[如何制作根文件系统](http://blog.chinaunix.net/uid-22241162-id-2979543.html) 2011-10-25 08:50:51

分类： LINUX

<1>.概述  
 这一节，将制作一个最小的根文件系统，能让内核在该文件系统中运行。  
 制作文件系统的主要分为五个部分。  
 ① 编译Busybox 得到我们能使用的长用命令。  
 ② /dev/console   /dev/null 这两个设备节点  
 ③ /etc/inittab  初始化表格  
 ④ 配置文件指定的程序  
 ⑤ C库函数  
<2>.编译Busybox  
 得到Busybox的源码之后，解压缩可以看到有个一README文件，这个文件中说明了一些Busybox的一些信息，在INSTALL文件中说明了如何编译Busybox源码。  
 具体的步骤如下：  
 ① 修改编译工具和指定CPU的体系结构  
  做交叉开发的一个特性，也是很重要的一点就是要指定，我们编译好的代码在什么样的系统上取运行，根据不同的CPU体系会有不同的开发工具。这里  
 我使用的是ARM内核的CPU，所以我也会在Makefile中指定这一点。  
 具体操作：  
  进入源码目录，打开顶层的Makefile文件。  
  ARCH  ?= $(SUBARCH)  
  CROSS\_COMPILE ?=  
  修改为：  
  ARCH  ?= arm     //指定CPU内核体系结构  
  CROSS\_COMPILE ?= arm-linux-    //指定交叉编译工具  
 ② 配置与编译Busybox  
   在内核的源码目录下执行：make menuconfig。在弹出的配置对话框中，进行具体命令的配置选择。选择好需要的命令后，保存退出。  
   在内核的源码目录下执行：make 。这时候系统会编译Busybox源码。  
 【注】编译的时候如果遇到有错误的地方，可以根据提示进行修改，一般的情况下，是某个命令的配置上有错误，这个时候可以选择重新执行make menuconfig  
 进行重新配置，去掉出错的命令，重新编译即可。如果真的需要该命令，这个时候可以使用Google搜索工具，到网上搜一下。  
 ③ 安装Busybox  
   在此步骤中一点要注意，安装的时候不要直接执行make install命令，因为如果这样的话，系统会把编译好的Busybox命令直接拷贝到系统的目录下，这样  
  这样很可能会破坏掉我们使用的linux系统。我们应该把这些编译好的命令可执行文件直接放到某一个指定的目录中去。  
  具体操作：  
  先建立一个目录：mkdir /opt/first\_fs  
  然后执行如下命令：make CONFIG\_PREFIX=/opt/first\_fs install  
  这样系统就会把编译好的Busybox拷贝到:/opt/first\_fs目录中。  
<3>.创建/dev目录  
 /dev目录下存放的跟设备有关的文件，一些驱动程序的设备节点文件就在此目录下。在linux启动的过程中首先需要的两个设备节点是:/dev/console 控制台  
 /dev/null 方能的垃圾桶  
 首先我们浏览下所使用linux系统的这两个设备节点一些具体信息。执行ls - l /dev/console /dev/null  
 可以看到这两个设备都是字符设备：/dev/console的主设备号是：5 次设备号是：1  
                                    /dev/null 的主设备号是：1 此设备号是：3  
    根据这些信息就可以创建自己的设备节点了。  
 进入到刚才建立的/opt/first\_fs目录中  
 首先创建dev目录：  
 mkdir dev  
 cd dev  
 mknod console c 5 1  
 mknod null c 1 3  
   
 这两个设备节点已经创建完毕。  
<4>.创建配置文件。  
 创建etc目录  
 mkdir etc  
 vi etc/inittab  
 inittab内容：console::askfirst:-/bin/sh  
   
<5>.C库  
 拷贝编译工具下的动态库文件。具体操作：  
 执行arm-linux-gcc -v命令，参看交叉编译工具安装在什么目录下，然后拷贝该编译工具的arm-linux目录的lib库下的所有动态文件。  
 我的交叉编译工具安装在opt/tools目录下。  
 在自己做的根文件系统下，建立lib目录。  
 mkdir lib  
 cd /opt/tools/gcc-3.4.5-glibc-2.3.6/arm-linux/lib  
 ls -l  
 cp \*so\* /opt/first\_fs/lib -d  
 -d的作用，原来是一个链接文件的还拷贝成一个链接文件，如果不带-d的话会拷贝到源文件，这个时候文件就会比较大。  
到目前为止一个最小的根文件系统已经制作好了。  
为了能够烧写到开发板中使用，需要制作根文件系统文件映像。  
<1>.制作mkyaffs2image工具  
 为了提高效率，我使用的是为东山老师修改好的yaffs源码文件。  
 具体操作：  
 cd /home/hzscall/Develop  
 patch -p1  cd yaffs2  
 cd utils  
 make  
 ls -l  
 cp mkyaffs2image /usr/local/bin  
 chmod +x /usr/local/bin/mkyaffs2image  
<2>.制作根文件系统映像文件  
 cd /opt/first\_fs  
 mkyaffs2image first\_fs first\_fs.yaffs2  
 生成第一个first\_fs.yaffs2根文件系统。

完善根文件系统  
<1>.挂接proc目录  
 proc目录的挂接具体操作：  
 cd /opt/first\_fs  
 mkdir proc  
 vi /etc/inittab  
 在inittab文件中，增加以下内容：“::sysinit:/etc/init.d/rcS”  
 cd etc  
 mkdir init.d  
 vi rcS  
 在rcS文件中增加：mount -t proc none /proc  
 chmod +x rcS  
 /\*\*重新制作文件系统映像就可以使用ps命令显示系统信息\*\*/  
 另外一种方法  
    不适用mount -t proc none /proc命令  
 使用mount -a命令。  
 具体操作：  
 cd /opt/first\_fs  
 mkdir proc  
 vi /etc/inittab  
 增加的内容是：“::sysinit:/etc/init.d/rcS”  
 mkdir /etc/init.d  
 vi /etc/init.d/rcS  
 增加的内容是：mount -a   
 vi etc/fstab  
 fstab的格式如下：  
 #device  mount-point  type   options  dump fsck oder  
 proc     /proc        proc   defaults  0     0  
   
 /\*\*\*\*\*重新编译即可\*\*\*\*\*\*\*\*\*\*\*\*/  
<2>.自动创建设备节点  
 使用mdev机制自动生成dev目录下的设备节点。  
 在busybox的源码中搜索mdev会在docs目录下的mdev.txt文件，里面有具体的操作步骤。  
 一共分为六个步骤：  
 [1] mount -t sysfs sysfs /sys  
 [2] echo /bin/mdev > /proc/sys/kernel/hotplug  
 [3] mdev -s  
 [4] mount -t tmpfs mdev /dev  
 [5] mkdir /dev/pts  
 [6] mount -t devpts devpts /dev/pts

 具体操作：  
 cd /opt/first\_fs  
 mkdir sys  
 vi /etc/fstab  
 #device  mount-point  type   options  dump fsck oder  
 proc     /proc        proc   defaults  0     0  
 sysfs    /sys         sysfs  defaults  0     0  
 tmpfs    /dev         tmpfs  defaults  0     0  
   
 vi etc/init.d/rcS  
 增加的内容：  
 mkdir /dev/pts  
 mount -t devpts devpts /dev/pts  
 echo /bin/mdev > /proc/sys/kernel/hotplug  
 mdev -s  
制作根文件系统的工作完成，现在的跟文件系统是比较完善的了。