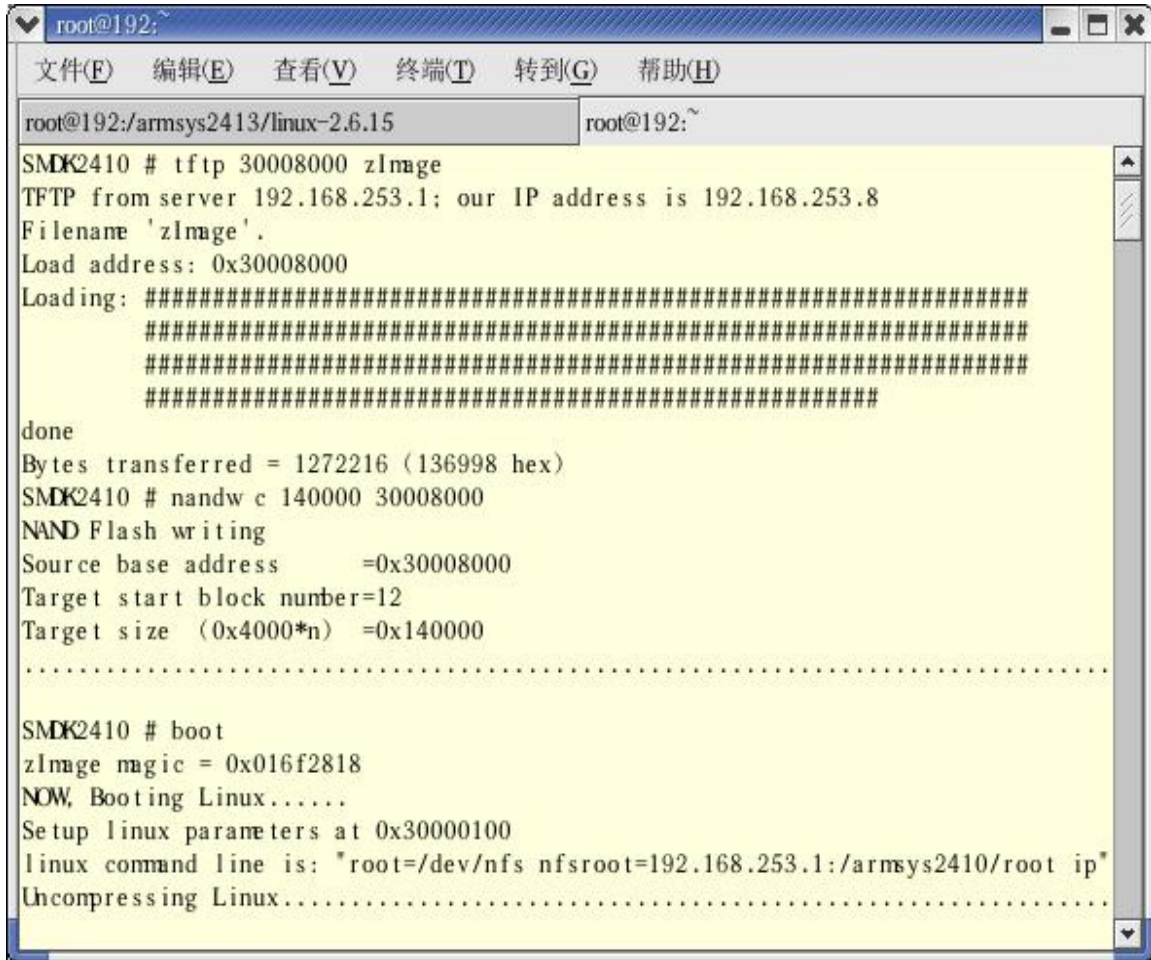
(1) 在 u-boot 命令提示符下输入命令行：

```
SMDK2410# tftp 30008000 zImage
```

开始下载内核映像文件：

(2) 固化内核：

```
SMDK2410# nandw c 140000 30008000
```



(3) 启动内核:

**SMDK2410# boot**

正确的启动信息如下:

SMDK2410 # boot

zImage magic = 0x016f2818

NOW, Booting Linux.....

Setup linux parameters at 0x30000100

linux command line is: "root=/dev/nfs nfsroot=192.168.253.1:/armsys2410/root ip"Uncompressing Linux.....Linux version 2.6.15.1 (root@192.168.253.100) (gcc version 3.4.3 (release) (Cod7CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)

Machine: SMDK2410

ATAG\_INITRD is deprecated; please update your bootloader.

Memory policy: ECC disabled, Data cache writeback

CPU S3C2410A (id 0x32410002)

S3C2410: core 200.000 MHz, memory 100.000 MHz, peripheral 50.000 MHz

S3C2410 Clocks, (c) 2004 Simtec Electronics

CLOCK: Slow mode (1.500 MHz), fast, MPLL on, UPLL on

CPU0: D VIVT write-back cache

CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets

CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets  
Built 1 zonelists  
Kernel command line: root=/dev/nfs nfsroot=192.168.253.1:/armsys2410/root ip=190irq: clearing  
subpending status 00000002  
PID hash table entries: 512 (order: 9, 8192 bytes)  
timer tcon=00500000, tcnt a2c1, tcfg 00000200,00000000, usec 00001eb8  
Console: colour dummy device 80x30  
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)  
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)  
Memory: 64MB = 64MB total  
Memory: 62128KB available (1996K code, 567K data, 108K init)  
Mount-cache hash table entries: 512  
CPU: Testing write buffer coherency: ok  
NET: Registered protocol family 16  
Now you use L70T84(800x480) to display!  
S3C2410: Initialising architecture  
SCSI subsystem initialized  
usbcore: registered new driver usbfs  
usbcore: registered new driver hub  
S3C2410 DMA Driver, (c) 2003-2004 Simtec Electronics  
DMA channel 0 at c4800000, irq 33  
DMA channel 1 at c4800040, irq 34  
DMA channel 2 at c4800080, irq 35  
DMA channel 3 at c48000c0, irq 36  
NetWinder Floating Point Emulator V0.97 (double precision)  
devfs: 2004-01-31 Richard Gooch (rgooch@atnf.csiro.au)  
devfs: devfs\_debug: 0x0  
devfs: boot\_options: 0x1  
yaffs Apr 6 2007 09:54:09 Installing.  
io scheduler noop registered  
Console: switching to colour frame buffer device 100x30  
fb0: s3c2410fb frame buffer device  
S3C2410 RTC, (c) 2004 Simtec Electronics  
s3c2410-rtc s3c2410-rtc: rtc disabled, re-enabling  
S3C2410 Watchdog Timer, (c) 2004 Simtec Electronics  
s3c2410-uart.0: s3c2410\_serial0 at MMIO 0x50000000 (irq = 70) is a S3C2410  
s3c2410-uart.1: s3c2410\_serial1 at MMIO 0x50004000 (irq = 73) is a S3C2410  
s3c2410-uart.2: s3c2410\_serial2 at MMIO 0x50008000 (irq = 76) is a S3C2410  
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize  
loop: loaded (max 8 devices)  
eth0: cs8900 rev K found at 0xf1300300  
cs89x0: Extended EEPROM checksum bad and no Cirrus EEPROM, relying on command lecs89x0  
media RJ-45, IRQ 53, programmed I/O, MAC 00:09:c0:ff:ec:48

```
cs89x0: request_region(0xf1300300, 0x10) failed
cs89x0: no cs8900 or cs8920 detected. Be sure to disable PnP with SETUP
S3C2410 NAND Driver, (c) 2004 Simtec Electronics
s3c2410-nand: mapped registers at c4a00000
s3c2410-nand: timing: Tacls 10ns, Twrph0 30ns, Twrph1 10ns
NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (Samsung NAND 64MiB 3,3V
8-bit)NAND_ECC_NONE selected by board driver. This is not recommended !!
Scanning device for bad blocks
Bad eraseblock 7 at 0x0001c000
Creating 3 MTD partitions on "NAND 64MiB 3,3V 8-bit":
0x00000000-0x00030000 : "boot"
0x00030000-0x00200000 : "kernel"
0x00200000-0x04000000 : "rootfs"
usbmon: debugfs is not available
Set S3C2410 UPLLCON!
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1
s3c2410-ohci s3c2410-ohci: irq 42, io mem 0x49000000
usb usb1: Product: S3C24XX OHCI
usb usb1: Manufacturer: Linux 2.6.15.1 ohci_hcd
usb usb1: SerialNumber: s3c24xx
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
mice: PS/2 mouse device common for all mice
ts: Compaq touchscreen protocol output
Register touchscreen driver for ARMSYS2410.
s3c2410 TouchScreen successfully loaded
input: s3c2410 TouchScreen as /class/input/input0
s3c2410-i2c s3c2410-i2c: slave address 0x10
s3c2410-i2c s3c2410-i2c: bus frequency set to 390 KHz
s3c2410-i2c s3c2410-i2c: i2c-0: S3C I2C adapter
SD driver init for ARMSYS2410.
mmci-s3c2410: probe: mapped sdi_base=c4e80000 irq=37 irq_cd=18 dma=0.
mmci-s3c2410: initialisation done.
UDA1341 audio driver initialized
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 2, 16384 bytes)
```

TCP bind hash table entries: 4096 (order: 2, 16384 bytes)  
TCP: Hash tables configured (established 4096 bind 4096)  
TCP reno registered  
TCP bic registered  
NET: Registered protocol family 1  
eth0: using half-duplex 10Base-T (RJ-45)

.....

## 4. NFS 方式 mount 主机

### 4. 1 在主机上启动 nfs 服务

启动 nfs 服务之前，必须准备好共享目录，也就是 2.4 节所说的 root 目录。假设该目录的路径为/armsys2410/root。

(1) 修改/etc/exports 文件的内容，添加一行：

```
/armsys2410/root *(rw, sync, no_root_squash)
```

**注意：root和\*之间有空格！**

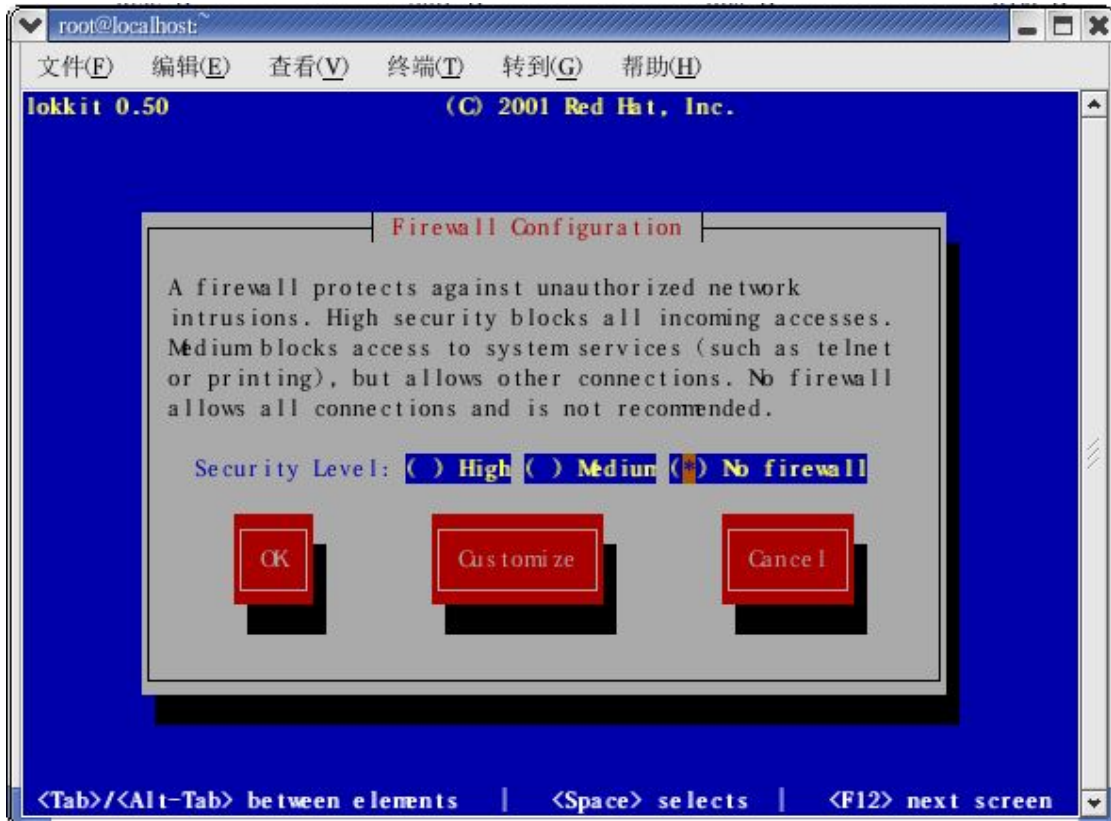
这里/armsys2410/root 代表要共享输出的 root 文件系统的共享目录；\*代表所有的客户机都可以挂接此文件系统；rw 代表客户机可以读写许可来挂接它们的根文件系统；no\_root\_squash 选项允许客户机以主机上的 root 身份挂接根文件系统。

(2) 关闭防火墙

终端下输入：

```
lokit
```

进入防火墙设置，将防火墙设置为无：



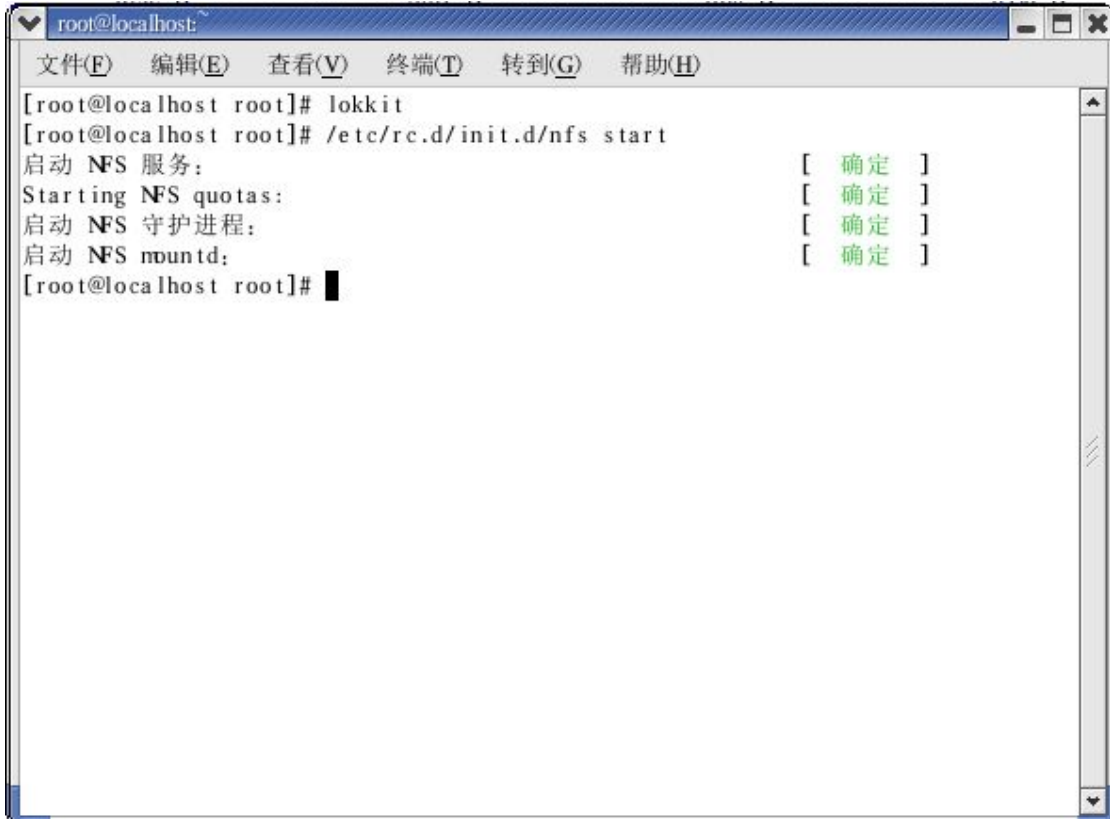
(3) 启动 NFS 服务:

输入命令行:

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

如果之前已经启动了 nfs, 则重新启动一次:

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



```
root@localhost:
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
[root@localhost root]# lokkit
[root@localhost root]# /etc/rc.d/init.d/nfs start
启动 NFS 服务: [ 确定 ]
Starting NFS quotas: [ 确定 ]
启动 NFS 守护进程: [ 确定 ]
启动 NFS mountd: [ 确定 ]
[root@localhost root]#
```

## 4. 2 在目标板上进行设定

### (1) 设定 Linux 的启动参数

按住 PC 机的空格键，然后复位开发板，此时从 minicom 的输出看到，进入了 u-boot 命令行方式。重新设置 Linux 的启动参数如下：

```
SMDK2410# setenv linux_arg root=/dev/nfs nfsroot=192.168.253.1:/armsys2410/root
ip=192.168.253.9:192.168.253.1:192.168.253.100:255.255.255.0:armsys.hzlitai.com:eth0:off
console=ttySAC0
SMDK2410# setenv bootcmd nandr c 140000 30008000\; bootm
```

命令行必须在一行中输入完。因此我们提供了一个 Linux2.6\start\_param 文本文件，其中已经写好了这两行，只要将它拷贝并粘贴到 SMDK2410#后即可。

注意，以上命令行假定主机地址为 192.168.253.1，ARMSYS2410 开发板的如果你的主机 IP 与此不同，请用 ifconfig 重新设定，或者修改该命令行的内容。

### (2) 保存参数

```
SMDK2410# saveenv
```

即可保存刚刚设定的参数，这样下一次就不需要重复输入上述命令行了。





```

root@localhost:~#
文件(E) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
eth0: cs8900 rev K(3.3 Volts) found at 0xd0000300
cs89x0 media RJ-45, IRQ 37
SCSI subsystem driver Revision: 1.00
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
UDA1341 audio driver initialized
NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (Samsung NAND 64MB 3,3V)
NAND_ECC_NONE selected by board driver. This is not recommended !!
Creating 3 MTD partitions on "NAND 64MB 3,3V":
0x00000000-0x00030000 : "boot"
0x00030000-0x00200000 : "kernel"
0x00200000-0x04000000 : "rootfs"
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
usb-ohci.c: USB OHCI at membase 0xe9000000, IRQ 26
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
usb.c: registered new driver usb_mouse

```

建立了 3 个 MTD 分区，分别用来存放 bootloader、kernel 和 rootfs。

这里，我们通过运行 root/bin/下的 Writerootfs 脚本文件，可以一次性完成 yaffs 根文件系统的下载和固化。

在 minicom 中回车，输入：

**[root@lyt /]#cd /**

**[root@lyt /]#Writerootfs**

显示结果如下：

```

root@192:~#
文件(E) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
root@192:/armsys2413/linux-2.6.15 root@192:~
[lyt /]# Writerootfs
=====
=          format.....          =
=====
argc 2 sh 0 optent 1
(1)Open flash.
(2)Get Info from flash.
Erasing and programming NAND
CK
=====
=      mount to yaffs part .....      =
=====
yaffs: dev is 32505858 name is "mtdblock2"
yaffs: Attempting MTD mount on 31.2, "mtdblock2"
=====
=      decompress file          =
=====
=====
=      Root FS has been updated.      =
=      Reboot and Enjoy ARMSYS2410!  =
=====

```

上图中的过程是以 YAFFS 作为文件系统格式化分区。等待约 5 分钟，直到出现：

```
=====
=          Root FS has been updated.          =
=          Reboot and Enjoy ARMSYS2410!       =
=====
```

说明已经完成了 root 文件系统的更新，下面我们就可以从更新的文件系统启动了。

## 5. 2 从 YAFFS 文件系统启动

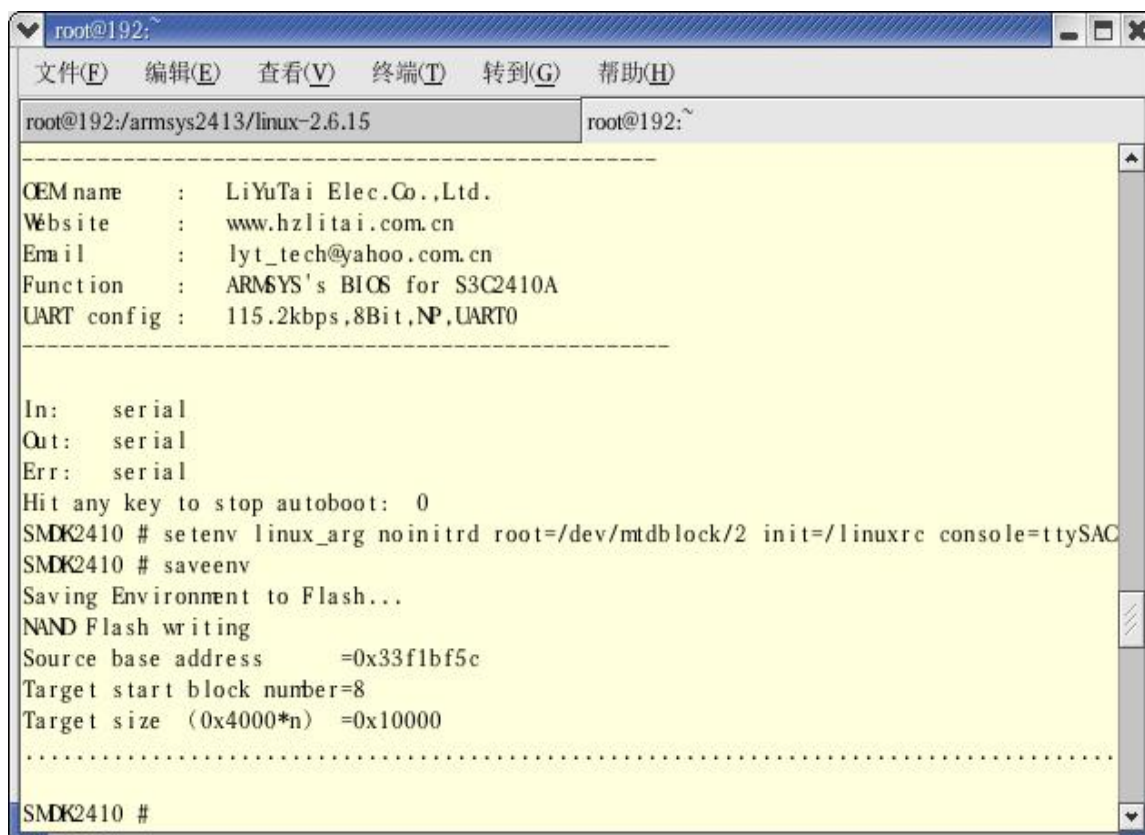
复位开发板，重新设置系统启动参数，同样可以从 Linux2.6/start\_param 文本文件中拷贝获得：

```
SMDK2410# Setenv linux_arg root=/dev/mtdblock/2 init=/linuxrc console=ttySAC0
```

这里我们从 mtdblock/2 分区启动 root 文件系统。

可以用以下命令行保存该设置：

```
SMDK2410# saveenv
```



为了验证能否成功启动文件系统，输入：

### **SMDK2410# boot**

启动信息如下：

```
SMDK2410 # boot
zImage magic = 0x016f2818
NOW, Booting Linux.....
Setup linux parameters at 0x30000100
linux command line is: "noinitrd root=/dev/mtdblock/2 init=/linuxrc console=tty"Uncompressing
Linux.....Linux version 2.6.15.1 (root@192.168.253.100) (gcc
version 3.4.3 (release) (Cod7CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
Machine: SMDK2410
ATAG_INITRD is deprecated; please update your bootloader.
Memory policy: ECC disabled, Data cache writeback
CPU S3C2410A (id 0x32410002)
S3C2410: core 200.000 MHz, memory 100.000 MHz, peripheral 50.000 MHz
S3C2410 Clocks, (c) 2004 Simtec Electronics
CLOCK: Slow mode (1.500 MHz), fast, MPLL on, UPLL on
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Built 1 zonelists
Kernel command line: noinitrd root=/dev/mtdblock/2 init=/linuxrc console=ttySAC0irq: clearing
subpending status 00000002
PID hash table entries: 512 (order: 9, 8192 bytes)
timer tcon=00500000, tcnt a2c1, tcfg 00000200,00000000, usec 00001eb8
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 64MB = 64MB total
Memory: 62128KB available (1996K code, 567K data, 108K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
Now you use L70T84(800x480) to display!
S3C2410: Initialising architecture
SCSI subsystem initialized
usbcore: registered new driver usbfs
usbcore: registered new driver hub
S3C2410 DMA Driver, (c) 2003-2004 Simtec Electronics
```

DMA channel 0 at c4800000, irq 33  
DMA channel 1 at c4800040, irq 34  
DMA channel 2 at c4800080, irq 35  
DMA channel 3 at c48000c0, irq 36  
NetWinder Floating Point Emulator V0.97 (double precision)  
devfs: 2004-01-31 Richard Gooch (rgooch@atnf.csiro.au)  
devfs: devfs\_debug: 0x0  
devfs: boot\_options: 0x1  
yaffs Apr 6 2007 09:54:09 Installing.  
io scheduler noop registered  
Console: switching to colour frame buffer device 100x30  
fb0: s3c2410fb frame buffer device  
S3C2410 RTC, (c) 2004 Simtec Electronics  
s3c2410-rtc s3c2410-rtc: rtc disabled, re-enabling  
S3C2410 Watchdog Timer, (c) 2004 Simtec Electronics  
s3c2410-uart.0: s3c2410\_serial0 at MMIO 0x50000000 (irq = 70) is a S3C2410  
s3c2410-uart.1: s3c2410\_serial1 at MMIO 0x50004000 (irq = 73) is a S3C2410  
s3c2410-uart.2: s3c2410\_serial2 at MMIO 0x50008000 (irq = 76) is a S3C2410  
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize  
loop: loaded (max 8 devices)  
eth0: cs8900 rev K found at 0xf1300300  
cs89x0: Extended EEPROM checksum bad and no Cirrus EEPROM, relying on command  
lecs89x0 media RJ-45, IRQ 53, programmed I/O, MAC 00:09:c0:ff:ec:48  
cs89x0: request\_region(0xf1300300, 0x10) failed  
cs89x0: no cs8900 or cs8920 detected. Be sure to disable PnP with SETUP  
S3C2410 NAND Driver, (c) 2004 Simtec Electronics  
s3c2410-nand: mapped registers at c4a00000  
s3c2410-nand: timing: Tacls 10ns, Twrph0 30ns, Twrph1 10ns  
NAND device: Manufacturer ID: 0xec, Chip ID: 0x76 (Samsung NAND 64MiB 3,3V  
8-bi)NAND\_ECC\_NONE selected by board driver. This is not recommended !!  
Scanning device for bad blocks  
Bad eraseblock 7 at 0x0001c000  
Creating 3 MTD partitions on "NAND 64MiB 3,3V 8-bit":  
0x00000000-0x00030000 : "boot"  
0x00030000-0x00200000 : "kernel"  
0x00200000-0x04000000 : "rootfs"  
usbmon: debugfs is not available  
Set S3C2410 UPLLCON!  
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI  
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1  
s3c2410-ohci s3c2410-ohci: irq 42, io mem 0x49000000  
usb usb1: Product: S3C24XX OHCI  
usb usb1: Manufacturer: Linux 2.6.15.1 ohci\_hcd



```
tv.tv_sec=-1
```

Please press Enter to activate this console.

以后，即使开发板掉电，root 文件系统仍旧保存。由于采用了可读写的 YAFFS 文件系统，你所保存的文件和设置在掉电后都不会丢失。