

嵌入式 Linux2.6 根文件系统制作过程

一：目录：

- (一)：使用busybox制作Linux根文件系统
- (二)：nfs服务器配置过程——RedHatAS5环境
- (三)：开发板通过nfs服务挂载虚拟机指定目录
- (四)：通过nfs服务挂载根文件系统
- (五)：cramfs格式根文件系统制作
- (六)：yaffs格式根文件系统制作
- (七)：busybox自带telnet服务器
- (八)：busybox自带的FTP服务器
- (九)：busybox自带的tftp服务器
- (十)：Linux 下实现U盘、SD卡自动挂载功能

(一)：使用busybox制作Linux根文件系统

构建最简单的嵌入式 Linux 根文件系统，下面我们一步步教大家做起。。

硬件环境：

S3C2440

软件环境：

Busybox-1.16.1 <http://www.rayfile.com/zh-cn/files/2e477e14-c1bb-11e0-b287-0015c55db73d/>

cross-4.3.2 <http://www.rayfile.com/zh-cn/files/99e727f3-c1a5-11e0-822a-0015c55db73d/>

STEP 1: 构建目录结构

创建根文件系统目录，主要包括以下目录

```
/dev /etc /lib /usr /var /proc /tmp /home /root /mnt  
/bin /sbin /sys
```

```
#mkdir /home/rootfs
```

```
#cd /home/rootfs
```

```
#mkdir dev etc lib usr var proc tmp home root mnt sys
```

STEP 2: 使用 busybox 构建/bin /sbin linuxrc

进入 busybox-1.16.1 目录，执行

```
#make defconfig
```

```
#make menuconfig
```

```
Busybox Setting ----->
```

```
Build Options ----->
```

```
//1 选择将busybox进行静态编译
```

```
[*]Build BusyBox as a static binary (no shared libs)
```

```
//2. 指定交叉编译器为
```

```
(/usr/local/arm/4.3.2/bin/arm-linux-)Cross Compiler prefix
```

```
Installation Options ----->
```

```
//3. 选择上 Don' t use /usr
```

```
Busybox Library Tuning--->
```

```
[*]Username completion
```

```
[*]Fancy shell prompts
```

```
[*]Query cursor position from terminal
```

```
//4. 编译出的busybox的shell命令解释器支持显示当前路径及主机信
```

息

保存退出

```
#make
#make install
```

在 busybox 目录下会看见 _install 目录，里面有/bin /sbin linuxrc 三个文件
将这三个目录或文件拷到第一步所建的 rootfs 文件夹下。

```
#cp bin/ sbin/ linuxrc /home/rootfs -ra
```

切记一定要带上-a的参数，因为bin目录里大部分都是链接，如果不带-a的参数，拷过去之后会做相应的复制，不再是链接的形式

STEP 3 构建etc目录:

1) 进入根文件系统rootfs的etc目录，执行如下操作:

```
拷贝Busybox-1.16.1/examples/bootfloppy/etc/* 到当前目录下
#cp -r busybox-1.16.1/examples/bootfloppy/etc/* rootfs/etc
```

```
修改 inittab, 把第二项改为::respawn:-/bin/login
```

```
删除第三、第四行代码
```

2) 拷贝虚拟机上的/etc/passwd, /etc/group, /etc/shadow到rootfs/etc下

```
# cp /etc/passwd rootfs/etc
# cp /etc/group rootfs/etc
# cp /etc/shadow rootfs/etc
```

对以下三个文件修改，只保存与root相关的项，根据具体情况内容会有所不同。

修改passwd为root:x:0:0:root:/root:/bin/sh，即只保存与root相关项，而且最后改成/bin/ash。

```
修改group为root:x:0:root
```

```
修改shadow为
```

```
root:$1$x9yv1WlB$abJ2v9j0l0c9xW/y0QwPs.:14034:0:99999:7:::
```

登陆开发板时需输入用户名密码，同虚拟机相同

3) 修改profile

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin           //可执行程序 环境变量
export LD_LIBRARY_PATH=/lib:/usr/lib         //动态链接库 环境变量

/bin/hostname sunplusedu

USER="`id -un`"

LOGNAME=$USER

HOSTNAME=' /bin/hostname'

PS1=' [\u@\h \W]# '                          //显示主机名、当前路径等
信息:
```

4) 修改 etc/init.d/rc.S文件

```
/bin/mount -n -t ramfs ramfs /var

/bin/mount -n -t ramfs ramfs /tmp

/bin/mount -n -t sysfs none /sys

/bin/mount -n -t ramfs none /dev

/bin/mkdir /var/tmp

/bin/mkdir /var/modules

/bin/mkdir /var/run

/bin/mkdir /var/log

/bin/mkdir -p /dev/pts                          //telnet服务需要

/bin/mkdir -p /dev/shm                          //telnet服务需要
```

```
echo /sbin/mdev > /proc/sys/kernel/hotplug//USB自动挂载需要
```

```
/sbin/mdev -s //启动mdev在/dev下自动创建设备文件节点
```

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

5)修改etc/fstab文件，增加以下文件

```
none /dev/pts devpts mode=0622 0 0
tmpfs /dev/shm tmpfs defaults 0 0
```

STEP 4 构建lib目录:

1)#cd

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

将以下动态库拷贝到rootfs/lib下

```
#cp *so* rootfs/lib -a
```

2)#cd

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

将以下动态库拷贝到rootfs/lib下

```
#cp ./libstdc++.so.* rootfs/lib -a
```

(二)：nfs服务器配置过程——RedHatAS5环境

nfs 文件系统介绍

NFS是Network File System的简写,即网络文件系统, NFS允许一个系统在网络上与他人共享目录和文件。通过使用NFS, 用户和程序可以像访问本地文件一样访问远端系统上的文件。

在嵌入式 Linux 系统中有两种应用:

1、通过网络把虚拟机的某个目录挂载到开发板某个目录下，实现虚拟机上修改、编译，开发板上运行、验证

2、通过网络直接挂载保存在虚拟机上的嵌入式根文件系统，实现根文件系统及应用的验证及修改

无论哪种用法，虚拟机或物理机上的 Linux 都作为服务器存在，配置过程都一样，我们以 Red Hat AS5 为例讲解其配置过程，其它版本的 Linux 请自行查找资料进行配置。

RedHatAS5 nfs服务器配置

1、在系统安装时选择以下选项：

2、安装完成后在系统菜单中可以看到以下选项：

2、设置开启 NFS 服务：

系统——管理——服务器设置——服务
依次选上 nfs 服务前的对勾、开始、保存

3、配置 nfs 服务器共享文件夹

4、nfs 服务重启（可选）

有时候时间长了，由于网络环境原因，nfs 服务不能正常执行，可以执行以下命令，重启 nfs 服务。

```
#service nfs restart
```

5、服务测试——本机挂本机

在终端中执行以下命令，用以测试 nfs 服务器的正确工作与否

```
#mount -o nolock 192.168.220.XX:/home /mnt
```

//在本机或其他机子运行，挂载 home 目录，访问/mnt 目录，如果能看到/home 目录下的文件，则证明 nfs 服务器正常工作。

说明:

因为前面设置的是/目录，所以所有的目录都可以挂载。

(三)：开发板通过nfs服务挂载虚拟机指定目录

经过前面的配置，nfs 服务器已经可以正常工作，在开发板上挂载虚拟机目录，做以下配置即可。

1、首先保证开发板跟虚拟机、windows 在一个网段

例如：开发板的 IP: 192.168.1.2

windows 的 IP:192.168.1.3

虚拟机的 IP: 192.168.1.4

保证开发板可以 ping 通虚拟机，证明网络环境正确

```
#ping 192.168.1.4
```

说明：如果网络不通，可以尝试关掉 windows 和 Linux 中的防火墙，包括一些杀毒软件、360 卫士等。

开发板设置 IP 可以通过以下命令完成：

```
#ifconfig eth0 192.168.1.2
```

2、在开发板中执行以下命令即可

```
#mount -o nolock,wisze=1024,rsiz=1024 192.168.1.4:/home /mnt
```

wsize,rsiz 后面的数字代表每次读写的最大字节数，不写这个在某些网络环境下会因为数据量过大造成网络断线。

(四)：通过nfs服务挂载根文件系统

前面我们已经完成了根文件系统的制作，本节介绍一下如何通过 nfs 服务远程访问虚拟机上的根文件系统，这样可以有效提高根文件系统的制作验证

在 vivi 和 u-boot 中的设置不太一样，我们分别做以下说明：

1、修改vivi启动参数，从虚拟机上启动根文件系统：

```
paramset linux_cmd_line "noinitrd root=/dev/nfs
nfsroot=/home/rootfs/(修改为实际的目录),rsize=1024,wsiz=1024
ip=192.168.220.11(开发板地址可以随便填):192.168.220.xx(虚拟机的地
址):192.168.220.254:255.255.255.0::eth0:offinit=/linuxrc
console=ttySAC0"
```

```
例：vivi> paramset linux_cmd_line "noinitrd root=/dev/nfs
nfsroot=/home/dengwei/rootfs/,rsize=1024,wsiz=1024
ip=192.168.220.123:192.168.220.56:192.168.220.254:255.255.255.0::eth0:o
ffinit=/linuxrc console=ttySAC0"
```

```
vivi> param save
```

2、修改u-boot启动参数，从虚拟机上启动根文件系统：

```
setenvbootargs 'noinitrd root=/dev/nfs nfsroot=/home/rootfs/(修改为实际
的目录),rsize=1024,wsiz=1024ip=192.168.220.11(开发板地址可以随便
填):192.168.220.xx(虚拟机的地址):192.168.220.254:255.255.255.0::eth0:off
init=/linuxrc console=ttySAC0'
```

```
例：u-boot> paramset linux_cmd_line "noinitrd root=/dev/nfs
nfsroot=/home/dengwei/rootfs/,rsize=1024,wsiz=1024
ip=192.168.220.123:192.168.220.56:192.168.220.254:255.255.255.0::eth0:o
ffinit=/linuxrc console=ttySAC0"
```

```
u-boot> saveenv
```


注意：vivi修改参数使用双引号，u-boot使用单引号

重启开发板即可

(五)：cramfs格式根文件系统制作

cramfs 是 Linux 的创始人 Linus Torvalds 参与开发的一种只读的压缩文件系统，在 cramfs 文件系统中，每一页(4KB)被单独压缩，可以随机页访问，其压缩比高达 2:1，为嵌入式系统节省大量的 Flash 存储空间，使系统可通过更低容量的 FLASH 存储相同的文件，从而降低系统成本，cramfs 文件系统以压缩方式存储，在运行时解压缩，不支持应用程序以 XIP 方式运行，所有的应用程序要求被拷到 RAM 里去运行。

制作 cramfs 格式的根文件系统需要以下工具:cramfs-1.1.tar.gz

下载地址为:<http://www.rayfile.com/zh-cn/files/8d9dd475-c1ba-11e0-b947-0015c55db73d/>

使用方法：

```
#tar zxvf cramfs-1.1.tar.gz
```

```
#cd cramfs-1.1
```

```
#make //会生成以下两个文件，mkcramfs和cramfsck
```

1、mkcramfs 工具用来创建 CRAMFS 文件系统

```
#mkcramfs dirname(文件夹) root.cramfs(生成的镜像)
```

2、cramfsck 工具用来进行 CRAMFS 文件系统的释放和检查

```
#cramfsck root.cramfs -x dir(文件夹)
```

将生成的镜像通过vivi或u-boot烧入FLASH即可

(六)：yaffs格式根文件系统制作

yaffs/yaffs2 是专为嵌入式系统使用 NAND 型闪存而设计的一种日志型文件系统。与 jffs2 相比，它减少了一些功能(例如不支持数据压缩)，所以速度更快，挂载时间很短，对内存的占用较小。

yaffs 是跨平台的文件系统，除了 Linux 和 eCos，还支持 WinCE, pSOS 和 ThreadX 等

yaffs 与 yaffs2 的主要区别在于：

前者仅支持小页(512 Bytes) NAND 闪存，后者则可支持大页(2KB) NAND 闪存。

yaffs2 在内存空间占用、垃圾回收速度、读/写速度等方面均有大幅提升。

下载地址为：<http://www.rayfile.com/zh-cn/files/b327a423-c1ba-11e0-bbe3-0015c55db73d/>

1、Yaffs 工具包 yaffs.tar.gz 的使用：

```
#tar zxvf yaffs.tar.gz
```

```
#cd yaffs
```

```
#cd utils
```

```
#make //会产生mkyaffsimage
```

2、用 mkyaffsimage 制作 yaffs 文件系统镜像：

```
#mkyaffsimage usr usr.yaffs
```

与 cramfs 不同,这个过程是不可逆的
最后将生成的镜像通过 vivi 或 u-boot 烧入 FLASH 即可。

(七)：busybox自带telnet服务器

telnet 服务器：

```
#telnetd
```

//默认的端口为 23

注意：必须在/dev目录下创建 pts目录，并挂载devpts文件系统

在windows中开始—》运行—》cmd 启动MS-DOS

```
>telnet 192.168.1.1(开发板的IP)
```

输入用户名及密码即可登录

如果想开机运行

在/etc/init.d/rcS中加入

/sbin/telnetd即可

(八) : busybox自带的FTP服务器

ftp 服务器:

启动方法 1:

```
#tcpsvd 0 21 ftpd -w /root &
```

// 上面的 0 表示对所有 ip 地址都进行侦听

// ftpd -w 这里的参数-w 表示 client 可以对目录执行写操作

// 21 指定 ftp 服务器的默认端口

// 可以使用-t 和-T 参数设置 client 在没有任何操作的最大时间之后 ftpd 主动断开 client 连接,

// 默认-t 为 2 分钟=2 * 60, -T 为 1 小时=1 * 60 * 60

启动方法 2:

```
#vi /etc/inetd.conf //增加以下代码
```

```
21 stream tcp nowait root ftpd ftpd -w /root
```

```
#inetd
```

//inetd 会执行/etc/inetd.conf 脚本中的命令行, 这样 ftpd 就作为 daemon 运行起来了.

(九)：busybox自带的tftp服务器

a、tftp server 服务器配置与使用

方法 1:

```
#udpsvd -vE 0 69 tftpd -c /root&
```

//上面的 0 表示对所有 ip 地址都进行侦听

方法 2:

```
#vi /etc/inetd.conf
```

```
69 dgram udp nowait root tftpd tftpd -c /root
```

```
#inetd
```

测试使用:

我们可以通过 windows 下 tftp32 软件进行下载、上传文件测试

将 tftp32 设为客户端，设置主机 ip，输入待发送文件名点击 put 或准备接收的文件名点击 get，如下图所示：

b、tftp client 的使用

tftp32 作为服务器

1.将开发板上的 **hello.c** 文件 **put** 到 PC 机端 **tftpd32** 的根目录下,并且重命名为 **world.c**

```
#tftp -l hello.c -r world.c -p (tftp 服务器的 ip 地址)
```

2.将 PC 上 **tftpd32** 目录下的 **world.c** 文件 **get** 到开发板机端,并且重命名为 **hello word.c**

```
# tftp -l hello_world.c -r world.c -g (tftp 服务器的 ip 地址)
```

(十) : Linux 下实现 U 盘、SD 卡自动挂载功能

在 Linux 中我们经常用到 U 盘、SD 卡的挂载问题,每次都手动挂载或卸载非常麻烦,我们可以采取以下方法实现自动挂载或卸载 U 盘、SD 卡

1、首先在 `/etc/init.d/rsC` 中加入以下语句

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

2、在 `/etc/udev/rules.d` 的文件,包含以下内容

```
sd[a-z][0-9]      0:0 666      @(/etc/hotplug/insert.sh $MDEV
$SUBSYSTEM)
sd[a-z]           0:0 666      $(/etc/hotplug/remove.sh $MDEV
$SUBSYSTEM)
ub[a-z][0-9]     0:0 666      @(/etc/hotplug/insert.sh $MDEV
$SUBSYSTEM)
ub[a-z]          0:0 666      $(/etc/hotplug/remove.sh $MDEV
$SUBSYSTEM)
mmcblk[0-9]p[0-9] 0:0 666      @(/etc/hotplug/insert.sh $MDEV
$SUBSYSTEM)
```

```
mmcblk[0-9]          0:0 666          $(/etc/hotplug/remove.sh $MDEV
$SUBSYSTEM)
```

3、在/etc/下建立以下文件夹hotplug,并在hotplug目录下建立以下文件,包含以下内容

insert.sh:

```
if [ -n "$1" ] ; then
    if [ -b /dev/$1 ]; then

        if [ ! -d /media ]; then
            mkdir -p /media
        fi

        if [ ! -d /media/$1 ]; then
            mkdir -p /media/$1
        fi

        mount /dev/$1 /media/$1
    fi
fi

if [ $? -ne 0 ]; then
    rm -rf /media/$1

    fi
fi
```

remove.sh:

```
MOUNTS=$(mount | grep $1 | cut -d' ' -f3)
umount $MOUNTS
rm -rf $MOUNTS
```

插入U盘、SD等设备,即可在根目录的/media目录下建立相应的文件夹,挂载对用的磁盘文件。

以上文章来自凌阳教育嵌入式培训网提供，更多嵌入式学习资料免费下载：
<http://emb.sunplusedu.com/download/>