

# 实验八 网卡驱动移植

原创

一条飞的鱼儿

于 2023-08-24 20:58:40 发布

阅读量388

收藏

点赞数

版权

文章标签: 

linux

运维

## 【实验目的】

掌握 [Linux](#) 内核配置的基本方法，完成对网卡驱动、NFS 等相关功能的配置

## 【实验环境】

- 1. ubuntu 14.04 发行版
- 2. FS4412 实验平台
- 3. 交叉编译工具: arm-none-linux-gnueabi-

## 【注意事项】

- 1. 实验步骤中以“\$”开头的命令表示在 ubuntu 环境下执行，以“#”开头的命令表示在开发板下执行

## 【实验步骤】

- 1. 在 [内核](#) 源码的顶层目录下执行如下命令，修改内核配置

```
1 | $ make menuconfig
```

给内核选配 [DM9000 网卡驱动](#)，然后选择“Save”保存

```
1 | Device Drivers --->
2 |     [*] Network device support --->
3 |         [*] Ethernet driver support --->
4 |             <*> DM9000 support
```

解释

因为内核要使用 NFS 去挂载根文件系统，而 [NFS 是基于 TCP 协议实现的](#)，所以需要选配 TCP 相关的网络 [协议](#)（部分功能默认已经选配），然后选择“Save”保存

```
1 | [*] Networking support --->
2 |     Networking options --->
3 |         <*> Packet socket
4 |         <*> Unix domain sockets
5 |         [*] TCP/IP networking
6 |         [*] IP: kernel level autoconfiguration
```


解释

因为内核要使用 NFS 去挂载根文件系统，所以需要给内核选配 [NFS 客户端及相关功能](#)

```

1 | File systems --->
2 |     [*] Network File Systems --->
3 |         <*> NFS client support
4 |         <*> NFS client support for NFS version 3 (NEW)
5 |         [*] NFS client support for the NFSv3 ACL protocol extension
6 |         [*] Root file system on NFS

```

 [登录](#) [复制](#)

设置完成后通过方向键选择‘Save’保存即可，然后选择‘Exit’退出该配置界面

## 2. 在设备树中添加网卡的硬件信息


```
1 | $ vi arch/arm/boot/dts/exynos4412-fs4412.dts
```

在文件的末尾，最后一个花括号前添加如下内容（即要写在根节点之内）

```

1 |     srom-cs1@5000000 {
2 |         compatible = "simple-bus";
3 |         #address-cells = <1>;
4 |         #size-cells = <1>;
5 |         reg = <0x5000000 0x1000000>;
6 |         ranges;
7 |     ethernet@5000000 {
8 |         compatible = "davicom,dm9000";
9 |         reg = <0x5000000 0x2 0x5000004 0x2>;
10 |         interrupt-parent = <&gpx0>;
11 |         interrupts = <6 4>;
12 |         davicom,no-EEPROM;
13 |         mac-address = [00 0a 2d a6 55 a2];
14 |     };
15 | };

```

 [解释](#) [🔗](#)

## 3. 修改时钟相关配置（忽略无用的时钟）

```
1 | $ vi drivers/clk/clk.c
```

将

```
1 | static bool clk_ignore_unused;
```

修改为以下内容，然后保存退出



```
1 | static bool clk_ignore_unused=true;
```

## 4. 修改 EMMC 相关配置

```
1 | $ vi drivers/mmc/core/mmc.c
```



将

```
1 | if (card->ext_csd.rev > 7) {
2 |     pr_err("%s: unrecognised EXT_CSD revision %d\n",
3 |         mmc_hostname(card->host), card->ext_csd.rev);
4 |     err = -EINVAL;
5 |     goto out;
6 | }
```

 解释 | 

修改为以下内容，然后保存退出

```
1 | #if 0
2 | if (card->ext_csd.rev > 7) {
3 | #else
4 | if (card->ext_csd.rev > 8) {
5 | #endif
6 |     pr_err("%s: unrecognised EXT_CSD revision %d\n", mmc_hostname(card->host),
7 |         err = -EINVAL;
8 |         goto out;
9 | }
```

 解释 | 

## 5. 编译内核和设备树

```
1 | $ make uImage
2 | $ make dtbs
```

## 6. 测试内核和设备树

将编译生成的内核和设备树拷贝到 tftp 的工作目录

```
1 | $ sudo cp arch/arm/boot/uImage /tftpboot
2 | $ sudo cp arch/arm/boot/dts/exynos4412-fs4412.dtb /tftpboot/
3 | $ sudo chmod 777 /tftpboot/*
```

重启 tftp 和 nfs 服务器

```
1 | $ sudo service tftpd-hpa restart
2 | $ sudo service nfs-kernel-server restart
```

重启开发板查看现象，如图所示，此时 Linux 内核已经能通过 NFS 去挂载根文件系统

在终端下执行‘ls’命令我们就能看到根文件系统的内容

```
[ 1.765000] clk: Not disabling unused clocks
[ 2.790000] VFS: Mounted root (nfs filesystem) on device 0:10.
[ 2.800000] devtmpfs: mounted
[ 2.800000] Freeing unused kernel memory: 228K (c050d000 - c0546000)
[ 13.520000] nfs: server 192.168.0.100 not responding, still trying
[ 13.525000] nfs: server 192.168.0.100 OK
```

Please press Enter to activate this console.

```
[root@farsight]# ls
bin      etc      lib      mnt      root     sys      usr
dev      home    linuxrc  proc     sbin     tmp      var
[root@farsight]#
```

英 , 