

《嵌入式系统》实验报告 4

学号： 21013134 姓名： 徐昊博 班级： 计 213 日期： 2024.5.31

成绩： _____ 指导教师： 罗飞

实验名称： 嵌入式文件系统的构造	实验地点：
------------------	-------

实验仪器： ECS 云计算环境（提供 Linux 实验环境）

- 一、实验目的：
- 1、 了解嵌入式操作系统中文件系统的类型和作用
 - 2、 了解 JFFS2 文件系统的优点及其在嵌入式系统中的作用
 - 3、 掌握利用 BusyBox 软件制作嵌入式文件系统的方法
 - 4、 掌握嵌入式 Linux 文件系统的挂载过程

二、实验内容

注： 如果 chrome 打不开， 下载云平台上的 firefox 并安装， 可以打开虚拟仿真实验。

1.简述利用 BusyBox 制作嵌入式文件系统的步骤。

（1） 建立工作目录， 并获取及解压文件系统源码

```
ecust@Ubuntu:~$ mkdir workplace
ecust@Ubuntu:~$ cd workplace
解压 busybox (/home/ecust/samba_share/embed/busybox/busybox-1.17.3.tar.bz2)
使用 tar 命令进行解压
ecust@Ubuntu ~/workplace$ tar xvf /home/ecust/samba_share/embed/busybox/busybox-1.17.3.tar.bz2
进入源码目录busybox-1.17.3
ecust@Ubuntu ~/workplace$ cd busybox-1.17.3
```

（2） 配置源码

```
ecust@Ubuntu ~/workplace/busybox-1.17.3$ make menuconfig， 显示界面如下：
Busybox Settings --->
    Build Options --->
        [*] Build BusyBox as a static binary (no shared libs)
        [] Force NOMMU build
        [] Build with Large File Support (for accessing files > 2 GB)
        (arm-none-linux-gnueabi-) Cross Compiler prefix
        () Additional CFLAGS
```

（3） 编译

```
ecust@Ubuntu ~/workplace/busybox-1.17.3$ make
```

（4） 安装

```
busybox 默认安装路径为源码目录下的_install
ecust@Ubuntu ~/workplace/busybox-1.17.3$ make install
进入安装目录下
ecust@Ubuntu ~/workplace/busybox-1.17.3$ cd install
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ ls
显示内容目录如下：
```

bin linuxrc sbin usr

(5) 创建其他需要的目录

主要目录: dev etc mnt proc var tmp sys root , 指令如下:

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir dev
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir etc
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir mnt
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir proc
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir var
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir tmp
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir sys
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkdir root
```

(6) 添加库

将工具链中的库拷贝到 _install 目录下

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ cp /home/ecust/workplace/toolchain/arm-none-linux-gnueabi/lib/ ./ -a
```

删除静态库和共享库文件中的符号表

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ rm lib/*.a
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ arm-none-linux-gnueabi-strip lib/*
```

删除不需要的库, 确保所有库大小不超过4M

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ du -mh lib/
```

(7) 添加系统启动文件

先进入etc文件夹: ecust@Ubuntu ~/workplace/busybox-1.17.3/_install\$ cd etc

①在etc下添加文件inittab: ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc\$ vim inittab

按下“i”进入编辑模式, 编辑内容如下:

```
#this is run first except when booting in single-user mode.

::sysinit:/etc/init.d/rcS

# /bin/sh invocations on selected ttys

#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
```

编辑完成后先按esc, 再按“:wq”, 退出编辑并进行保存

②在etc下添加文件fstab: ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc\$ vim fstab

按下“i”进入编辑模式, 编辑内容如下:

文件内容如下:

#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

编辑完成后先按esc, 再按“:wq”, 退出编辑并进行保存

这里挂载的文件系统有三个proc、sysfs和tmpfs。在内核中proc和sysfs默认都支持, 而tmpfs是没有支持的, 我们需要添加tmpfs的支持

修改内核配置:

File systems --->

Pseudo filesystems --->

[*] Virtual memory file system support (former shm fs)

[*] Tmpfs POSIX Access Control Lists

重新编译内核 (实验3内容)

在etc下创建init.d目录, 并在init.d下创建rcS文件

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc$ mkdir init.d
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc$ cd init.d
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc/init.d$ vim rcS
```

按下 “i” 进入编辑模式, 编辑内容如下:

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

```
# This is the first script called by init process
```

```
/bin/mount -a
```

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

```
/sbin/mdev -s
```

编辑完成后先按esc, 再按 “:wq”, 退出编辑并进行保存

为rcS添加可执行权限:

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc/init.d$ chmod +x init.d/rcS
```

在etc下添加profile文件: ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc/init.d\$ vim profile

按下 “i” 进入编辑模式, 编辑内容如下:

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

```
export HOSTNAME=farsight
```

```
export USER=root
```

```
export HOME=root
```

```
export PS1="[$USER@$HOSTNAME \W]\# "
```

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
```

```
LD_LIBRARY_PATH=/lib:/usr/lib:$LD_LIBRARY_PATH
```

```
export PATH LD_LIBRARY_PATH
```

编辑完成后先按esc, 再按 “:wq”, 退出编辑并进行保存

(8) cramfs文件系统镜像制作

由于系统提供制作cramfs文件系统的工具, 可以直接使用。具体操作如下:

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc/init.d$ cd ..
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install/etc/$ cd ..
```

```
ecust@Ubuntu ~/workplace/busybox-1.17.3/_install$ mkfs.cramfs /home/ecust/workplace/busybox-1.17.3/_install rootfs.cramfs
```

三、思考：

- 1、什么是根文件系统？查看当前所使用系统的文件系统目录与所制作的根文件系统目录，说明它们的异同；能否在建立根文件系统时中构建 sys、proc 子目录？请说明原因。

当前使用系统的文件系统目录

是一个典型的 Linux 系统的文件系统目录结构，包含多个子目录，每个子目录都有特定的用途：

/bin - 存放基本的用户命令。

/sbin - 存放系统管理命令。

/usr/bin - 存放更多的用户命令。

/usr/sbin - 存放更多的系统管理命令。

/etc - 存放配置文件。

/lib - 存放基本的共享库。

/usr/lib - 存放更多的共享库。

/dev - 存放设备文件。

/proc - 虚拟文件系统，存放系统和进程信息。

/sys - 虚拟文件系统，存放系统设备和硬件信息。

/home - 存放用户主目录。

/var - 存放可变数据文件（如日志）。

/tmp - 存放临时文件。

/mnt - 存放临时挂载点。

/root - 存放超级用户的主目录。

制作的根文件系统目录

这是一个针对嵌入式 Linux 系统的简化根文件系统，通常包括以下目录：

/bin - 存放基本的用户命令（通常由 BusyBox 提供）。

/sbin - 存放基本的系统管理命令（通常由 BusyBox 提供）。

/usr/bin - 存放更多的用户命令（由 BusyBox 提供）。

/usr/sbin - 存放更多的系统管理命令（由 BusyBox 提供）。

/etc - 存放配置文件。

/lib - 存放基本的共享库（从工具链拷贝）。

/dev - 存放设备文件（通常手动创建基础设备节点）。

/proc - 虚拟文件系统的挂载点（启动后挂载）。

/sys - 虚拟文件系统的挂载点（启动后挂载）。

/mnt - 存放临时挂载点。

/var - 存放可变数据文件（如日志）。

/tmp - 存放临时文件。

/root - 存放超级用户的主目录。

相同点：

基本目录结构：两个文件系统都有类似的基本目录结构，如 /bin, /sbin, /etc, /lib, /dev, /proc, /sys, /tmp, /var, /mnt, 和 /root。

功能：两个文件系统目录基本上具有相同的功能，例如 /bin 和 /sbin 存放用户命令和系统管理命令，/etc 存放配置文件，/lib 存放共享库文件，等等。

虚拟文件系统挂载点：两个文件系统都包含 /proc 和 /sys 作为虚拟文件系统的挂载点。

不同点：

目录内容：

当前使用的系统文件系统目录包含完整的 Linux 系统所需的所有文件和目录，包括许多应用程序、库文件和用户数据。

制作的根文件系统目录是一个最小化的系统，只包含启动和运行嵌入式系统所需的基本文件和目录，通常由 BusyBox 提供基本命令和工具。

库文件：

当前使用的系统文件系统目录可能包含大量的共享库文件，支持各种应用程序和服务。制作的根文件系统目录中的库文件通常经过精简，只保留嵌入式系统运行所需的基本库文件，确保库文件的总大小不超过 4M。

设备文件：

当前使用的系统文件系统目录中的 /dev 目录包含所有可能的设备文件，由 udev 或 mdev 动态管理。制作的根文件系统目录中的 /dev 目录可能只包含手动创建的基础设备文件，并在启动时由 mdev 动态管理其他设备文件。

用户数据：

当前使用的系统文件系统目录中有用户的主目录（如 /home），存放用户的个人数据和配置文件。制作的根文件系统目录通常没有或只有最小的用户数据存储需求，可能不包含 /home 目录，或者仅包含超级用户的主目录 /root。

能否在建立根文件系统时中构建 sys、proc 子目录？请说明原因。

在建立根文件系统时，必须构建 sys 和 proc 子目录。这些目录对于系统的正常运行是至关重要的，原因如下：

挂载点：

/proc 和 /sys 是虚拟文件系统，它们不是存储在磁盘上的实际文件，而是由内核提供的接口，供用户空间程序访问内核数据和系统信息。

/proc 目录包含关于系统和进程的信息，例如内存使用情况、正在运行的进程等。系统启动后，内核会将 /proc 挂载到 /proc 目录中。

/sys 目录包含关于系统设备和硬件的信息。系统启动后，内核会将 /sys 挂载到 /sys 目录中。

必要性：

没有这些挂载点，许多系统工具和应用程序将无法正常工作，因为它们依赖于从这些虚拟文件系统获取系统和硬件信息。

例如，/proc/mounts 文件包含了当前挂载的文件系统信息，许多系统管理工具需要读取这个文件来获取挂载状态。

/sys 目录提供了对内核设备模型的访问，许多硬件管理工具需要从中获取信息。

兼容性：

大多数 Linux 发行版和嵌入式系统都期望 /proc 和 /sys 存在，以确保一致性和兼容性。如果这些目录缺失，可能会导致无法预期的行为。