

uClinux - Blackfin 移植手记

硬件平台：ADSP-BF533-EZKIT

扩展平台：OP-EX

ADI 软件环境：Visual DSP++5.0

宿主机操作系统：Redhat9

目录

uClinux - Blackfin 移植手记	1
1. 安装 Visual DSP++	3
2. 烧写 uboot	3
3. 搭建环境	3
3.1 宿主机上安装和配置 Linux	3
3.2 配置 Minicom、超级终端	3
3.2.1 在 Redhat 配置 Minicom	3
3.2.2 在 Windows 配置超级终端	6
3.2.3 安装交叉编译工具链	9
3.2.4 测试交叉编译环境	11
4. uClinux 内核的编译及烧写	15
4.1 编译 uclinux 内核	15
4.2 烧写 uImage 到目标板	18
5. 升级 u-boot	19
5.1 编译 u-boot	19
5.2 调试过程中出现的问题以及测试方法和结论	22
5.3 烧写 u-boot	22
6. uClinux 应用开发	22
6.1 编写应用程序	22
6.2 运行应用程序	23
7. Linux 基本知识	24
7.1 Linux 常用命令	24
OP 有话说	27

1. 安装 Visual DSP++

略！详细见 OpenADSP 其他文档中 VDSP 安装章节。

2. 烧写 uboot

略！详细见 OpenADSP 其他文档中 Flash 烧写章节

3. 搭建环境

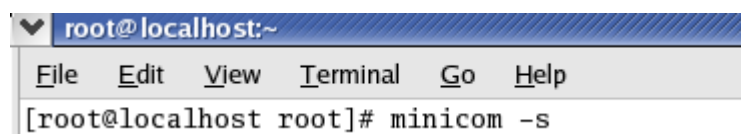
3.1 宿主机上安装和配置 Linux

首先编译环境需要安装宿主机(PC机)操作系统，推荐使用Redhat9。交叉编译(Crossing Compile)是相对本机编译(Native Compile)而言，指在异种CPU上进行的编译。比如在x86机器上编译运行于DSP处理器上的程序就是交叉编译。Linux环境下使用GNU工具完成编译、链接等过程，包括编译器GCC、运行库GLIBC和实用工具BINUTILS。
注意：在安装 Redhat9 时尽量把工具都装全，比如（NFS、TFTP）。

3.2 配置 Minicom、超级终端

3.2.1 在 Redhat 配置 Minicom

<1>将开发板和 pc 连接， Minicom 设置请参照下图：



```
root@localhost:~  
File Edit View Terminal Go Help  
[root@localhost root]# minicom -s
```

<2>选择 “Serial port setup”

```
—[configuration]—  
Filenames and paths  
File transfer protocols  
Serial port setup  
Modem and dialing  
Screen and keyboard  
Save setup as dfl  
Save setup as..  
Exit  
Exit from Minicom
```

<3>

A- Serial Device : /dev/ttyS0 是 COM 口的选择

B- Bps/Par/Bits : 57600 8N1 是波特率的设置

```
A - Serial Device      : /dev/ttyS0  
B - Lockfile Location  : /var/lock  
C - Callin Program     :  
D - Callout Program    :  
E - Bps/Par/Bits       : 57600 8N1  
F - Hardware Flow Control : No  
G - Software Flow Control : No  
  
Change which setting?
```

<4>参照上图设置好后，按“Enter”键，选择 Modem and dialing

```
—[configuration]—  
Filenames and paths  
File transfer protocols  
Serial port setup  
Modem and dialing  
Screen and keyboard  
Save setup as dfl  
Save setup as..  
Exit  
Exit from Minicom
```

<5>Modem and dialing 的设置请参照下图

[Modem and dialing parameter setup]

```
A - Init string .....
B - Reset string .....
C - Dialing prefix #1....
D - Dialing suffix #1....
E - Dialing prefix #2....
F - Dialing suffix #2....
G - Dialing prefix #3....
H - Dialing suffix #3....
I - Connect string ..... CONNECT
J - No connect strings .. NO CARRIER          BUSY
                               NO DIALTONE       VOICE
K - Hang-up string ..... ~~~~ATH^M
L - Dial cancel string .. ^M

M - Dial time ..... 45      Q - Auto bps detect ..... No
N - Delay before redial . 2  R - Modem has DCD line .. Yes
O - Number of tries ..... 10 S - Status line shows ... DTE speed
P - DTR drop time (0=no). 1  T - Multi-line untag .... No
```

Change which setting? █ (Return or Esc to exit)

<6>按“Enter”键，选择 Save setup as del 进行保存设置

[configuration]

```
Filenames and paths
File transfer protocols
Serial port setup
Modem and dialing
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom
```

<7>选择 Exit

```
—[configuration]—
Filenames and paths
File transfer protocols
Serial port setup
Modem and dialing
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom
```

```
root@localhost:~
File Edit View Terminal Go Help

Welcome to minicom 2.00.0

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Jan 25 2003, 00:15:18.

Press CTRL-A Z for help on special keys
```

<8>按开发板“Reset”键，

```
CPU: ADSP BF533 Rev.: 0.5
Board: ADI BF533 EZ-Kit Lite board
Support: http://blackfin.uclinux.org/
Clock: VC0: 594 MHz, Core: 594 MHz, System: 118 MHz
SDRAM: 32 MB
FLASH: 2.1 MB
In: serial
Out: serial
Err: serial
dm9000 i/o: 0x20320000, id: 0x90000a46
MAC: 00:d0:09:26:0a:5b
I2C: ready
Hit any key to stop autoboot: 0
ezkit>
```

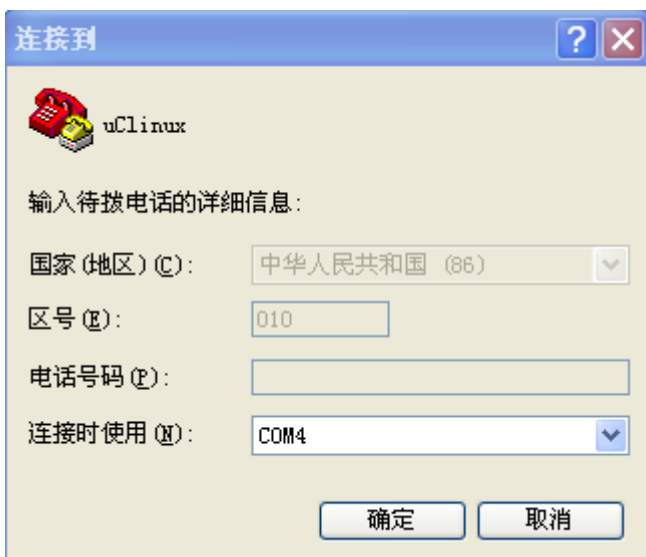
3.2.2 在 Windows 配置超级终端

选择“开始”->“所有程序”->“附件”->“通讯”->“超级终端”

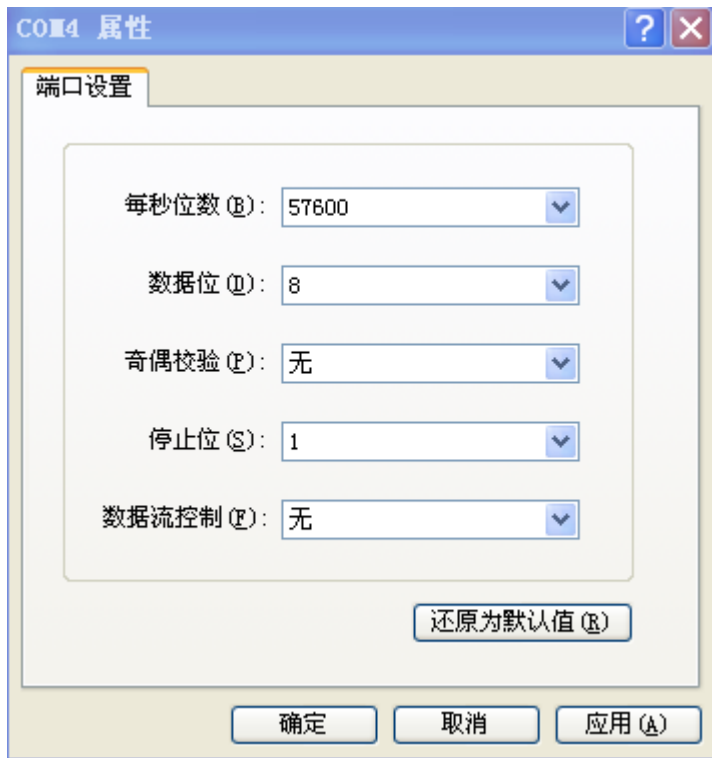
<1>输入名称，选择“确定”



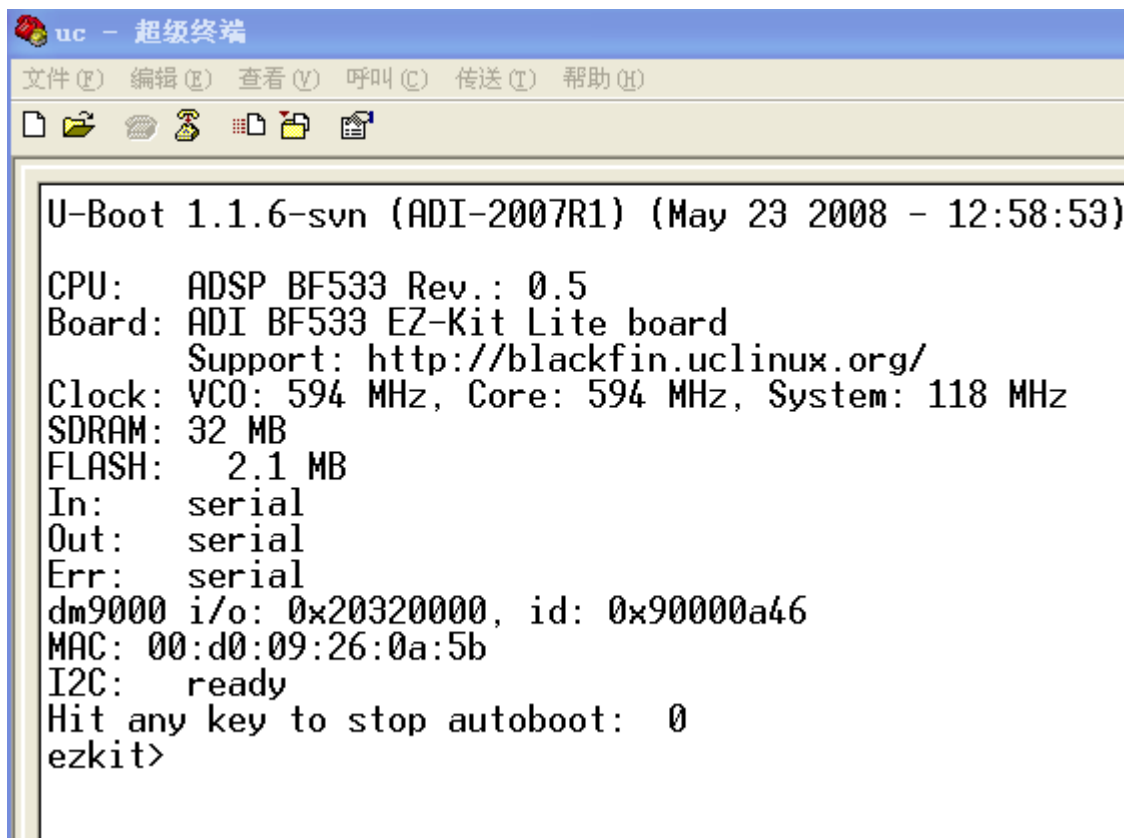
<2>选择连接时使用的 COM 口，具体在设备管理器上查看，点击“确定”



<3>将每秒位数设置为 57600，其他参照下图设置。点击“应用”、“确定”



<4>按开发板“Reset”键，出现如图：



3.2.3 安装交叉编译工具链

以下操作是配置交叉编译工具链:

<1>将光盘工具链拷贝到 Redhat 里面

```
[root@localhost Tool]# ls
blackfin-toolchain-08r1-8.i386.tar.gz
blackfin-toolchain-elf-gcc-4.1-08r1-8.i386.tar.gz
blackfin-toolchain-uclibc-default-08r1-8.i386.tar.gz
[root@localhost Tool]#
```

<2>先用 `gzip -d file.tar.gz` 解压成 `file.tar` 如下图(解压时请耐心等待)

```
[root@localhost Tool]# ls
blackfin-toolchain-08r1-8.i386.tar
blackfin-toolchain-elf-gcc-4.1-08r1-8.i386.tar
blackfin-toolchain-uclibc-default-08r1-8.i386.tar
[root@localhost Tool]#
```

<3>然后用 `tar -xvf file.tar` 分别解压, 生成/opt 目录

```
[root@localhost Tool]# ls
blackfin-toolchain-08r1-8.i386.tar
blackfin-toolchain-elf-gcc-4.1-08r1-8.i386.tar
blackfin-toolchain-uclibc-default-08r1-8.i386.tar
opt
[root@localhost Tool]#
```

<4>进入/opt 目录, 将 uClinux 目录拷贝到根目录下的/opt 目录里

```
[root@localhost Tool]# ls
blackfin-toolchain-08r1-8.i386.tar
blackfin-toolchain-elf-gcc-4.1-08r1-8.i386.tar
blackfin-toolchain-uclibc-default-08r1-8.i386.tar
opt
[root@localhost Tool]# cd opt/
[root@localhost opt]# pwd
/share/Tool/opt
[root@localhost opt]# ls
uClinux
[root@localhost opt]# cp -a uClinux /opt/
```

<5>进入根目录下/opt

```
[root@localhost /]# ls
bin  dev  home  lib  misc  opt  root  share  tmp  var
boot  etc  initrd  lost+found  mnt  proc  sbin  tftpboot  usr
[root@localhost /]# cd /opt/
[root@localhost opt]# ls
uClinux
```

<6>输入 find -name "bin" 注：此操作是为了查找需要在/root/.bashrc 里添加的环境变量

```
[root@localhost ~]# ls
bin  dev  home  lib          misc  opt   root  share  tmp  var
boot etc  initrd lost+found mnt   proc  sbin  tftpboot usr
[root@localhost ~]# cd /opt/
[root@localhost opt]# ls
uClinux
[root@localhost opt]# find -name "bin"
./uClinux/bfin-linux-uclibc/bfin-linux-uclibc/bin
./uClinux/bfin-linux-uclibc/bin
./uClinux/bfin-uclinux/bin
./uClinux/bfin-uclinux/bfin-uclinux/bin
./uClinux/bfin-elf/bfin-elf/bin
./uClinux/bfin-elf/bin
[root@localhost opt]#
```

<7>用 vim 打开/root/.bashrc

```
[root@localhost root]# pwd
/root
[root@localhost root]# vim .bashrc
```

<8>添加如下代码到.bashrc 最后

export

```
PATH=$PATH:/opt/uClinux/bfin-linux-uclibc/bfin-linux-uclibc/bin:/opt/uClinux/bfin-linux-uclibc/bin:/opt/uClinux/bfin-uclinux/bin:/opt/uClinux/bfin-uclinux/bfin-uclinux/bin:/opt/uClinux/bfin-elf/bfin-elf/bin:/opt/uClinux/bfin-elf/bin
```

```
root@localhost:~  
File Edit View Terminal Go Help  
# .bashrc  
  
# User specific aliases and functions  
  
alias rm='rm -i'  
alias cp='cp -i'  
alias mv='mv -i'  
  
# Source global definitions  
if [ -f /etc/bashrc ]; then  
    . /etc/bashrc  
fi  
  
export PATH=$PATH:/opt/uClinux/bfin-linux-uclibc/bfin-linux-uclibc/bin:/opt/uClinux/bfin-linux-uclibc/bin:/opt/uClinux/bfin-uclinux/bin:/opt/uClinux/bfin-uclinux/bin:/opt/uClinux/bfin-elf/bfin-elf/bin:/opt/uClinux/bfin-elf/bin
```

然后保存、退出.bashrc 后，输入 source .bashrc 命令。最好是 reboot 一下 Redhat

3.2.4 测试交叉编译环境

注：以下操作是为了测试工具链的安装是否正确，内核编译通过，表示工具链安装正确

<1>将光盘上 uClinux-dist-2008R1-RC8.tar.bz2 拷贝到 Redhat 里。

<2>在 Redhat 终端输入 bunzip2 -d uClinux-dist-2008R1-RC8.tar.bz2 命令，

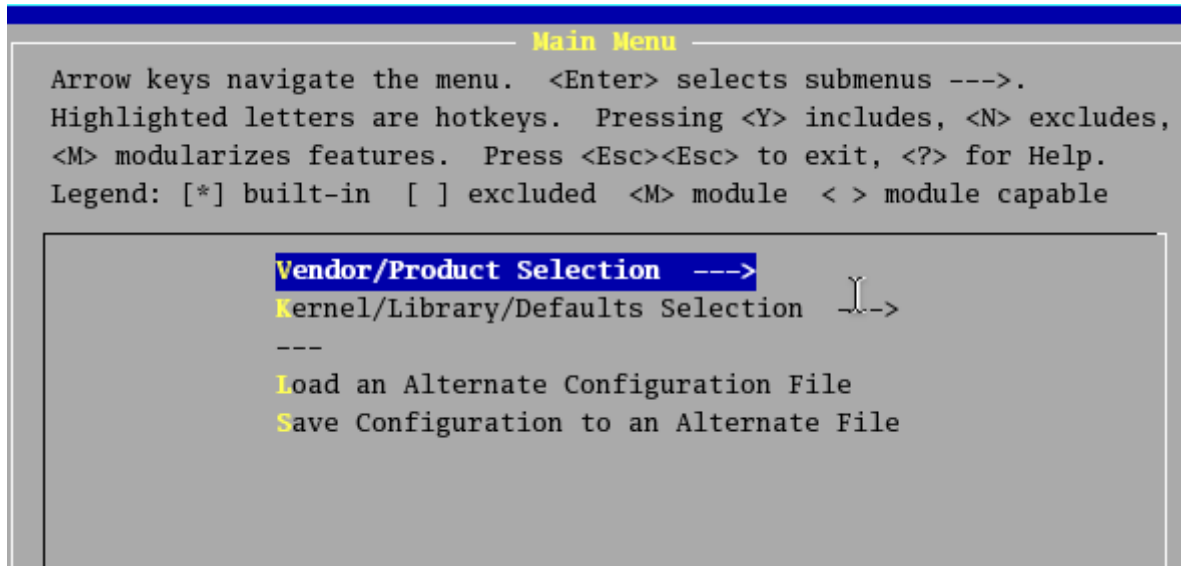
<3>然后输入 tar -xvf uClinux-dist-2008R1-RC8.tar

```
[root@localhost share]# ls  
Tool          uClinux-dist-2008R1-RC8      uCResourceV0.1  
toolchain     uClinux-dist-2008R1-RC8.tar  
[root@localhost share]#
```

<4>进入 uClinux-dist-2008R1-RC8 目录里，输入 make menuconfig 命令

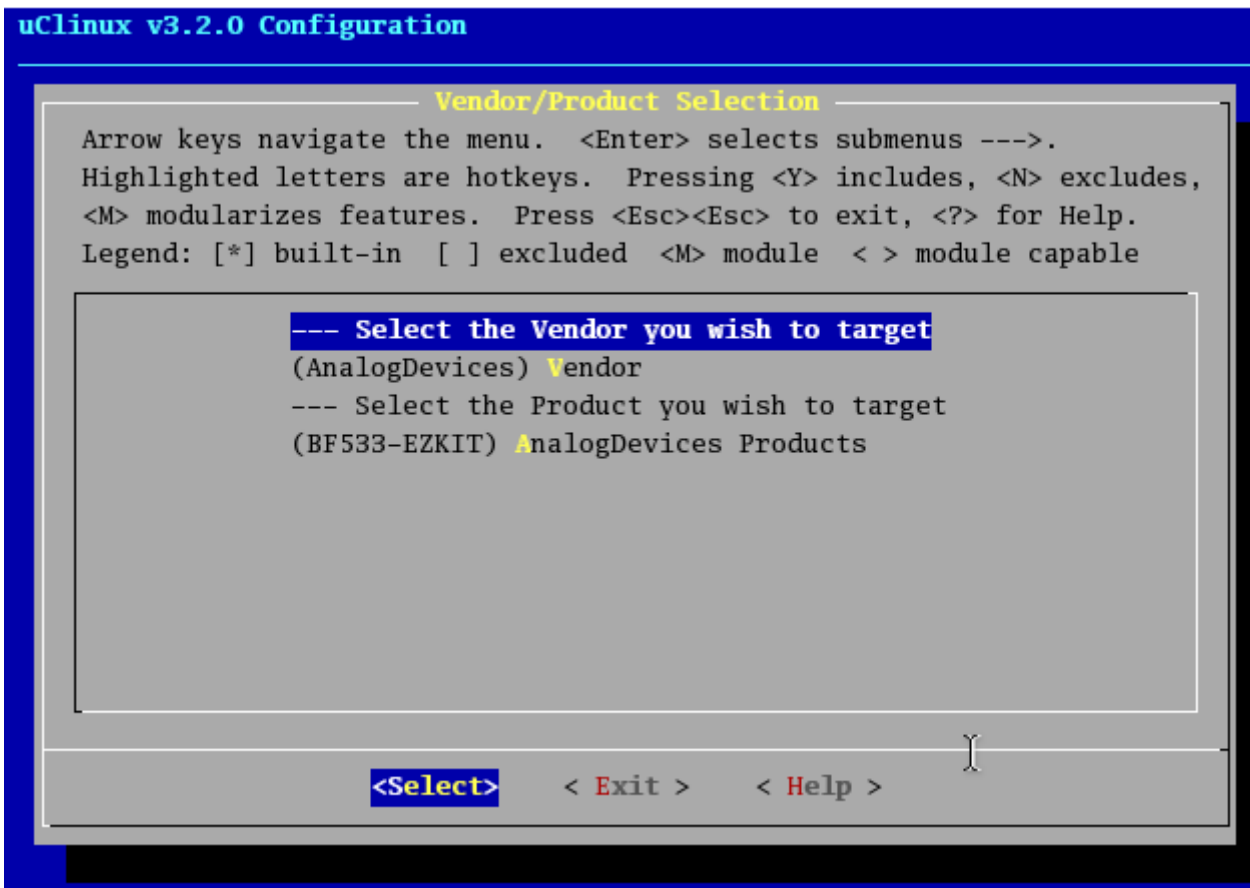
```
[root@localhost uClinux-dist-2008R1-RC8]# ls  
autoconf.h  config.arch  freeswan    Makefile    romfs       tools  
bfin_patch  config.in    images      openswan    SOURCE       user  
bin          COPYING     lib         README     staging     vendors  
config       Documentation linux-2.6.x release_notes testsuites  
[root@localhost uClinux-dist-2008R1-RC8]# make menuconfig
```

<5>进入 Vendor/Product Selection

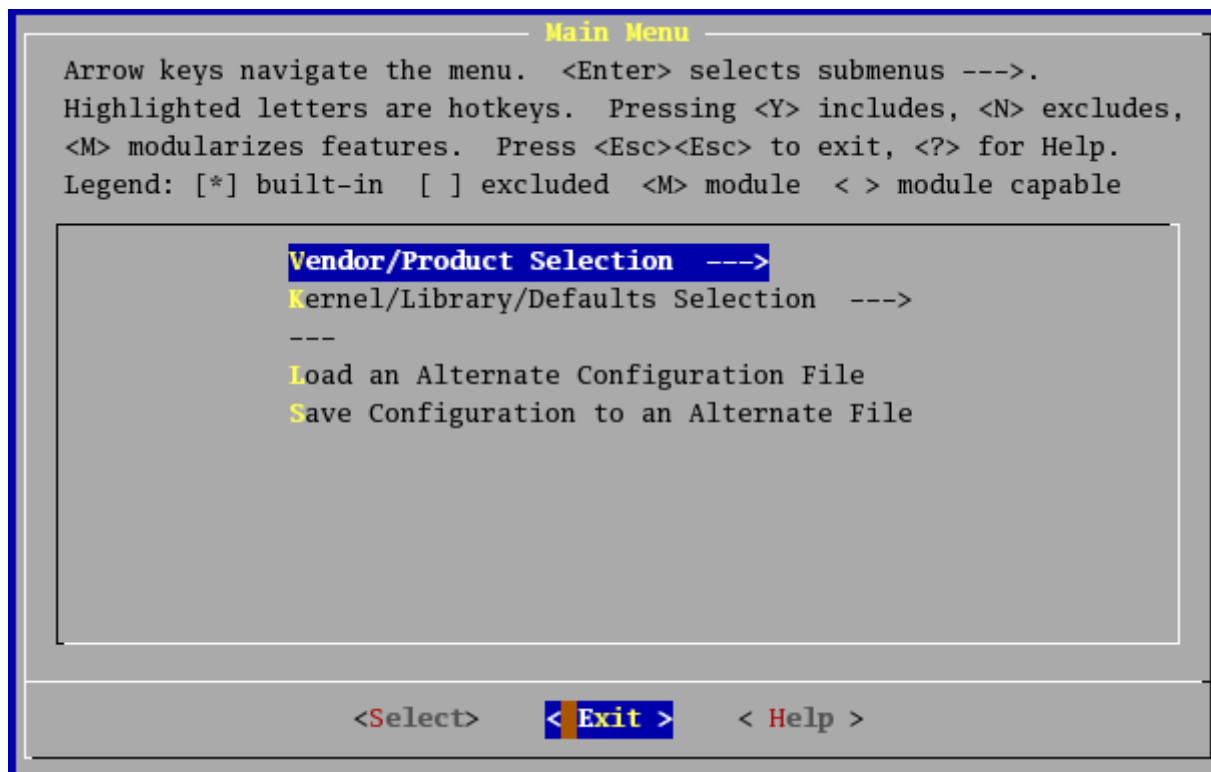


<6>选择 Vendor 为 Analog Devices.

<7>选择 AnalogDevices Products 为 BF533-EZKIT.



<8>然后退出，回到主界面



<9>选择 exit，然后输入 make

```
[root@localhost uClinux-dist-2008R1-RC8]# make
```

等待内核编译完成。编译不成功请检查是否按以上步骤操作。

注：编译不成功可能出错的原因：

<1>make 版本过低

<2>检查是否按步骤操作

1：检查 make 的版本。

```
[root@localhost uClinux-dist-2008R1-RC8]# make -ve
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i686-pc-linux-gnu
[root@localhost uClinux-dist-2008R1-RC8]#
```

Make 版本低于 GNU Make 3.81 请将光盘 make-3.81.tar.bz2 和 findutils-4.4.0.tar.gz 安装操作步骤如下：

(一)：安装 make-3.81.tar.bz2

<1>解压 make-3.81.tar.bz2

```
[root@localhost share]# bunzip2 -d make-3.81.tar.bz2
[root@localhost share]# ls
findutils-4.4.0.tar.gz  Tool          uClinux-dist-2008R1-RC8  uCResourceV0.1
make-3.81.tar          toolchain    uClinux-dist-2008R1-RC8.tar  updata
[root@localhost share]# tar -xvf make-3.81.tar
```

<2>进入 make-3.81。 [root@localhost share]# cd make-3.81

<3>[root@localhost make-3.81]# ./configure

<4>[root@localhost make-3.81]# make

<5>[root@localhost make-3.81]# make install

(二) 安装 findutils-4.4.0.tar.gz

<1>[root@localhost share]# gzip -d findutils-4.4.0.tar.gz

<2>[root@localhost share]# tar -xvf findutils-4.4.0.tar

<3>[root@localhost share]# cd findutils-4.4.0

<4>[root@localhost findutils-4.4.0]# ./configure

<5>[root@localhost findutils-4.4.0]# make

<6>[root@localhost findutils-4.4.0]# make install

以上 2 个安装包装完后，重启 Redhat。然后用 make -ve 查看 make 是否升级。

4. uClinux 内核的编译及烧写

4.1 编译 uclinux 内核

<1> 内核编译的一些选项：

```
Vendor/Product Selection --->
Kernel/Library/Defaults Selection --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File
```

```
--- Select the Vendor you wish to target
(AnalogDevices) Vendor
--- Select the Product you wish to target
(BF533-EZKIT) AnalogDevices Products
```

<2>选择 **Kernel/Library/Defaults Selection** 配置 **Embedded Linux Configuration**.

```
Vendor/Product Selection --->
Kernel/Library/Defaults Selection --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File
```

```
--- Kernel is linux-2.6.x
--- Libc is None
[ ] Default all settings (lose changes)
[*] Customize Kernel Settings
[*] Customize Vendor/User Settings
[ ] Update Default Vendor Settings
```

<3>配置完了，选择退出。

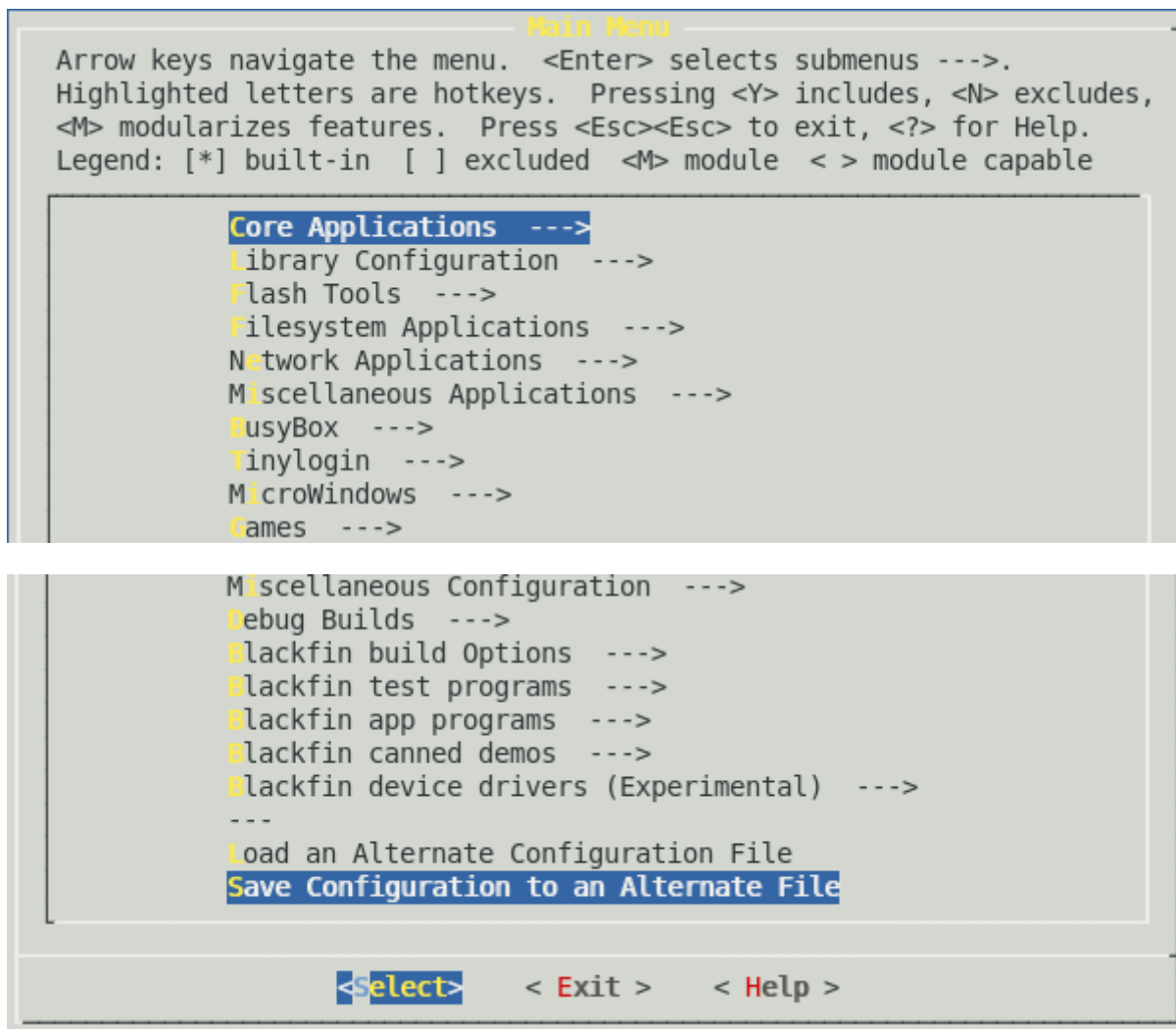
<4>等待进入 linux kernel configuration 的配置选项，配置完后，选择退出

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

Code maturity level options --->
General setup --->
Loadable module support --->
Block layer --->
Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
Blackfin Processor Options --->
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
Executable file formats --->
Power management options --->
CPU Frequency scaling --->

Networking --->
Device Drivers --->
File systems --->
Profiling support --->
Kernel hacking --->
Security options --->
Cryptographic options --->
Library routines --->
---
Load an Alternate Configuration File
```

<5>选择 Main Menu 的配置选项，配置完后，选择退出



<6>在 Terminal 输入 make 命令，开始编译内核。

```
make[2]: Leaving directory `/share/uClinux-dist-2008R1-RC8/config'
make[1]: Leaving directory `/share/uClinux-dist-2008R1-RC8'
[root@localhost uClinux-dist-2008R1-RC8]# make
```

<7>编译完成后，在当前目录下生成/images 目录

```
[root@localhost uClinux-dist-2008R1-RC8]# ls
autoconf.h  config.arch  freeswan    Makefile     romfs        tools
bfin_patch  config.in    images      openswan     SOURCE       user
bin          COPYING     lib         README       staging       vendors
config      Documentation linux-2.6.x release_notes testsuites
[root@localhost uClinux-dist-2008R1-RC8]#
```

<8>进入/images 目录，拷贝 uImage 到根目录下的/tftpboot 目录

```
[root@localhost uClinux-dist-2008R1-RC8]# cd images/
[root@localhost images]# ls
linux                rootfs.initramfs.contents  uImage                vmlinux
linux.initramfs      rootfs.initramfs.gz       uImage.initramfs
linux.initramfs.gz   System.map.initramfs      uImage.initramfs.gz
rootfs.initramfs     System.map.initramfs.gz   vmImage
[root@localhost images]# cp uImage /tftpboot/

[root@localhost tftpboot]# pwd
/tftpboot
[root@localhost tftpboot]# ls
uImage
[root@localhost tftpboot]#
```

注：根目录下的/tftpboot 是在 Redhat 系统上装 tftp 服务才有的。Tftp 具体用法请参看网上资料。

4.2 烧写 uImage 到目标板

在 Terminal 里输入 minicom（具体设置请参看 3.2 节设置 minicom），然后输入：

```
<1> ezkit> setenv serverip    10.0.0.81      注：此命令是设置 tftp server 端的 ip 地址
<2> ezkit> setenv ipaddr      10.0.0.84      注：此命令是设置开发板的 ip 地址
<3> ezkit> saveenv            注：此命令是保存设置
```

```
ezkit> setenv serverip 10.0.0.81
ezkit> setenv ipaddr 10.0.0.84
ezkit> saveenv
```

<4>然后输入 tftp 0x1000000 uImage 如图：

```
ezkit> tftp 0x1000000 uImage
dm9000 i/o: 0x20320000, id: 0x90000a46
MAC: 00:d0:09:26:0a:5b
TFTP from server 10.0.0.81; our IP address is 10.0.0.84
Filename 'uImage'.
Load address: 0x1000000
Loading: T #####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
done
Bytes transferred = 3464913 (34ded1 hex)
ezkit>
```

```
<5>输入 bootm 0x1000000
ezkit> bootm 0x1000000
## Booting image at 01000000 ...
   Image Name:   Linux-2.6.22.18-ADI-2008R1-svn
   Created:      2008-08-31   6:52:18 UTC
   Image Type:   Blackfin Linux Kernel Image (gzip compressed)
   Data Size:    3464849 Bytes =  3.3 MB
   Load Address: 00001000
   Entry Point:  00144000
```

成功后，就可以在 minicom 里看见 uclinux 的界面了，如图：

```

d888888b.
8P"YP"Y88
8|o||o|88
8'   .88
8' ._. ' Y8.
d/      `8b.
dP      Y8b.
d8:' " `::88b
d8"      'Y88b
:8P      ' :888
8a.      : a88P
./"Yaa : .| 88P|
\      YP"  | 8P |
/      \.____.d| . '
\---.._)8888P`._.' jgs/a:f

      (TM)
      For embedded processors including
      the Analog Devices Blackfin

For further information, check out:
- http://blackfin.uclinux.org/
- http://docs.blackfin.uclinux.org/
- http://www.uclinux.org/
- http://www.analog.com/blackfin

Have a lot of fun...

BusyBox v1.4.1 (2008-08-31 13:57:12 CST) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

root:/>
```

5. 升级 u-boot

5.1 编译 u-boot

ADZS-BF533-EZKIT 不带有网络功能,为方便使用,现需在 uboot 中添加 OP-EX 扩展板中 DM9000 驱动. uboot 源码中的默认网络控制芯片为 SMC91111,需要将其改为 DM9000.

修改内容涉及到的文件有:

- 1.uboot 下主 Makefile 文件
- 2.lib_blackfin\board.c
- 3.include\configs\bf533-ezkit.h
- 4.drivers\dm9000x.c
- 5.drivers\Makefile

修改内容:

- 1.将 uboot 下 Makefile 文件中 SUBDIRS = tools examples 中的 examples 屏蔽,不对 examples 进行编译.

- 2.在 lib_blackfin\board.c 文件中:

- a.添加#include "../drivers/dm9000x.h",将目标网卡头文件包含进来
- b.屏蔽 SMC91111 相关内容,改为 dm9000 内容

- 3.include\configs\bf533-ezkit.h 是重点需要修改的地方

屏蔽 SMC91111 内容,添加 dm9000 内容,如下所示:

```
#define CONFIG_DRIVER_DM9000 1
#define CONFIG_DM9000_BASE 0x20320000
#define DM9000_IO CONFIG_DM9000_BASE
#define DM9000_DATA (CONFIG_DM9000_BASE + 0x10)
#define CONFIG_DM9000_USE_16BIT 1
#define CONFIG_ETHADDR 08:00:3e:26:0a:5b
#define CONFIG_NETMASK 255.255.255.0
#define CONFIG_IPADDR 10.0.0.138
#define CONFIG_SERVERIP 10.0.0.137
#define CONFIG_GATEWAYIP 10.0.0.1
```

其中,CONFIG_DM9000_BASE 是重点,可以参考 ssk 文档,找到网卡映射地址.

- 4.drivers\dm9000x.c 中需要在每一个 inb,inw,inl,outb,outw,outl 添加延时函数(asm("ssync;")),这样才能实现主机和目标板的通讯正常.

- 5.在 drivers\dm9000x.c 中需要将环境变量 ethaddr 读入,写入 dm9000 的相关位置,需要屏蔽

```
/* Set Node address */
for (i = 0; i < 6; i++)
    ((u16 *) bd->bi_enetaddr)[i] = read_srom_word(i);
```

用以下语句替换:

```
char *tmp = getenv ("ethaddr");
char *end;
```

```
for (i=0; i<6; i++)
{
    bd->bi_enetaddr[i] = tmp ? simple_strtoul(tmp, &end, 16) : 0;
    if (tmp)
        tmp = (*end) ? end+1 : end;
```

```
}
```

为解决“could not establish link”提示和慢响应的问题,屏蔽下面语句:

注: 被屏蔽的语句是 MII 接口用的, 放在这是错误的, 并且浪费了 10 秒钟

```
i = 0;
while (!(phy_read(1) & 0x20))
{
    /* autonegotiation complete bit */
    udelay(1000);
    i++;
    if (i == 10000)
    {
        printf("could not establish link\n");
        return 0;
    }
    printf(" link=%d\n",i);
}
```

```
/* see what we've got */
lnk = phy_read(17) >> 12;
printf("operating at ");
switch (lnk)
{
case 1:
    printf("10M half duplex ");
    break;
case 2:
    printf("10M full duplex ");
    break;
case 4:
    printf("100M half duplex ");
    break;
case 8:
    printf("100M full duplex ");
    break;
default:
    printf("unknown: %d ", lnk);
    break;
}
("ssync;");
```

6.drivers\Makefile 中添加 dm9000x.o 实现对 dm9000 的编译.

5.2 调试过程中出现的问题以及测试方法和结论

- 1.若出现 "dm9000 not found"错误,可能的原因是 dm9000 baseaddr 不正确,这样 ID 也读取不正常.
- 2.使用交叉网线,否则主机和目标板不能通讯.
- 3.如果启动正常,主机右下角的网络连接图标会显示联机.若 uboot 启动后,显示无连接,且 ssk D5,D6,D7 都不亮,那么可能是网卡初始化不正常,或者在 lib_blackfin\board.c dm9000 定义处没有添加 extern int eth_init(bd_t *bd); eth_init(gd->bd);出现的错误提示为:ping faied;host is not alive:
- 4.若 uboot 启动后显示网络连接成功,可以执行 ping 命令,此时需要检查主机的 ip, mask,gatewayip 与目标板是否一致.目标板中可以使用 printenv 查看设置.
执行 ping 命令后,若提示 ping faied;host is not alive 则 dm9000x.c 延时函数需要修改.
- 5.若以上调试步骤正确,在执行 ping 命令时,出现 ping failed; host is not alive 的情况,则说明驱动程序中延时匹配不正确,则需要对程序中 inb, inw,inl,outb,outw,outl 语句后面添加 asm("ssync;");确保写入和读出数据完整,具体的修改内容可以参见 dm9000x.c.
- 6.以上工作进行完毕,进行测试时,启动 uboot 则在主机右下角会出现网络连接成功提示,执行 ping 命令后,在超级终端中会出现 " host is alive" 字样.多次执行 ping 命令(连续执行 70 次),结果正常.

5.3 烧写 u-boot

此部分的操作请参看 OpenADSP 开源社区其他文档中关于 Flash 烧写的内容。

6. uClinux 应用开发

6.1 编写应用程序

编写一个 helloworld 来演示如何在 ADZS-BF533-EZKIT 上编译、运行应用程序

```
#include <stdio.h>
int main()
{
    printf("Hello, DiMa\n");
    return 0;
}
```

<1>在 Terminal 输入[root@localhost ~]# bfin-uclinux-gcc -v 注: 此命令是查看 Redhat9 里是否安装交叉编译工具, 如果没安装, 请参看安装交叉工具链的章节。

<2>输入[root@localhost helloworld]# bfin-uclinux-gcc -Wl,-elf2flt hello.c -o hello

```
[root@localhost helloworld]# bfin-uclinux-gcc -Wl,-elf2flt hello.c -o hello
[root@localhost helloworld]# ls
hello hello.c hello.gdb
[root@localhost helloworld]#
```

注：-Wl,-elf2flt 其中 W 是大写，逗号两边没有空格

6.2 运行应用程序

<1>将编译好的./helloworld 目录拷贝到 uclinux 源代码./uClinux-dist-2008R1-RC8/romfs/下。

注：./helloworld 为 hello.c 的目录

```
[root@localhost share]# cp -a helloworld/ uClinux-dist-2008R1-RC8/romfs/
```

<2>进入 uClinux-dist-2008R1-RC8 目录

```
[root@localhost share]# cd uClinux-dist-2008R1-RC8/
```

<3>输入 make image 编译 image

```
[root@localhost uClinux-dist-2008R1-RC8]# make image
```

注：下图是演示上面步骤<1>、<2>、<3>。

```
[root@localhost share]# ls
bfin-t350mcqb-fb.c          helloworld
blackfin-toolchain-08r1-8.i386.tar  opt
blackfin-toolchain-elf-gcc-4.1-08r1-8.i386.tar  uClinux-dist-2008R1-RC8
blackfin-toolchain-uclibc-full-08r1-8.i386.tar  uClinux-dist-2008R1-RC8.tar
[root@localhost share]# cp -a helloworld/ uClinux-dist-2008R1-RC8/romfs/
[root@localhost share]# cd uClinux-dist-2008R1-RC8/romfs/
[root@localhost romfs]# ls
bin etc helloworld home lib mnt proc root sbin sys tmp usr var
[root@localhost romfs]# pwd
/share/uClinux-dist-2008R1-RC8/romfs
[root@localhost romfs]# cd ..
[root@localhost uClinux-dist-2008R1-RC8]# ls
autoconf.h  config.arch  freeswan  Makefile  romfs  tools
bfin_patch  config.in   images    openswan  SOURCE  user
bin          COPYING    lib        README    staging  vendors
config       Documentation  linux-2.6.x  release_notes  testsuites
[root@localhost uClinux-dist-2008R1-RC8]# make image
```

<4>编译完后，进入 images 目录

```
[root@localhost uClinux-dist-2008R1-RC8]# cd images/
```

<5>把 uImage 拷贝到/tftpboot/下

```
[root@localhost images]# cp uImage /tftpboot/
```



```
[root@localhost uClinux-dist-2008R1-RC8]# cd images/
[root@localhost images]# ls
linux          rootfs.initramfs.contents  uImage          vmlinux
linux.initramfs  rootfs.initramfs.gz       uImage.initramfs
linux.initramfs.gz  System.map.initramfs      uImage.initramfs.gz
rootfs.initramfs  System.map.initramfs.gz   vmImage
[root@localhost images]# cp uImage /tftpboot/
```

<6>在 minicom 里输入 tftp 0x1000000 uImage

<7>输入 bootm 0x1000000

<8>请参照下图:

```
root: /> ls
bin      helloworld  lib      root      tmp
dev      home       mnt      sbin     usr
etc      init       proc     sys      var
root: /> cd helloworld/
root: /helloworld> ls
hello    hello.c    hello.gdb
root: /helloworld> ./hello
Hello , DiMa
root: /helloworld> █
```

7. Linux 基本知识

7.1 Linux 常用命令

下面中 Linux 常用命令，也适合于嵌入式 Linux 系统。

<1>文件列表 -ls

ls #以默认方式显示当前目录文件列表

ls-a #显示所有文件包括隐藏文件

ls-l #显示文件属性，包括大小，日期，符号连接，是否可读写是否可执行

<2>目录切换-cd

cd dir #切换到当前目录下的 dir 目录

cd/ #切换到根目录

cd.. #切换到上一级目录

<3> 复制 -cp

`cp source target` #将文件 source 复制为 target
`cp/root/source.` #将/root 下的文件 source 复制到当前目录
`cp-av source_dir target_dir` #将整个目录复制到，两目录完全一样
`cp-fr source_dir target_dir` #将整个目录复制，并且是以非法链接方式复制，当 source 目录带有符号链接时，两个目录不相同

<4> 删除 - rm

`rm file` #删除某一个文件
`rm-fr dir` #删除当前目录下叫 dir 的整个目录

<5> 移动 - mv

`mv source target` #将文件 source 更名为 target

<6> 比较 - diff

`diff dir1 dir2` #比较目录 1 与目录 2 的文件列表是否相同，但不比较文件的实际内容，不同则列出
`diff file1 file2` #比较文件 1 与文件 2 的内容是否相同，如果是文本格式的文件，则将不相同的内容显示，如果是二进制代码则只表示两文件是不同的

<7> 回显 - echo

`echo message` #显示一串字符
`echo "message message2"` #显示不连续的字符串

<8> 容量查看 - du

`du` #计算当前目录的容量
`du -sm/root` #计算/root 目录的容量并以 M 为单位

<9> 文件内容查看 - cat

`cat file` #显示文件的内容，和 DOS 的 type 相同
`cat file | more` #显示文件的内容并传输到 more 程序实现分页显示，使用命令 `less file` 可实现相同的功能

<10> 分页查看 - more

`more` #分页命令，一般通过管道将内容传给它，如 `ls | more`

<11> 时间日期 - date

`date` #显示当前日期时间

date -s 20:30:30 #设置系统时间为 20: 30: 30

date -s 2002-3-5 #设置系统时期为 2002-3-5

<12> 查找 -find

find - name/path file # 在/path 目录下查找看是否有文件 file

<13> 搜索 - grep

grep -ir "chars" #在当前目录的所有文件查找字串 chars，并忽略大小写，-i 为大小写，-r 为下一级目录

<14> 设置环境变量 - export

export LC_ALL=zh_CN.GB2312 #将环境变量 LC_ALL 的值设为 zh_CN.GB2312

<15> 编辑 - vi

vi file #编辑文件 file

vi 原基本使用及命令：

输入命令的方式为先按 ctrl+c，然后输入：x（退出），：x!（退出并保存）：w（写入文件），：w!（不询问方式写入文件），：r file（读文件 file），%s/oldchars/newchars/g（将所有字符串 oldchars 换成 newchars）

这一类的命令进行操作

<16> 压缩与解压 - tar

tar xfv file.tgz #将文件 file.tgz 解压

tar cfzv file.tgz source_path #将文件 source_path 压缩为 file.tgz

<17> 挂接 - mount

mount -t yaffs/dev/mtdblock/0/mnt #把/dev/mtdblock/0 装载到/mnt 目录

mount -t nfs 192.168.0.1:/root/mnt #将 nfs 服务的共享目录 root 挂接到/mnt 目录

<18> 启动信息显示 - dmesg

dmesg #显示 kernle 启动及驱动装载信息

<19> -chmod

chmod a+x file #将 file 文件设置为可执行，脚本类文件一定要这样设置一个，否则得用 bash file 才

能执行

chmod 666 file #将文件 file 设置为可读写

<20> 创建节点 - mknod

Mknod/dev/ttyl c 4 1 # 创建字符设备 ttyl,主设备呈为 4, 从设备呈为 1, 即第一个 tty 终端

<21> 进程查看 - ps

ps #显示当前系统进程信息

ps -ef #显示系统所有进程信息

<22> 杀死进程 - kill

kill -9500 #将进程编号为 500 的程序杀死

OP 有话说

最后一章有点班门弄斧，作者仅仅是把自己在 Blackfin 上玩 uC 的一个过程简单的记录了下来，各位可做参考，谢谢支持 OpenADSP 开源社区。

官方网站: www.openadsp.com

社区论坛: www.openadsp.com/bbs

淘宝网店: <http://dsp-tools.taobao.com/>

OpenADSP 实体店: 北京新中发电子市场 3659 室

联系我们: sale@openadsp.com

