

制作busybox根文件系统

实验材料：

busybox-1.30.0源码 (下载地址：<https://busybox.net/downloads/>)

qemu硬件模拟器

ubuntu虚拟机

实验概述：

下载busybox源码之后解压，进入到主目录进行相关配置再编译和安装，此时会产生新的目录_install，里面包含编译和安装产生的文件，我们在其中添加一些新的文件和文件夹并配置其内容，最后将_install目录下的所有文件打包成内核可以识别并运行的文件系统镜像，最终将其与内核一起用qemu运行。

解压配置并编译安装

相关命令：

```
tar xvf busybox-1.30.0.tar.bz2          //解压

make menuconfig                          //进入图像化配置界面
Busybox Settings  --->
  Build Options  --->
    [*] Build BusyBox as a static binary (no shared libs)      //配置文件
系统源码，将其编译成静态包，以便在运行时不依赖其他的库文件
make && make install                    //将源码编译并安装
注：所有的以上配置操作都在busybox的最上级目录下进行
```

在_install目录下进行文件创建和配置

相关命令：

```
首先进入到_install目录，在该目录下进行如下操作
# mkdir etc dev mnt                      //创建3个文件夹(etc,dev,mnt)
# mkdir -p etc/init.d/                  //-p参数 递归创建多级目录
# vim etc/fstab                          //在etc下新建文件fstab,并将内容设定为
如下
proc /proc proc defaults 0 0
temps /tmp rpoc defaults 0 0
none /tmp ramfs defaults 0 0
sysfs /sys sysfs defaults 0 0
mdev /dev ramfs defaults 0 0
# vim etc/init.d/rcS                      //etc/init.d下新建文件rcs,并将内容设
置如下(很显然此文件是一个脚本)
```

```

mkdir -p /proc
mkdir -p /tmp
mkdir -p /sys
mkdir -p /mnt
/bin/mount -a
mkdir -p /dev/pts
mount -t devpts devpts /dev/pts
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
# chmod 755 etc/init.d/rcS //给rcs可执行权限
# vim etc/inittab //新建inittab, 并设置内容如下
::sysinit:/etc/init.d/rcS
::respawn:-/bin/sh
::askfirst:-/bin/sh
::cttlaltdel:/bin/umount -a -r
# chmod 755 etc/inittab //给可执行权限
# cd dev
# mknod console c 5 1 //在dev目录下创建设备名字为console的字
符设备, 其主设备号为5, 次设备号为1(下同)
# mknod null c 1 3
# mknod tty1 c 4 1

```

制作根文件系统镜像

步骤

思路:

- 1.先制作一个空的镜像文件;
- 2.然后把此镜像文件格式化为ext3格式;
- 3.然后把此镜像文件挂载, 并把根文件系统复制到挂载目录;
- 4.卸载该镜像文件。
- 5.打成gzip包

相关命令

```

在源码最上层目录下执行以下命令
rm -rf rootfs.ext3 //清除该文件
rm -rf fs
dd if=/dev/zero of=./rootfs.ext3 bs=1M count=32 //dd命令用于读取、转换并输出数据
mkfs.ext3 rootfs.ext3 //将文件格式化成ext3文件系统
mkdir fs
mount -o loop rootfs.ext3 ./fs //挂在到文件夹fs
cp -rf ./_install/* ./fs //将所有的文件复制到fs
umount ./fs //将fs卸载
gzip --best -c rootfs.ext3 > rootfs.img.gz //将文件打包成rootfs.img.gz

```

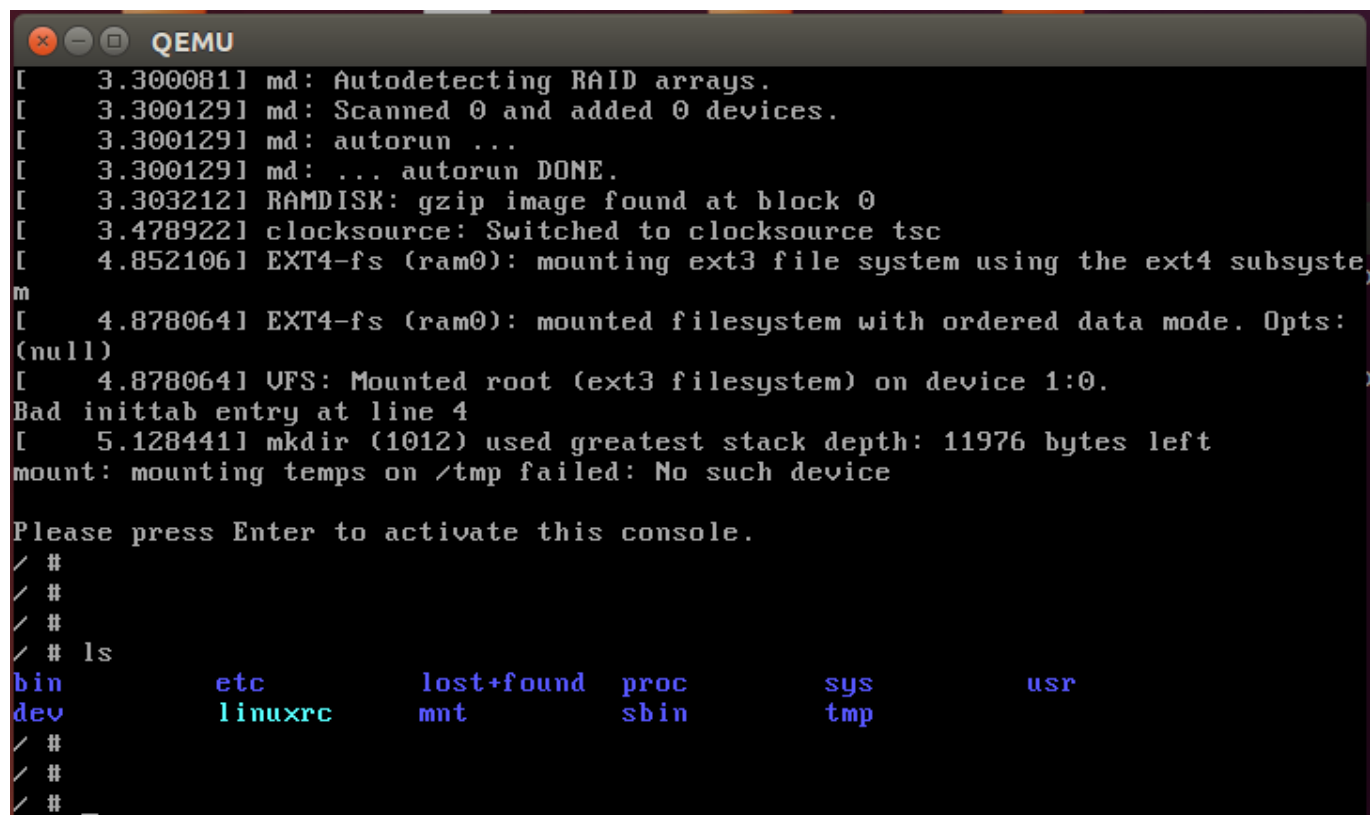
运行加了文件系统的内核镜像

相关命令

路径就是自己编译出来的内核镜像和文件系统镜像，后两个参数可以照抄

```
qemu-system-x86_64  
-kernel ./linux-4.9.229/arch/x86_64/boot/bzImage  
-initrd ./busybox-1.30.0/rootfs.img.gz  
-append "root=/dev/ram init=/linuxrc"  
-serial file:output.txt
```

运行效果



```
QEMU  
[ 3.300081] md: Autodetecting RAID arrays.  
[ 3.300129] md: Scanned 0 and added 0 devices.  
[ 3.300129] md: autorun ...  
[ 3.300129] md: ... autorun DONE.  
[ 3.303212] RAMDISK: gzip image found at block 0  
[ 3.478922] clocksource: Switched to clocksource tsc  
[ 4.852106] EXT4-fs (ram0): mounting ext3 file system using the ext4 subsystem  
[ 4.878064] EXT4-fs (ram0): mounted filesystem with ordered data mode. Opts: (null)  
[ 4.878064] VFS: Mounted root (ext3 filesystem) on device 1:0.  
Bad inittab entry at line 4  
[ 5.128441] mkdir (1012) used greatest stack depth: 11976 bytes left  
mount: mounting tmpfs on /tmp failed: No such device  
  
Please press Enter to activate this console.  
/#  
/#  
/#  
/# ls  
bin          etc          lost+found  proc        sys          usr  
dev          linuxrc      mnt        sbin        tmp  
/#  
/#  
/#
```