

# 制作根文件系统

## 目录

实验内容 .....	2
实验目的 .....	2
实验材料 .....	2
实验工具 .....	2
实验步骤 .....	3
1、编译/安装 BUSYBOX (BUSYBOX-1.16.0.TAR.BZ2) , .....	3
1.添加动态库与加载器, .....	5
3. 构建 ETC 目录, .....	6
3.1 创建 <i>etc/inittab</i> 文件.....	6
3.2 创建 <i>etc/fstab</i> 文件.....	7
3.3 创建 <i>etc/init.d/rcS</i> 文件.....	7
1.构建 DEV 目录, .....	8
2.构建其他目录, .....	9
3.制作 YAFFS 文件系统映像文件 .....	9
7. 移植 .....	10

## 实验内容

能在 linux2.6.32 与 ARM 平台运行的根文件系统的制作

## 实验目的

掌握简单 linux 根系统的制作过程，搭建开发平台

## 实验材料

交叉编译器： arm-linux-gcc-4.3.2

Busybox:           busybox-1.16.0.tar.bz2

## 实验工具

Linux 操作系统： Ubuntu9.04

ARM 交叉编译套件

## 实验步骤

### 1 、 编 译 / 安 装 Busybox (busybox-1.16.0.tar.bz2) ,

具体步骤:

创建制作根文件系统的路径 (假定为  
/root/myroot/rootfs)

```
#> mkdir /root /myroot
```

```
#> mkdir /root /myroot/rootfs
```

将 busybox 压缩包解压到/root /myroot 目录下

```
#>tar jxvf busybox-1.16.0.tar.bz2 -C /root/myroot/
```

进入解压后软件所在目录

```
#> cd busybox-1.16.0
```

修改 Makefile 文件

```
#> vi Makefile
```

修改前:

```
190 ARCH                ?= $(SUBARCH)
```

```
164 CROSS_COMPILE ?=
```

修改后:

```
164 CROSS_COMPILE ?= arm-linux-
```

```
190 ARCH                ?= arm
```

添加临时环境变量

```
#> export PATH=/usr/local/arm/4.3.2/bin:$PATH
```

配置 Busybox

```
#> make menuconfig
```

在 Busybox Settings ----->

Installation Options ----->

(./\_install) BusyBox installation prefix

输入/root/myroot/rootfs,

其他默认原设置, 保存退出

编译 busybox

```
#> make
```

安装 busybox

```
#> make install
```

安装结束后, 在目录/root/myroot/rootfs/下将生成 4 个文件: bin、sbin、usr、linuxrc

## 1. 添加动态库与加载器,

具体步骤:

动态库与加载器的位置是:

/usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/  
lib/

注：若是 4.4.3 则，位置是：

/usr/local/arm/4.4.3/arm-none-linux-gnueabi/lib

创建库目录

```
#> mkdir -p /root/myroot/rootfs/lib
```

进入 lib 目录

```
#> cd /root/myroot/rootfs/lib/
```

添加库函数

```
#> cp
```

/usr/local/arm/4.3.2/arm-none-linux-gnueabi/libc/armv4t/  
lib/\* .

### 3. 构建 etc 目录,

具体步骤:

#### 3.1 创建 etc/inittab 文件

进入根文件系统目录 (/root/myroot/rootfs)

```
#> cd /root/myroot/rootfs/
```

创建 etc 目录

```
#> mkdir etc
```

进入 etc 目录

```
#> cd /root/myroot/rootfs/etc
```

创建 inittab 文件

```
#> vi inittab
```

内容如下:

```
# /etc/inittab
::sysinit:/etc/init.d/rcS
::askfirst:/bin/sh
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
```

#### 3.2 创建 etc/fstab 文件

进入根文件系统 etc 目录 (/root/myroot/rootfs/etc)

```
#> cd /root/myroot/rootfs/etc/
```

创建 fstab 文件

```
#> vi fstab
```

内容如下：

```
# device mount-point type options dum fsck order
proc /proc proc defaults 0 0
tmpfs /tmp tmpfs defaults 0 0
sysfs /sys sysfs defaults 0 0
```

### 3.3 创建 etc/init.d/rcS 文件

创建/root/myroot/rootfs/etc/init.d 目录

```
#> mkdir -p /root/myroot/rootfs/etc/init.d
```

进入/root/myroot/rootfs/etc/init.d 目录

```
#> cd /root/myroot/rootfs/etc/init.d
```

创建 etc/init.d/rcS 文件

```
#> vi rcS
```

内容如下：

```
#!/bin/sh
```

```
mount -a
```

```
echo /sbin/mdev > /proc/sys/kernel/hotplug
```

```
mdev -s
```

修改文件属性，使之能执行

```
#>
```

```
chmod +x /root/myroot/rootfs/etc/init.d/rcS
```

## 1. 构建 dev 目录，

具体步骤：



创建/root/myroot/rootfs/dev 目录

```
#> mkdir /root/myroot/rootfs/dev
```

进入/root/myroot/rootfs/dev 目录

```
#> cd /root/myroot/rootfs/dev
```

静态创建设备文件

```
#> mknod console c 5 1
```

```
#> mknod null c 1 3
```

## 2. 构建其他目录，

具体步骤：

进入/root/myroot/rootfs/

```
#> cd /root/myroot/rootfs/
```

创建其他目录

```
#> mkdir proc mnt tmp sys root
```

## 3. 制作 yaffs 文件系统映像文件

具体步骤：

进入/root/myroot/

```
#> cd /root/myroot/
```

复制制作 yaffs 文件工具

```
#> cp /usr/sbin/mkyaffs2image-128M .
```

制作 yaffs2 映像文件

```
#> ./mkyaffs2image-128M rootfs root.image
```

最后在目录/root/myroot/下，生成 yaffs2 映像文件

——root.image

## 7. 移植

将 root.image 移植到开发板上，重启即可

---