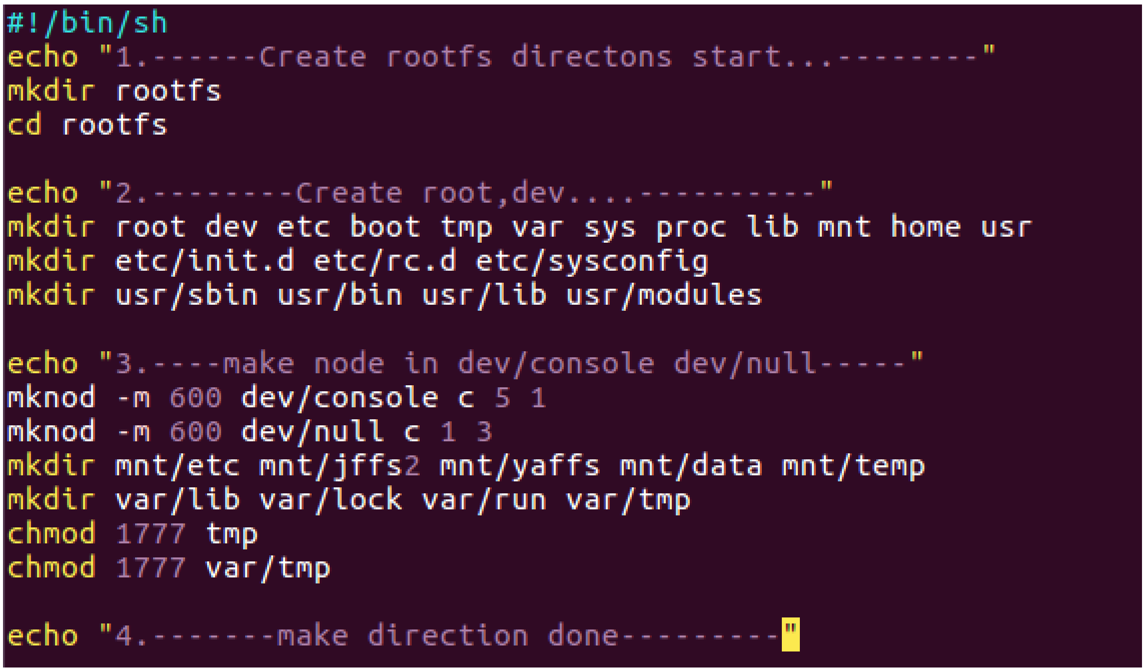
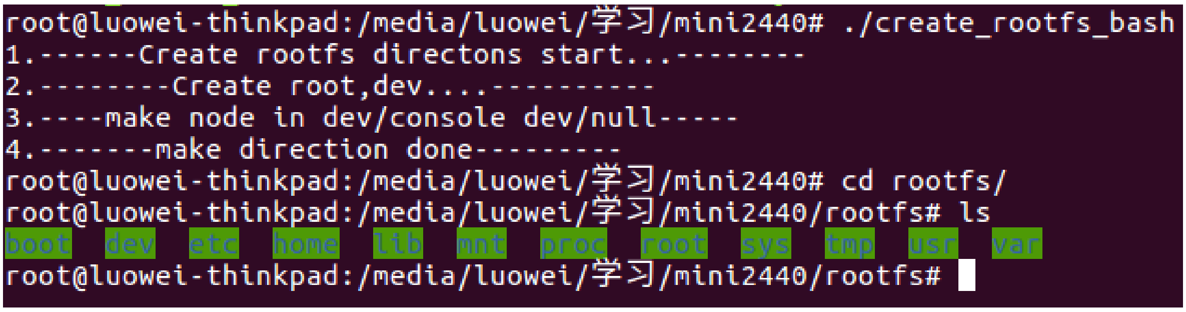
一、创建基本的目录和设备文件，拷贝动态库文件

1.新建如下图的脚本(批处理)创建根文件系统目录，并创建了两个设备文件 console 和 null



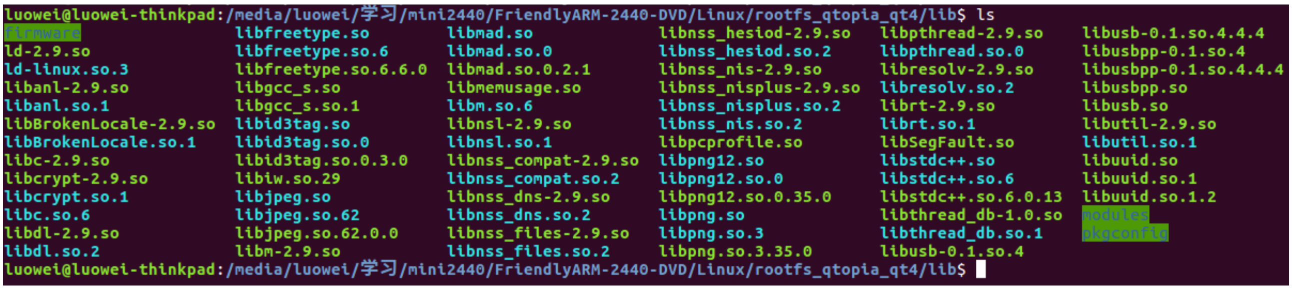
```
#!/bin/sh
echo "1.------Create rootfs directons start...--------"
mkdir rootfs
cd rootfs

echo "2.--------Create root,dev....----------"
mkdir root dev etc boot tmp var sys proc lib mnt home usr
mkdir etc/init.d etc/rc.d etc/sysconfig
mkdir usr/sbin usr/bin usr/lib usr/modules

echo "3.----make node in dev/console dev/null-----"
mknod -m 600 dev/console c 5 1
mknod -m 600 dev/null c 1 3
mkdir mnt/etc mnt/jffs2 mnt/yaffs mnt/data mnt/temp
mkdir var/lib var/lock var/run var/tmp
chmod 1777 tmp
chmod 1777 var/tmp

echo "4.-------make direction done---------"
```

2.运行脚本得到 rootfs 根目录文件夹

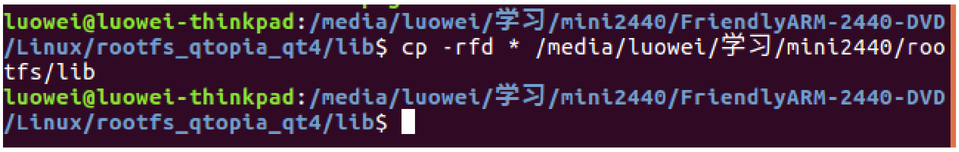


```
root@luowei-thinkpad:/media/luowei/学习/mini2440# ./create_rootfs_bash
1.------Create rootfs directons start...--------
2.--------Create root,dev....----------
3.----make node in dev/console dev/null-----
4.-------make direction done---------
root@luowei-thinkpad:/media/luowei/学习/mini2440# cd rootfs/
root@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs# ls
boot  dev  etc  home  lib  mnt  proc  root  sys  tmp  usr  var
root@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs# 
```

3.动态链接库直接用友善之臂的，即将已经移植好的/lib 文件夹下的所有文件拷贝到新的根文件系统/lib 中



```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/FriendlyARM-2440-DVD/Linux/rootfs_qtopia_qt4/lib$ ls
firmware                libfreetype.so        libmad.so             libnss_hesiod-2.9.so   libpthread-2.9.so    libusb-0.1.so.4.4.4
ld-2.9.so               libfreetype.so.6      libmad.so.0           libnss_hesiod.so.2     libpthread.so.0      libusbpp-0.1.so.4
ld-linux.so.3           libfreetype.so.6.6.0  libmad.so.0.2.1       libnss_nis-2.9.so      libresolv-2.9.so     libusbpp-0.1.so.4.4.4
libanl-2.9.so           libgcc_s.so           libmemusage.so        libnss_nisplus-2.9.so  libresolv.so.2       libusbpp.so
libanl.so.1             libgcc_s.so.1         libm.so.6             libnss_nisplus.so.2    librt-2.9.so         libusb.so
libBrokenLocale-2.9.so  libid3tag.so          libnsl-2.9.so         libnss_nis.so.2        librt.so.1           libutil-2.9.so
libBrokenLocale.so.1    libid3tag.so.0        libnsl.so.1           libpcprofile.so        libSegFault.so       libutil.so.1
libc-2.9.so             libid3tag.so.0.3.0    libnss_compat-2.9.so  libpng12.so            libstdc++.so         libuuid.so
libcrypt-2.9.so         libiw.so.29           libnss_compat.so.2    libpng12.so.0          libstdc++.so.6       libuuid.so.1
libcrypt.so.1           libjpeg.so            libnss_dns-2.9.so     libpng12.so.0.35.0     libstdc++.so.6.0.13  libuuid.so.1.2
libc.so.6               libjpeg.so.62         libnss_dns.so.2       libpng.so              libthread_db-1.0.so  modules
libdl-2.9.so            libjpeg.so.62.0.0     libnss_files-2.9.so   libpng.so.3            libthread_db.so.1    pkgconfig
libdl.so.2              libm-2.9.so           libnss_files.so.2     libpng.so.3.35.0       libusb-0.1.so.4
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/FriendlyARM-2440-DVD/Linux/rootfs_qtopia_qt4/lib$ 
```

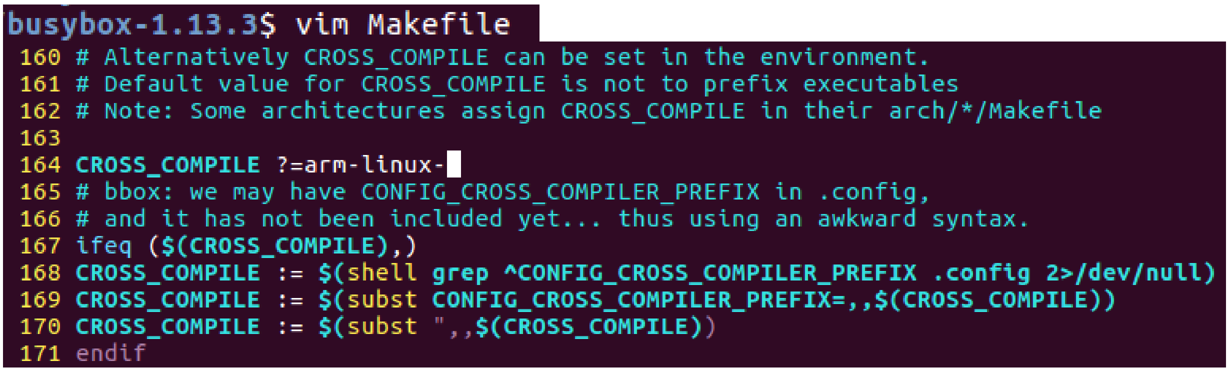


```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/FriendlyARM-2440-DVD
/Linux/rootfs_qtopia_qt4/lib$ cp -rfd * /media/luowei/学习/mini2440/roo
tfs/lib
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/FriendlyARM-2440-DVD
/Linux/rootfs_qtopia_qt4/lib$ 
```

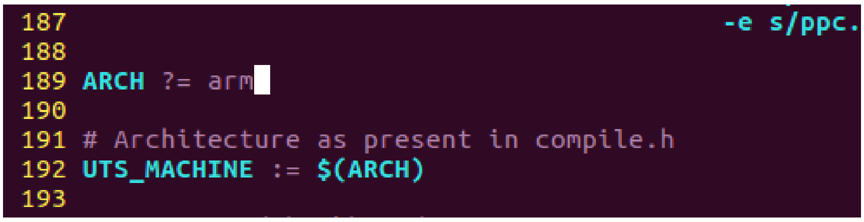
二、利用 busybox 软件产生命令文件和初始化程序

Bosybox 是一个遵循 GPL v2 协议的开源项目，它在编写过程总对文件大小进行优化,，并考虑了系统资源有限(比如内存等)的情况，使用 Busybox 可以自动生成根文件系统所需的 bin、usr/bin、usr/sbin 和 linuxrc 文件

1.修改主 Makefile（下面几部和内核的配置编译类似）

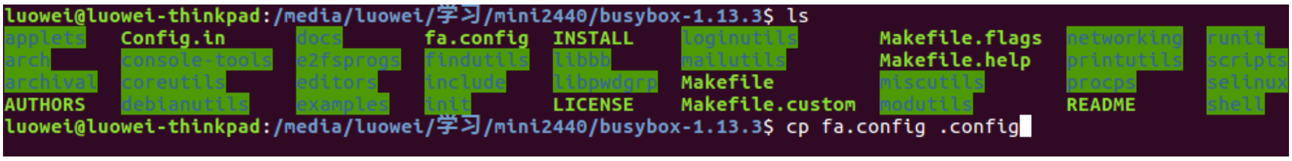


```
busybox-1.13.3$ vim Makefile
160 # Alternatively CROSS_COMPILE can be set in the environment.
161 # Default value for CROSS_COMPILE is not to prefix executables
162 # Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile
163
164 CROSS_COMPILE ?=arm-linux-
165 # bbox: we may have CONFIG_CROSS_COMPILER_PREFIX in .config,
166 # and it has not been included yet... thus using an awkward syntax.
167 ifeq ($(CROSS_COMPILE),)
168 CROSS_COMPILE := $(shell grep ^CONFIG_CROSS_COMPILER_PREFIX .config 2>/dev/null)
169 CROSS_COMPILE := $(subst CONFIG_CROSS_COMPILER_PREFIX=,,$(CROSS_COMPILE))
170 CROSS_COMPILE := $(subst ",,$(CROSS_COMPILE))
171 endif
```



```
187                                                    -e s/ppc.
188
189 ARCH ?= arm
190
191 # Architecture as present in compile.h
192 UTS_MACHINE := $(ARCH)
193
```

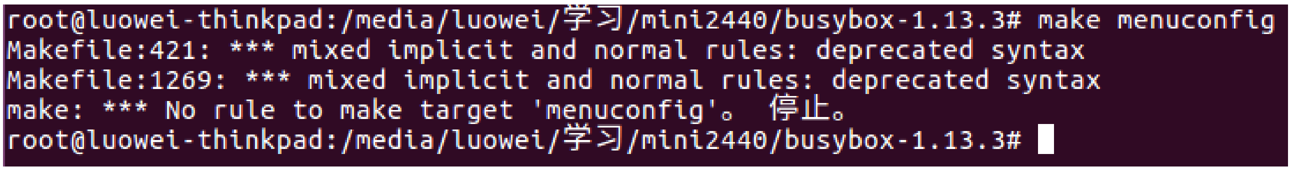
2.复制配置文件：**cp fa.config .config**



```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/busybox-1.13.3$ ls
applets   Config.in      docs       fa.config  INSTALL    loginutils       Makefile.flags  networking  runit
arch      console-tools  e2fsprogs  findutils  libbb      mailutils        Makefile.help   printutils  scripts
archival  coreutils      editors    include    libpwdgrp  Makefile         miscutils       procps      selinux
AUTHORS   debianutils    examples   init       LICENSE    Makefile.custom  modutils        README      shell
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/busybox-1.13.3$ cp fa.config .config
```

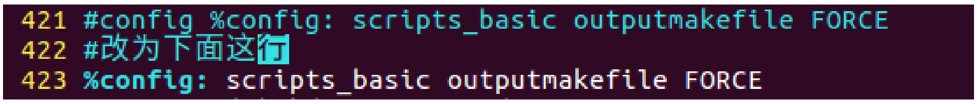
3.make menuconfig（上一步复制完配置文件已经完成了所有配置，不再需要 make menuconfig 做任何修改）

但是试了一下发现 make menuconfig 会报错：

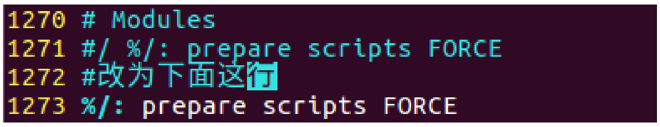


```
root@luowei-thinkpad:/media/luowei/学习/mini2440/busybox-1.13.3# make menuconfig
Makefile:421: *** mixed implicit and normal rules: deprecated syntax
Makefile:1269: *** mixed implicit and normal rules: deprecated syntax
make: *** No rule to make target 'menuconfig'。 停止。
root@luowei-thinkpad:/media/luowei/学习/mini2440/busybox-1.13.3# 
```

修改之后可以才可以正常进入 make menuconfig

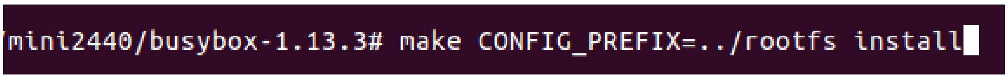


```
421 #config %config: scripts_basic outputmakefile FORCE
422 #改为下面这行
423 %config: scripts_basic outputmakefile FORCE
```



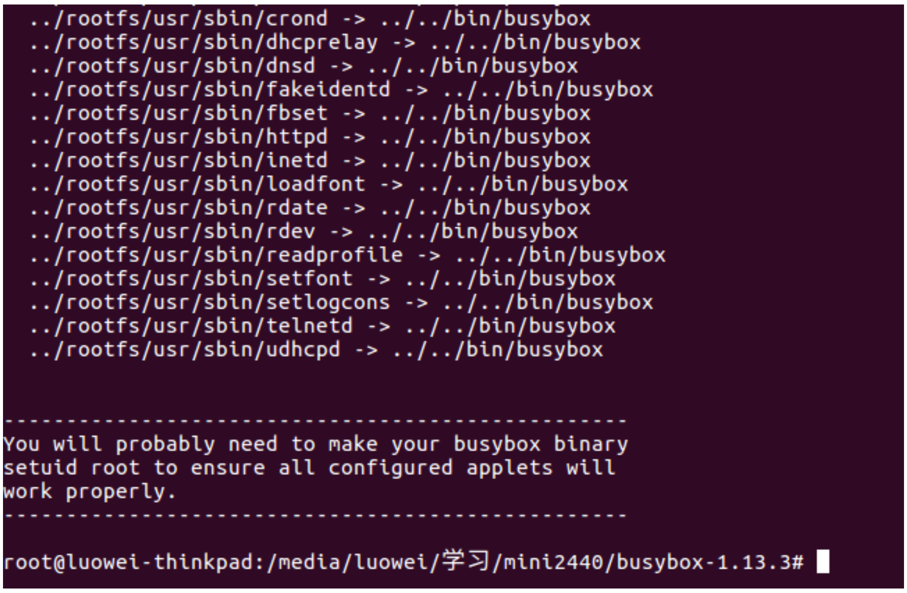
```
1270 # Modules
1271 #/ %/: prepare scripts FORCE
1272 #改为下面这行
1273 %/: prepare scripts FORCE
```

4.编译并安装 busybox：**make CONFIG_PREFIX=[文件系统根目录] install**



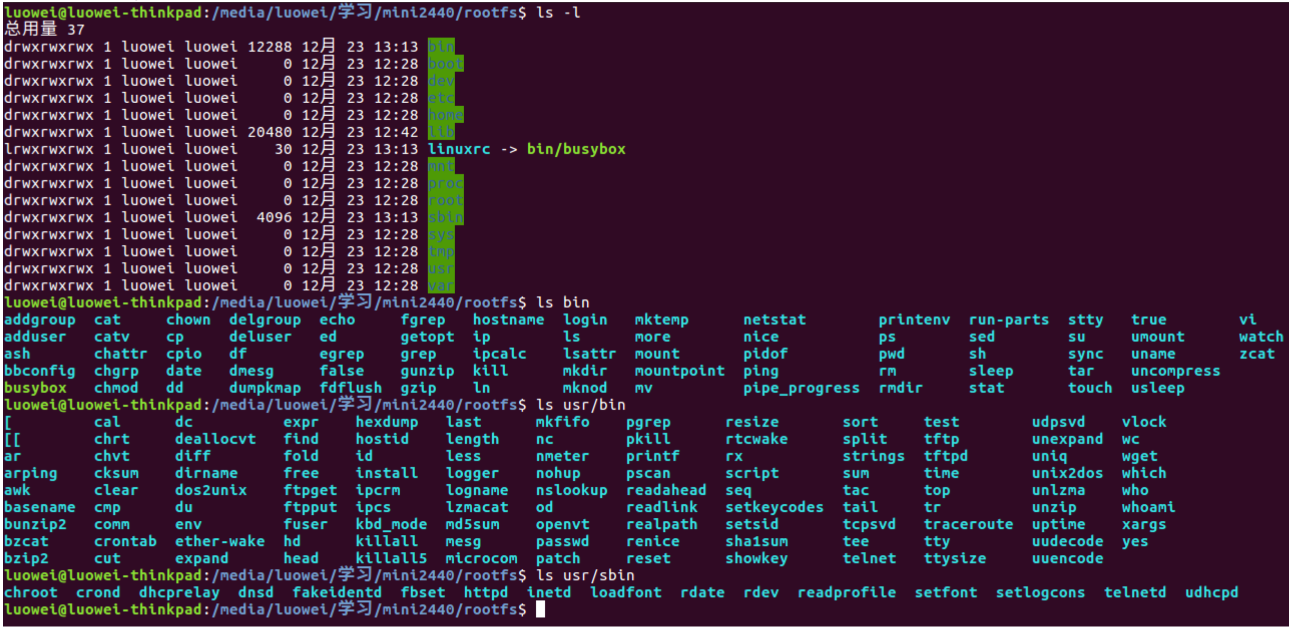
```
mini2440/busybox-1.13.3# make CONFIG_PREFIX=../rootfs install
```

产生如下结果表示编译和安装成功：



```
 ../rootfs/usr/sbin/crond -> ../../bin/busybox
 ../rootfs/usr/sbin/dhcprelay -> ../../bin/busybox
 ../rootfs/usr/sbin/dnsd -> ../../bin/busybox
 ../rootfs/usr/sbin/fakeidentd -> ../../bin/busybox
 ../rootfs/usr/sbin/fbset -> ../../bin/busybox
 ../rootfs/usr/sbin/httpd -> ../../bin/busybox
 ../rootfs/usr/sbin/inetd -> ../../bin/busybox
 ../rootfs/usr/sbin/loadfont -> ../../bin/busybox
 ../rootfs/usr/sbin/rdate -> ../../bin/busybox
 ../rootfs/usr/sbin/rdev -> ../../bin/busybox
 ../rootfs/usr/sbin/readprofile -> ../../bin/busybox
 ../rootfs/usr/sbin/setfont -> ../../bin/busybox
 ../rootfs/usr/sbin/setlogcons -> ../../bin/busybox
 ../rootfs/usr/sbin/telnetd -> ../../bin/busybox
 ../rootfs/usr/sbin/udhcpd -> ../../bin/busybox


--------------------------------------------------
You will probably need to make your busybox binary
setuid root to ensure all configured applets will
work properly.
--------------------------------------------------

root@luowei-thinkpad:/media/luowei/学习/mini2440/busybox-1.13.3# 
```

5.查看 busybox 在 rootfs 目录产生的 bin、sbin 目录下的指令和 linuxrc 等文件：



```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs$ ls -l
总用量 37
drwxrwxrwx 1 luowei luowei 12288 12月 23 13:13 bin
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 boot
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 dev
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 etc
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 home
drwxrwxrwx 1 luowei luowei 20480 12月 23 12:42 lib
lrwxrwxrwx 1 luowei luowei    30 12月 23 13:13 linuxrc -> bin/busybox
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 mnt
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 proc
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 root
drwxrwxrwx 1 luowei luowei  4096 12月 23 13:13 sbin
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 sys
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 tmp
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 usr
drwxrwxrwx 1 luowei luowei     0 12月 23 12:28 var
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs$ ls bin
addgroup  cat     chown  delgroup  echo     fgrep   hostname  login   mktemp      netstat        printenv  run-parts  stty   true        vi
adduser   catv    cp     deluser   ed       getopt  ip        ls      more        nice           ps        sed        su     umount      watch
ash       chattr  cpio   df        egrep    grep    ipcalc    lsattr  mount       pidof          pwd       sh         sync   uname       zcat
bbconfig  chgrp   date   dmesg     false    gunzip  kill      mkdir   mountpoint  ping           rm        sleep      tar    uncompress
busybox   chmod   dd     dumpkmap  fdflush  gzip    ln        mknod   mv          pipe_progress  rmdir     stat       touch  usleep
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs$ ls usr/bin
[         cal      dc          expr    hexdump   last      mkfifo    pgrep      resize       sort     test        udpsvd    vlock
[[        chrt     deallocvt   find    hostid    length    nc        pkill      rtcwake      split    tftp        unexpand  wc
ar        chvt     diff        fold    id        less      nmeter    printf     rx           strings  tftpd       uniq      wget
arping    cksum    dirname     free    install   logger    nohup     pscan      script       sum      time        unix2dos  which
awk       clear    dos2unix    ftpget  ipcrm     logname   nslookup  readahead  seq          tac      top         unlzma    who
basename  cmp      du          ftpput  ipcs      lzmacat   od        readlink   setkeycodes  tail     tr          unzip     whoami
bunzip2   comm     env         fuser   kbd_mode  md5sum    openvt    realpath   setsid       tcpsvd   traceroute  uptime    xargs
bzcat     crontab  ether-wake  hd      killall   mesg      passwd    renice     sha1sum      tee      tty         uudecode  yes
bzip2     cut      expand      head    killall5  microcom  patch     reset      showkey      telnet   ttysize     uuencode
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs$ ls usr/sbin
chroot  crond  dhcprelay  dnsd  fakeidentd  fbset  httpd  inetd  loadfont  rdate  rdev  readprofile  setfont  setlogcons  telnetd  udhcpd
luowei@luowei-thinkpad:/media/luowei/学习/mini2440/rootfs$ 
```

三、手动创建 etc 目录下的配置文件

1. etc/mdev.conf 文件：内容为空
2. 从主机或已经移植好的文件系统拷贝 etc/passwd etc/group etc/shadow
3. etc/sysconfig/HOSTNAME 文件: luowei
4. etc/inittab 文件:

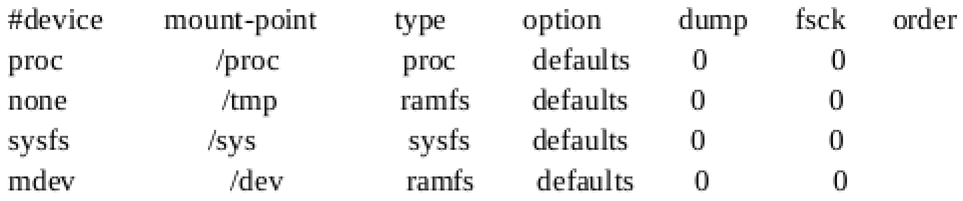
::sysinit:/etc/init.d/rcS
s3c2410_serial0::askfirst:-/bin/sh
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a –r

5. etc/init.d/rcS 文件：
#!/bin/sh

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
runlevel=S
prevlevel=N
umask 022
export PATH runlevel prevlevel

mount -a

echo /sbin/mdev>/proc/sys/kernel/hotplug
mdev -s

echo 'Start up successfully!'

#/bin/hostname -F /etc/sysconfig/HOSTNAME
 /bin/hostname luowei
```

使用以下命令改变 rcS 的执行权限：
Chmod 777 rcS

6. etc/fstab 文件：



```
#device      mount-point     type     option      dump    fsck    order
proc              /proc       proc     defaults     0        0
none              /tmp        ramfs    defaults     0        0
sysfs            /sys          sysfs   defaults     0        0
mdev               /dev        ramfs    defaults     0        0
```

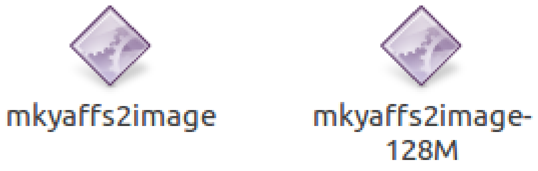
7. etc/profile 文件：

```
# Ash profile
# vim: syntax=sh

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

USER="`id -un`"
LOGNAME=$USER
PS1='[\u@\h \W]\# '  这句是登录后显示的[root@luowei pwd]#
PATH=$PATH

HOSTNAME=`/bin/hostname`

export USER LOGNAME PS1 PATH
```

四、制作根文件目录镜像

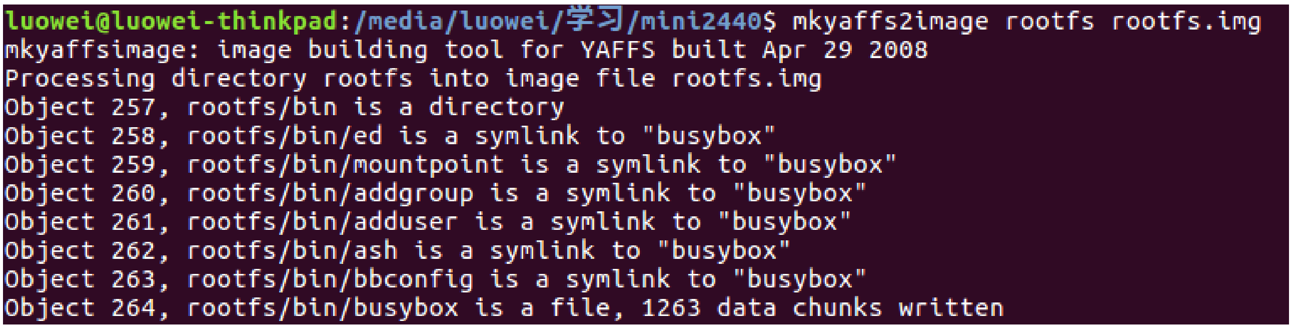
1.解压制作工具 mkyaffs2image.tgz

解压得到

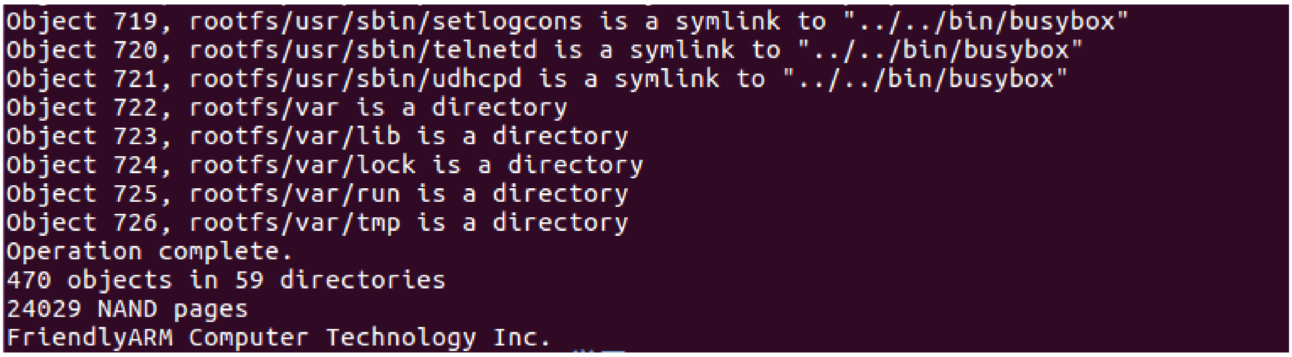


放入/usr/sbin/目录方便这两条命令可以正常识别。

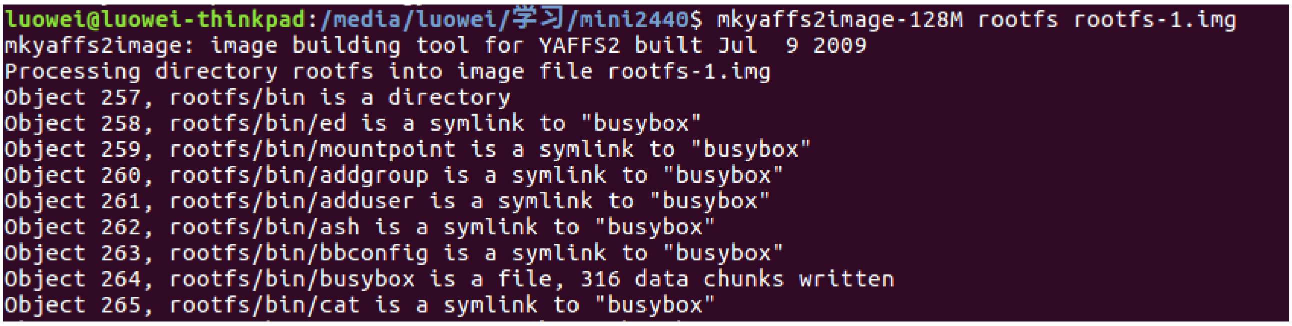
2.制作根文件目录镜像：**mkyaffs2iamge-128M rootfs xxx.img**
以下是两条命令各自产生生的镜像



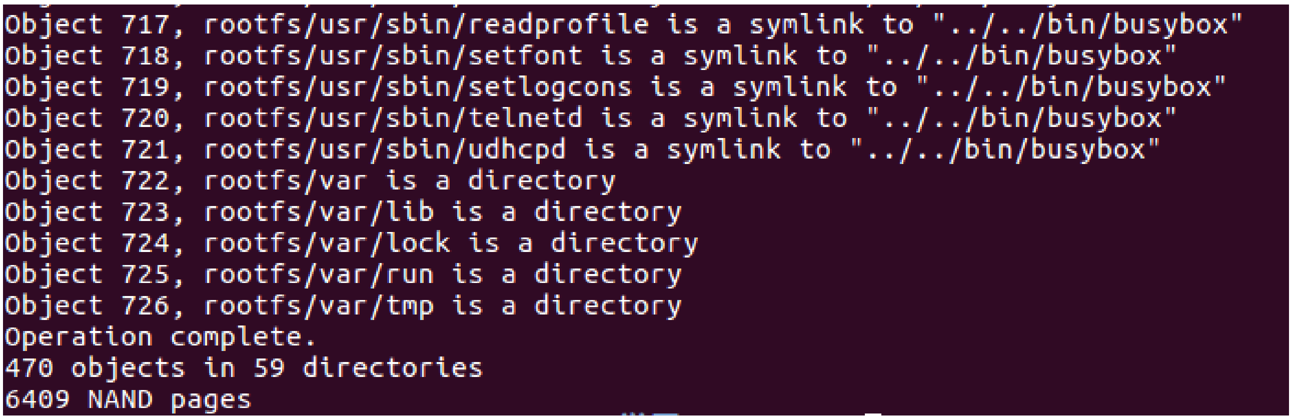
```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440$ mkyaffs2image rootfs rootfs.img
mkyaffsimage: image building tool for YAFFS built Apr 29 2008
Processing directory rootfs into image file rootfs.img
Object 257, rootfs/bin is a directory
Object 258, rootfs/bin/ed is a symlink to "busybox"
Object 259, rootfs/bin/mountpoint is a symlink to "busybox"
Object 260, rootfs/bin/addgroup is a symlink to "busybox"
Object 261, rootfs/bin/adduser is a symlink to "busybox"
Object 262, rootfs/bin/ash is a symlink to "busybox"
Object 263, rootfs/bin/bbconfig is a symlink to "busybox"
Object 264, rootfs/bin/busybox is a file, 1263 data chunks written
```



```
Object 719, rootfs/usr/sbin/setlogcons is a symlink to "../../bin/busybox"
Object 720, rootfs/usr/sbin/telnetd is a symlink to "../../bin/busybox"
Object 721, rootfs/usr/sbin/udhcpd is a symlink to "../../bin/busybox"
Object 722, rootfs/var is a directory
Object 723, rootfs/var/lib is a directory
Object 724, rootfs/var/lock is a directory
Object 725, rootfs/var/run is a directory
Object 726, rootfs/var/tmp is a directory
Operation complete.
470 objects in 59 directories
24029 NAND pages
FriendlyARM Computer Technology Inc.
```

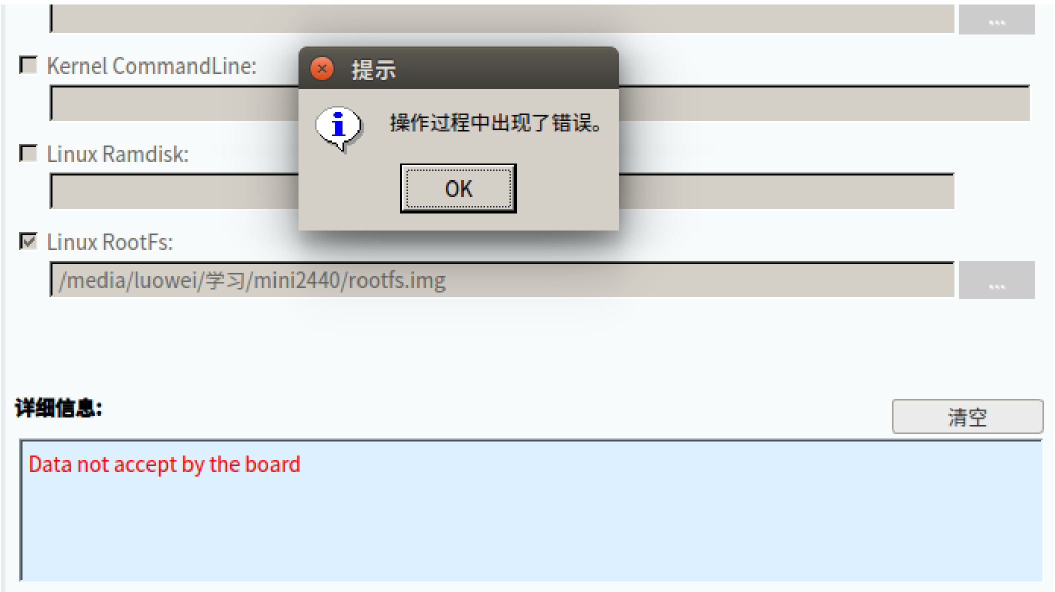


```
luowei@luowei-thinkpad:/media/luowei/学习/mini2440$ mkyaffs2image-128M rootfs rootfs-1.img
mkyaffs2image: image building tool for YAFFS2 built Jul  9 2009
Processing directory rootfs into image file rootfs-1.img
Object 257, rootfs/bin is a directory
Object 258, rootfs/bin/ed is a symlink to "busybox"
Object 259, rootfs/bin/mountpoint is a symlink to "busybox"
Object 260, rootfs/bin/addgroup is a symlink to "busybox"
Object 261, rootfs/bin/adduser is a symlink to "busybox"
Object 262, rootfs/bin/ash is a symlink to "busybox"
Object 263, rootfs/bin/bbconfig is a symlink to "busybox"
Object 264, rootfs/bin/busybox is a file, 316 data chunks written
Object 265, rootfs/bin/cat is a symlink to "busybox"
```



```
Object 717, rootfs/usr/sbin/readprofile is a symlink to "../../bin/busybox"
Object 718, rootfs/usr/sbin/setfont is a symlink to "../../bin/busybox"
Object 719, rootfs/usr/sbin/setlogcons is a symlink to "../../bin/busybox"
Object 720, rootfs/usr/sbin/telnetd is a symlink to "../../bin/busybox"
Object 721, rootfs/usr/sbin/udhcpd is a symlink to "../../bin/busybox"
Object 722, rootfs/var is a directory
Object 723, rootfs/var/lib is a directory
Object 724, rootfs/var/lock is a directory
Object 725, rootfs/var/run is a directory
Object 726, rootfs/var/tmp is a directory
Operation complete.
470 objects in 59 directories
6409 NAND pages
```

3. 使用 minitools 进行烧录根文件镜像：



☐ Kernel CommandLine:

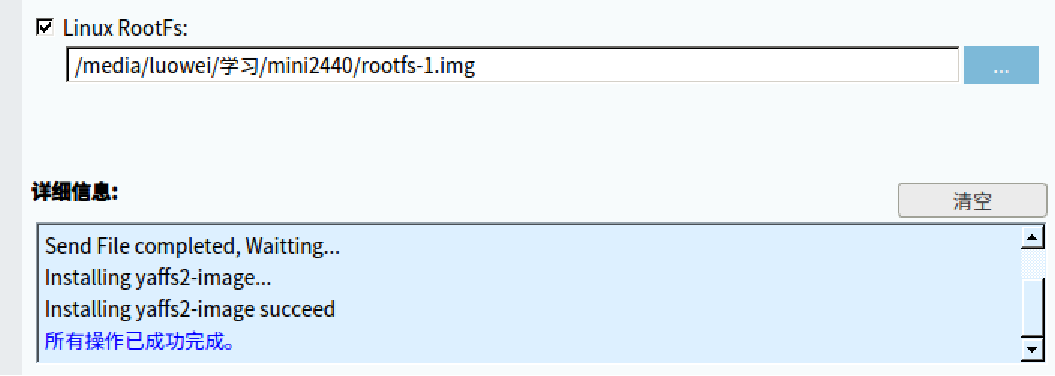
☐ Linux Ramdisk:

☑ Linux RootFs:
/media/luowei/学习/mini2440/rootfs.img

操作过程中出现了错误。

详细信息:

Data not accept by the board



☑ Linux RootFs:
/media/luowei/学习/mini2440/rootfs-1.img

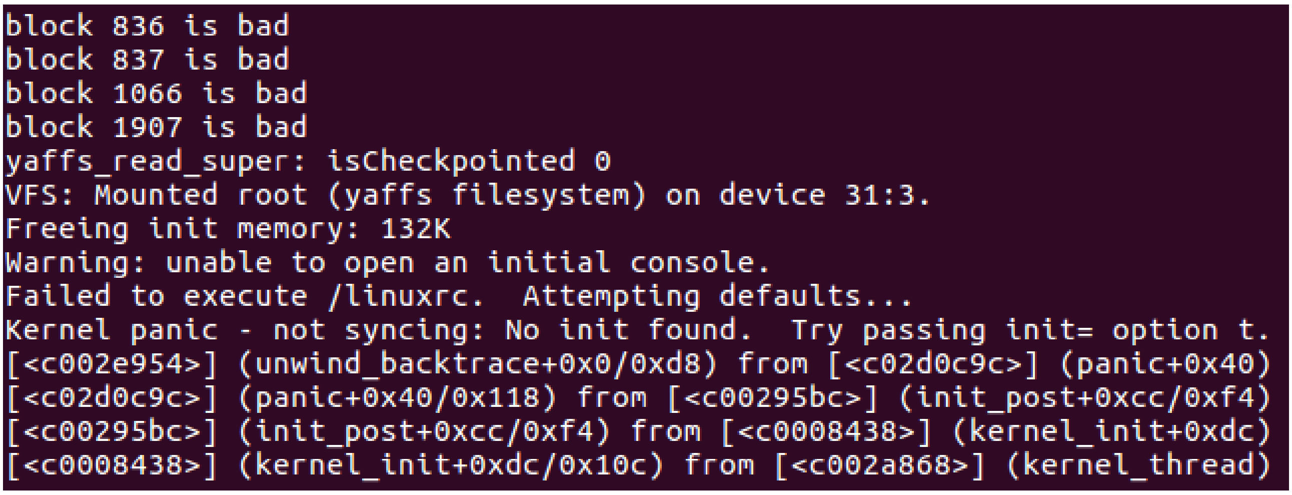
详细信息:

```
Send File completed, Waitting...
Installing yaffs2-image...
Installing yaffs2-image succeed
所有操作已成功完成。
```

由此可见 mkyaffs2image 存在问题，产生的镜像无法烧录。只能采用 **mkyaffs2image-128M** 制作镜像。并且发现：

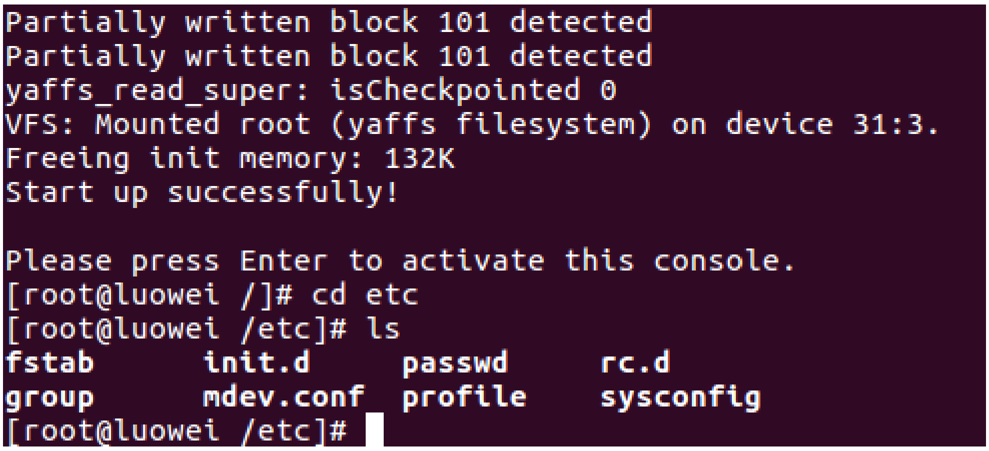
如果把 dev/目录下的两个设置文件删掉，mkyaffs2image 制作的镜像就可以成功烧录了，但是烧录后缺少设备文件 console 是无法正常登录。

如果缺少 dev/下的设备文件报错如下：不能打开 console



```
block 836 is bad
block 837 is bad
block 1066 is bad
block 1907 is bad
yaffs_read_super: isCheckpointed 0
VFS: Mounted root (yaffs filesystem) on device 31:3.
Freeing init memory: 132K
Warning: unable to open an initial console.
Failed to execute /linuxrc.  Attempting defaults...
Kernel panic - not syncing: No init found.  Try passing init= option t.
[<c002e954>] (unwind_backtrace+0x0/0xd8) from [<c02d0c9c>] (panic+0x40)
[<c02d0c9c>] (panic+0x40/0x118) from [<c00295bc>] (init_post+0xcc/0xf4)
[<c00295bc>] (init_post+0xcc/0xf4) from [<c0008438>] (kernel_init+0xdc)
[<c0008438>] (kernel_init+0xdc/0x10c) from [<c002a868>] (kernel_thread)
```

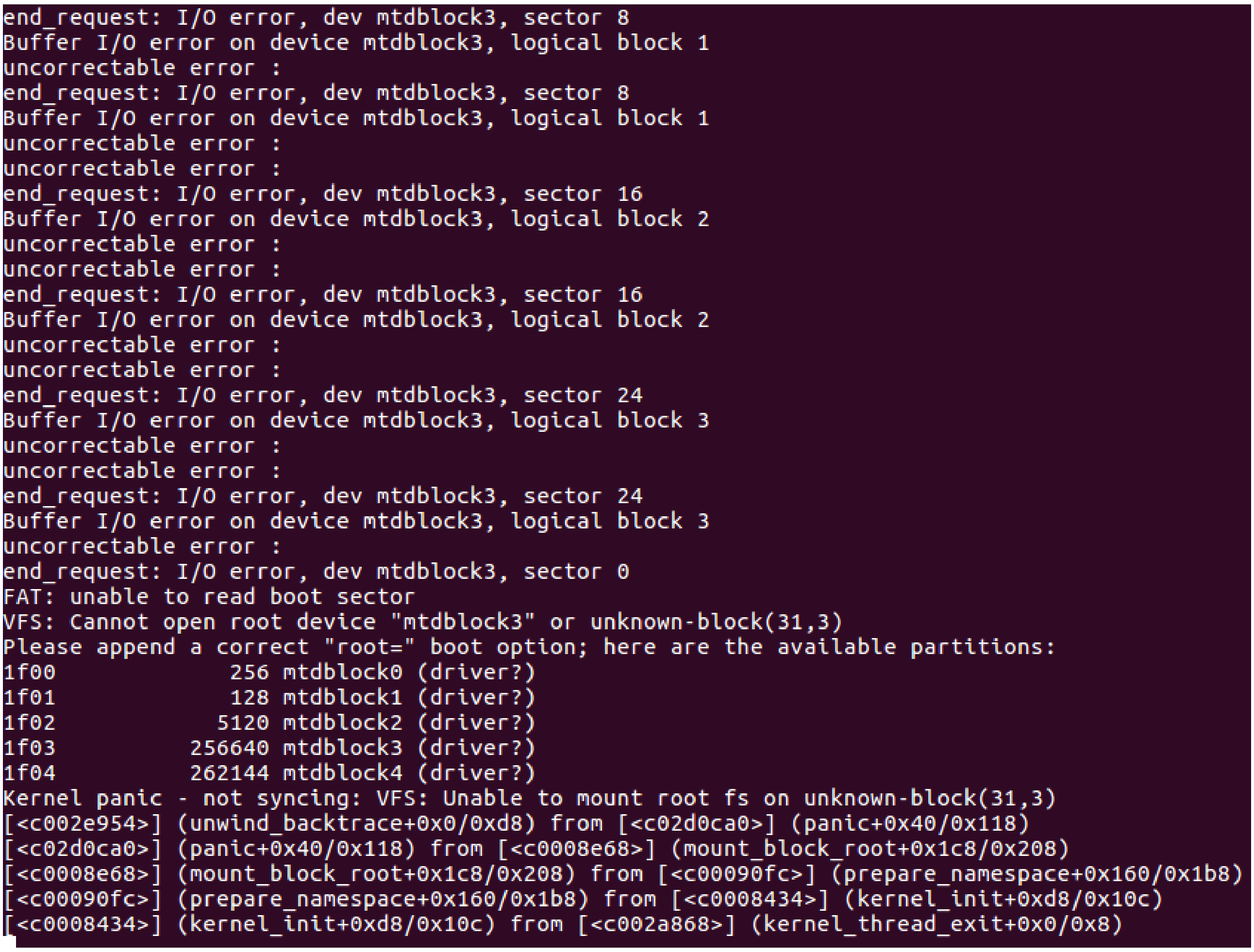
4.开机登录



```
Partially written block 101 detected
Partially written block 101 detected
yaffs_read_super: isCheckpointed 0
VFS: Mounted root (yaffs filesystem) on device 31:3.
Freeing init memory: 132K
Start up successfully!

Please press Enter to activate this console.
[root@luowei /]# cd etc
[root@luowei /etc]# ls
fstab       init.d      passwd      rc.d
group       mdev.conf   profile     sysconfig
[root@luowei /etc]#
```

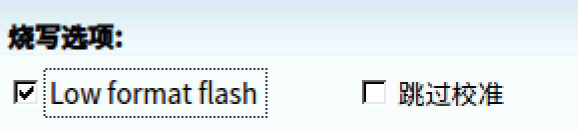
五、开机过程中可能会遇到的问题

如果出现如下问题，可能是 ECC 校验问题，内核编译启用了 MTD 的 ECC，文件系统制作过程禁用了的 ECC，将内核源码 drivers/mtd/nand/s3c2410.c 中的 NAND_ECC_SOFT 改为 NAND_ECC_NONE 可以避免此问题。



```
end_request: I/O error, dev mtdblock3, sector 8
Buffer I/O error on device mtdblock3, logical block 1
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 8
Buffer I/O error on device mtdblock3, logical block 1
uncorrectable error :
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 16
Buffer I/O error on device mtdblock3, logical block 2
uncorrectable error :
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 16
Buffer I/O error on device mtdblock3, logical block 2
uncorrectable error :
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 24
Buffer I/O error on device mtdblock3, logical block 3
uncorrectable error :
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 24
Buffer I/O error on device mtdblock3, logical block 3
uncorrectable error :
end_request: I/O error, dev mtdblock3, sector 0
FAT: unable to read boot sector
VFS: Cannot open root device "mtdblock3" or unknown-block(31,3)
Please append a correct "root=" boot option; here are the available partitions:
1f00             256 mtdblock0 (driver?)
1f01             128 mtdblock1 (driver?)
1f02            5120 mtdblock2 (driver?)
1f03          256640 mtdblock3 (driver?)
1f04          262144 mtdblock4 (driver?)
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(31,3)
[<c002e954>] (unwind_backtrace+0x0/0xd8) from [<c02d0ca0>] (panic+0x40/0x118)
[<c02d0ca0>] (panic+0x40/0x118) from [<c0008e68>] (mount_block_root+0x1c8/0x208)
[<c0008e68>] (mount_block_root+0x1c8/0x208) from [<c00090fc>] (prepare_namespace+0x160/0x1b8)
[<c00090fc>] (prepare_namespace+0x160/0x1b8) from [<c0008434>] (kernel_init+0xd8/0x10c)
[<c0008434>] (kernel_init+0xd8/0x10c) from [<c002a868>] (kernel_thread_exit+0x0/0x8)
```

如果在烧写文件系统过程出错或者在启动过程中大量提示 flash 问题请注意勾选此选项，可以格式化 nand flash 再进行刷机，首先会进行格式化，格式化完成后 nand flash 中的 bootloader、内核、根文件目录都会被擦除，只剩下 nor flash 中还有 bootloader 作为通过 USB 线和电脑连接刷机的基础。nor flash 中的 bootloader 需要用 JLINK 烧录。



烧写选项:

☑ Low format flash   □ 跳过校准