文件系统

§1 实验内容

- 制作 test.img，当用 mount 指令把 test.img 挂载在目录 t（任意目录）下时，find t 指令应显示：-aa, -bb, -ft, -fo, -fo/aa.
- 修改对应 romfs 的源代码来隐藏指定的一个文件（目录）
  - 首先加载模块 romfs，传入参数 hided_file_name = “aa”，再挂载做好的镜像 test.img 在指定的目录 t 下

    `insmod romfs hide_file_name = “aa”`

    `mount –o loop test.img t`
  - 用 ls 指令检测完成效果, terminal 里输入 ls t, ls t/fo 等，应显示找不到 aa 和 fo/aa
  - terminal 里输入 ls t/aa, ls fo/aa 等，应显示找不到 aa(请特别注意此条要求)
- 修改对应 romfs 的源代码来实现加密
  - `insmod romfs encry_file_name = “aa”`

    `mount –o loop test.img t`
  - 执行 cat t/aa 应显示不同于原文件的加密后的内容(如：原文为”aaaaa”，加密后为”*****”)
- 修改对应 romfs 的源代码来实现为特定文件加 execution 位
  - `insmod romfs addex_file_name = “aa”`

    `mount –o loop test.img t`
  - 修改代码之前，ls –l aa 对应显示’-rw-r-r—‘，代码修改并加载模块后，’ls –l aa’显示’-rwxr-xr-x’，同时，对应文件名变为绿色

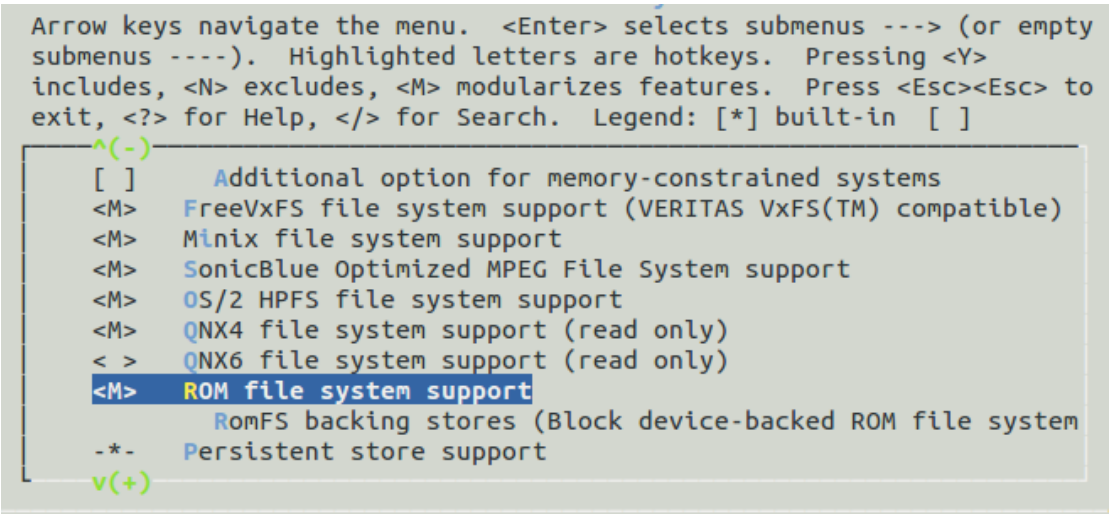
Tip: 同学们也可以尝试一次性传入三个参数加载模块。

§2 实验参考：romfs 文件系统简介

本实验要挂载的 img 是基于 romfs 文件系统的。romfs 文件系统是一个只读性质的小型文件系统。ROMFS 是一种简单的只读文件系统，主要是用来当做初始文件系统来使用的，在嵌入式 linux 或是 uclinux 中通常使用这中文件系统来作为引导系统的文件系统。linux 操作系统启动中一个是要加载内核，另一个就是要加载一个用于系统简单初始化的文件系统。这个文件系统的格式也是经过了很多发展的。现在一般使用的是一中 cpio 的格式。在嵌入式系统中一般使用 romfs＋其它的可读文件系统。romfs 由于它的小巧性（其内核编译只有 4000 字节），所以非常适合作为系统

启动初始化的文件系统。关于 ROMFS 最为权威的资料是内核源代码树下的“Documentation/filesystems/romfs.txt”，感兴趣同学可以参阅。

Ubuntu 中的 linux 系统以模块的形式支持 romfs，因为这是在 make menuconfig 的时候执行了将 romfs 以模块的形式进行编译：



这样方便修改 romfs 的源码后进行模块编译而不是整体内核编译。

在/usr/src 中的源码目录下找到 romfs 的源码文件夹，建议把它复制到自己习惯的工作目录下。这样一来，系统便会有两个 romfs 的模块可加载：一个是系统源码文件夹中自带的，另一个是自己修改的（注意要加载自己的 romfs 模块）。

更详细信息感兴趣同学可参考 http://romfs.sourceforge.net/ *What is romfs?*下的内容。

§ 3 实验参考：制作 romfs img 文件

通过 apt-get install genromfs，可以安装 romfs img 的制作工具 genromfs。按照§ 1 中的要求创建目录和对应文件，然后在该目录下通过 genromfs –f [path]将该目录制作成 img 文件，path 指定目标存放点。这是使用 genromfs 指令最简单的方法，其他可在 terminal 中用 man 指令查看。或者参见 http://linuxcommand.org/man_pages/genromfs8.html。

§ 4 实验参考：挂载文件系统

img 文件制作完毕后，并且已经加载了 romfs 模块，那么通过 mount -o loop XXX.img [directory]将自己的 XXX.img 挂载到 directory 目录下，此步骤可能需要使用 –t 来指定 img 所加载的文件系统类型。

挂载（mount）成功后，就可以在 directory 下访问用来制作 img 文件的文件夹了。

请同学通过 man 指令或者搜索引擎了解 mount/umount 详细用法。

§ 5 实验参考：romfs 文件系统源代码目录解析

super.c: 该模块的主入口文件，实现了模块的加载、卸载，和一些对 romfs 中文件的 inode 管理，文件读取等等方法。

storage.c: 进一步实现了一些方法，供 super.c 调用。

Makefile: 整个目录的编译文件。

目录下原始的 Makefile 只适用于内核编译的时候，因为仅仅指定了编译的.c 的文件和输出文件，没有指定编译使用的 build，所以需要修改。方法类似之前模块化编程中 Makefile 的写法。有遗忘的同学请及时复习/参考。

§ 6 实验参考：文件权限

文件的 rwx 访问权限被记录在文件所对应的 inode 的 i_mode 中，它是一个 short 值，以二进制的形式查看后 9 位，分别是 user 的 rwx 权限、group 的 rwx 权限和 other 的 rwx 权限，所以只要将 i_mode 和 1001001（二进制）即 0x49（十六进制）进行按位或运算，即可添加各自的 x 权限。

同学们需要修改 super.c 中对应语句实现 exeeceution 位的增加。

§ 7 参考文献

[1] ROMFS 文件系统分析: http://zhwen.org/?page_id=756\

[2] http://linuxcommand.org/man_pages/genromfs8.html