

# 嵌入式系统设计实验报告

## 实验五

### 通过 busybox 制作根文件系统

学    号：PE20060014

姓    名：王晨

专    业：计算机科学与技术

指导老师：张辉

2020 年 1 月 8 日

## 一、 实验要求

通过 busybox 创建文件系统

检查生成的文件系统

在开发板上启动自己生成的文件系统

## 二、 实验条件

- 1、 硬件条件：Macbook pro
- 2、 软件条件：Mac OS Big Sur 11.01

VMware fusion pro 虚拟机

Ubuntu 20.04.1 64 位

## 三、 实验过程

### 一、 使用 busybox 创建文件系统

这部分参考 <https://www.cnblogs.com/electronic/p/11146614.html> 进行制作。

1. 首先到官网 <https://busybox.net/downloads/> 下载 busybox, 这里选了 1.14.4 (尝试了高版本 1.30.0 失败)

2. 配置交叉编译工具

打开 Makefile 修改内容:

164 行 CROSS\_COMPILE ?= 改为: CROSS\_COMPILE ?= arm-linux-

190 行 ARCH ?= \$(SUBARCH) 改为: ARCH ?= arm

以及:

解决方法：

a.修改busybox-1.7.0 顶层Makefile 405行:

```
config%config: scripts_basic outputmakefile  
FORCE
```

改为:

```
%config:scripts_basic outputmakefile FORCE
```

b.修改busybox-1.7.0 顶层Makefile 1242行:

```
/%/: prepare scripts FORCE
```

改为:

```
%/:prepare scripts FORCE
```

1. 14.4 版本的 busybox 还需要修改一处地方才能正常 make，但是我忘记是哪里了。

3. 导出配置文件 make defconfig

4. 自定义图形化配置界面

make menuconfig，添加 insmod rmmod modinfo 等命令，1.14.4busybox 有些命令没有，不用管。

5. 编译

```
make all -j4 V=1
```

```
make install
```

编译完成后可以看到在/busybox1.14.4 目录下有文件夹/\_install

6. 构建基础文件和目录

进入/\_install 目录创建根文件系统下所需目录:

```
_install$ mkdir dev etc home lib mnt proc root sy  
s tmp var -p
```

在\_install/etc 目录下创建 inittab 文件: \_install \$ touch etc/inittab

并将如下代码复制如 inittab 文件中:

```
#this is run first except when booting in single-user mode.
::sysinit:/etc/init.d/rcS
# /bin/sh invocations on selected ttys
::respawn:-/bin/sh
# Start an "askfirst" shell on the console (whatever that may be)
::askfirst:-/bin/sh
# Stuff to do when restarting the init process
::restart:/sbin/init
# Stuff to do before rebooting
::ctrlaltdel:/sbin/reboot
::shutdown:/sbin/swapoff -a
```

继续创建目录及文件如下：

- (1) `_install $ mkdir etc/init.d/ -p`
- (2) `_install $ touch etc/init.d/rcS`
- (3) `_install $ gedit etc/init.d/rcS`      在此文件内添加内容如下：

```
#!/bin/sh
#This is the first script called by init process
/bin/mount -a
echo /sbin/mdev>/proc/sys/kernel/hotplug
mdev -s
```

- (4) `_install $ touch etc/fstab`
- `_install $ gedit etc/fstab`      文件中添加内容如下：

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

- (5) `_install $ touch etc/profile`
- `_install $ gedit etc/profile`      文件中添加内容如下：

```
#!/bin/sh
export HOSTNAME=farsight
export USER=root
export HOME=root
export PS1="[$USER@$HOSTNAME \W]\# "
#export PS1="[\[\033[01;32m\]$USER@\[\033[00m\]\[\033[01;34m\]$HOSTNAME\[\033[00m\ \W]\$ "
PATH=/bin:/sbin:/usr/bin:/usr/sbin
LD_LIBRARY_PATH=/lib:/usr/lib:$LD_LIBRARY_PATH
export PATH LD_LIBRARY_PATH
```

## 7. 添加动态库支持

```
_install$ cp/usr/local/arm/gcc-4.6.4/arm-none-linux-gnueabi/sysroot/lib
/* lib -ra
```

这里添加了 4.6.4 版本的动态库，是不行的，最后烧写的时候会报错

Failed to execute /linuxrc. Attempting defaults...

或者 Kernel panic - not syncing: Attempted to kill init!

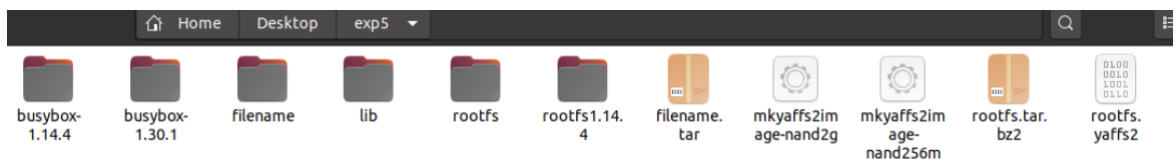
必须采用 4.3.2 版本的动态库，猜测可能是因为需要和飞凌给的 uboot 和系统镜像匹配。

因此我直接将飞凌的文件系统下 lib 的动态库都 cp 到了 busybox 文件系统下的 lib 目录下。最后可以烧写成功。

## 8. 制作 yaffs2 文件系统映像

这里按照开发手册制作即可。将文件系统映像制作工具，mkyaffs2image-nand2g 拷贝到你的工作目录下，输入命令（注意这里使用 256m 版本工具是不行的）：

#./mkyaffs2image-nand2g \_install rootfs.yaffs2 （\_install 下文件一定要开放权限否则烧写系统可能失败）



得到 rootfs.yaffs2 后同实验 1 重新烧写系统即可。

最后在串口看到如下文件系统即为完成：

