

第 7 章 嵌入式 Linux 的文件系统开发

内容简介:

本章介绍嵌入式 Linux 文件系统的开发步骤，如何配置各种文件系统的内容等等

1. 英贝德为你提供的文件系统开发资源

英贝德提供的可以对文件系统开发的所有资源包括：

- EBD9200 的 ext2 文件系统：ramdisk;
- minigui 的源代码（你也可以从网上下载）；
- 关键性说明文档。

这些内容可以让你轻松面对文件系统的操作。

2. 开发 Linux 文件系统的一般开发步骤

对于 Linux 文件系统的开发，有两种方法：一种是自己从头开始建立根文件系统，另外一种是从获取一个已经生成的文件系统，然后在此基础上添加和修改，最后形成自己的文件系统。

2. 1 在一个已经生成的文件系统为基础来创建自己的文件系统

对于 EBD 9200 的开发板上的文件系统，我们建议在已经具备的文件系统上进行开发，这样你将会省去很多的开发时间，在英贝德提供的文件系统上开发，按下面的步骤进行操作：（其中 ramdisk 是文件系统映象文件，英贝德提供 ramdisk 和 ramdiskM(配置了 minigui 的 USB 鼠标、键盘)两种。）

- 加上扩展名
ramdisk 改为 ramdisk.gz
- 解开压缩
\$ gunzip ramdisk.gz
- 映象文件挂装
\$ mount -o loop ramdisk /mnt/your_ramdisk_directory
- 对”/mnt/your_ramdisk_directory”目录进行操作，可随意增减文件：
\$ cd /mnt/your_ramdisk_directory
\$ do_whatever_you_want (create directories, files ...) //比如加入你的应用程序等
- 到你的映象文件目录下
\$ cd where_your_ramdisk_file_is
- 卸装文件系统
\$ umount /mnt/your_ramdisk_directory
- 压缩文件系统，生成最终的文件系统映象
\$ gzip -c -v9 ramdisk > /tftpboot/ramdisk

2. 2 自己建立根文件系统

当然，你也可以自己制作根文件系统，一般创造根文件系统可以有下面步骤：

- 创建一定大小的根文件系统
mke2fs -vm0 /dev/ram 4096

- 根文件系统挂装
mount -t ext2 /dev/ram /mnt
- 对文件系统进行操作
cd /mnt
cp /bin, /sbin, /etc, /dev ... files in mnt
cd ../
- 去除挂装
umount /mnt
- 文件系统生成
dd if=/dev/ram bs=1k count=4096 of=ext2ramdisk
- 得到文件系统映像
gzip -v9 ext2ramdisk

mke2fs 是用于在任何设备上创建 ext2 文件系统的实用程序 — 它创建超级块、索引节点以及索引节点表等。

在上面的用法中，**/dev/ram** 是上面构建有 4096 个块的 ext2 文件的设备。然后，将这个设备（**/dev/ram**）挂装在名为 **/mnt** 的临时目录上并且复制所有必需的文件。一旦复制完这些文件，就卸装这个文件系统并且设备（**/dev/ram**）的内容被转储到一个文件（**ext2ramdisk**）中，它就是所需的 **Ramdisk**（Ext2 文件系统）。

上面的顺序创建了一个 4 MB 的 **Ramdisk**，并用必需的文件实用程序来填充它。

一些要包含在 **Ramdisk** 中的重要目录是：

/bin — 保存大多数象 **init**、**shell**、文件管理实用程序等二进制文件。

/dev — 包含用在设备中的所有设备节点。

/etc — 包含系统的所有配置文件。

/lib — 包含所有必需的库，如 **libc**、**libdl** 等。

2.3 改变 ramdisk 的大小

要想使用 **ramdisk** 你必须得到内核的支持或是以模块的形式将他加载到系统中。其中内核的配置选项是 **CONFIG_BLK_DEV_RAM**。把 **ramdisk** 编译成一个可加载的模块的好处是你可以加载时重新确定 **ramdisk** 的大小。

第一个办法。在 **lilo.conf** 文件中加入这一行：

```
ramdisk_size=20000 (or ramdisk=20000 for old kernels)
```

这样在你使用 **lilo** 命令和重新启动计算机之后，**ramdisk** 的默认大小将会是 20M。这是一个 **/etc/lilo.conf** 文件的例子：

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz
    label=linux
    root=/dev/hda2
    read-only
    ramdisk_size=20000
```

确切的说,我目前使用了 19M 多的空间,文件系统也将占用一定空间。

当你以模块的形式编译 `ramdisk` 时,你可以在加载的时候决定 `ramdisk` 的大小。这也可以通过修改 `/etc/conf.modules` 的选项设置来得到。

```
options rd rd_size=20000
```

或是在命令行中指定参数给 `ismod`:

```
insmod rd rd_size=20000
```

以下是介绍如何使用这样的模块的例子:

卸载 `ramdisk`:

```
umount /tmp/ramdisk0.
```

卸载模块(再上一节所提到的过程中自动加载): **`rmmod rd`**

加载 `ramdisk` 模块并且把它的大小设为 20M: **`insmod rd rd_size=20000`**

创建一个文件系统: `mke2fs /dev/ram0`

加载 `ramdisk`: `mount /dev/ram0 /tmp/ramdisk0`

3. 如何开发 EBD9200 的文件系统

英贝德提供给你 EBD9200 的开发板时,也为你提供了已经开发好的一个文件系统(展开时大小大约为 19M 多),供你在建立自己的文件系统时使用。

3.1 创建大容量的文件系统

上节我们谈了创建文件系统的一般步骤,但是上面所创造的文件系统只有 4M,英贝德为你提供创建大容量文件系统以及在成熟文件系统中建立自己文件系统的方法。

需要准备的工作

比如英贝德为你已经提供的文件系统 `ramdisk`,我们把它挂装到 `/mnt/test1` 上,而我们自己创建的文件系统,我们挂装到 `/mnt/test2` 上,文件设备名假使为 `/tmp/loop_tmp`。

操作步骤

STEP 1: 把 `ramdisk` 进行挂装

```
Mount -o loop .../ramdisk /mnt/test1
```

STEP 2: 建立 `loop` 设备的临时挂接点和大小为 `size` 的文件,并清零

```
Mkdir /mnt/loop 2>/dev/null
```

```
Dd if=/dev/zero of=/mnt/loop_tmp bs=1k count=size >/dev/null
```

STEP 3: 将 `loop` 设备和临时文件联系

```
Losetup /dev/loop0 /mnt/loop_tmp
```

STEP 4: 建立大容量文件系统并配置自身

```
Mke2fs -m 0 /dev/loop0 2>/dev/null
```

STEP 5: 虚拟盘挂接

```
Mount /dev/loop0 /mnt/test2 -t ext2
```

STEP 6: 拷贝文件,改变初始化等文件,卸载挂接点

从 `/mnt/test1` 和其他文件中选择拷贝到 `/mnt/test2` 中

拷贝自己的应用程序到相应的目录下

对 `inittab` 等脚本文件进行修改

```
umount /mnt/test2
```

```
umount /mnt/test1
```

STEP 7: 得到最后的文件系统

```
Dd if=/dev/loop0 bs=1k count=size of =.../ext2ramdisk1
```

Gzip -v9 ../ext2ramdisk1

4. 对于嵌入式 Linux 的文件系统开发，你还需要什么？

至此，我们已经开发完毕文件系统，你可以进行内核和文件系统的下载，系统已经能够正常启动了，可能你这时已经信心十足，完全可以进行你的产品开发而专注于自己的顶层应用程序，这时，英贝德恭喜你，我们的任务已经基本完成，但是英贝德还会给你更多的技术支持，帮助你进入真正的开发领域。

这时候，你可能会着重考虑 linux 下的网络编程或者你自己的产品软件设计了，如果你对顶层程序的编写还有什么疑惑的话，请你继续阅读剩下的文档。