

Linux系统下以模块方式安装卸载文件系统Linux认证考试 PDF
转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/580/2021_2022_Linux_E7_B3_BB_E7_BB_c103_580560.htm 以Fedora8下面安装minix文件系统

为例：为了保证与系统内核相匹配，首先得获得相应版本的minix源代码，首先通过uname -r查询本机的内核版本：

```
[cocobear@cocobear ~]$ uname -r 2.6.24.4-64.fc8
```

 在Kernel.org主

页上可以获得2.6.24.4-64内核的源代码，其实我们只需要其中linux-2.6.24.4/fs/minix/目录中的代码。因为我们不需要对整个内核进行重新编译，因此我们只需要

在linux-2.6.24.4/fs/minix/目录下写一个Makefile，生成相应的minix.ko就可以了。在开始写Makefile之前要确认系统已经安装了以下的包：

```
[cocobear@cocobear ~]$ rpm -qa | grep kernel
```

```
kernel-devel-2.6.24.4-64.fc8 kernel-headers-2.6.24.4-64.fc8
```

```
kernel-2.6.24.4-64.fc8
```

 在模块编译的过程中需要用到。在源代码中已经有一个Makefile：

```
## Makefile for the Linux minix
```

```
filesystem routines. # obj-$(CONFIG_MINIX_FS) = minix.o
```

```
minix-objs := bitmap.o itree_v1.o itree_v2.o namei.o inode.o file.o
```

```
dir.o 修改该文件为：## Makefile for the Linux minix filesystem
```

```
routines. # make minix fs as kernel module obj-m = minix.o
```

```
minix-objs := bitmap.o itree_v1.o itree_v2.o namei.o inode.o file.o
```

```
dir.o KERNELDIR:=/lib/modules/$(shell uname -r)/build
```

```
PWD:=$(shell pwd) default: make -C $(KERNELDIR) M=$(PWD)
```

```
modules clean: rm -rf *.o *.mod.c *.ko *.symvers
```

 这里简单的解释

一下，obj-m表示该文件将以模块的方式编译；因为本模块由多个文件组成，采用模块名加objs（minix-objs）后缀的形式

来定义模块的组成文件。KERNELDIR定义了代码树的位置，PWD定义了当前文件夹位置；而make命令中-C选项指定了代码树的位置（由KERNELDIR给出），M=\$(PWD)指定了在当前目录进行构建工作。最后一行清理编译过程产生的文件。完成了Makefile后我们就可以开始编译这个文件系统模块了，直接输入make就开始编译了：

```
[cocobear@cocobear minix]$  
make make -C /lib/modules/2.6.24.4-64.fc8/build
```

```
M=/home/cocobear/minix modules make[1]: Entering directory  
`/usr/src/kernels/2.6.24.4-64.fc8-i686` CC [M]  
/home/cocobear/minix/bitmap.o CC [M]  
/home/cocobear/minix/itree_v1.o CC [M]  
/home/cocobear/minix/itree_v2.o CC [M]  
/home/cocobear/minix/namei.o CC [M]  
/home/cocobear/minix/inode.o CC [M]  
/home/cocobear/minix/file.o CC [M] /home/cocobear/minix/dir.o  
LD [M] /home/cocobear/minix/minix.o Building modules, stage 2.  
MODPOST 1 modules CC /home/cocobear/minix/minix.mod.o  
LD [M] /home/cocobear/minix/minix.ko make[1]: Leaving  
directory `/usr/src/kernels/2.6.24.4-64.fc8-i686` 编译结束后会面
```

当前目录下生成minix.ko文件，这就是我们需要的东西，使用insmod命令就可以安装这个minix文件系统模块了。当然这里需要有root权限。我们来演示一下minix模块的加载：

```
[cocobear@cocobear minix]$ cat /proc/modules | grep minix  
[cocobear@cocobear minix]$ 这里可以看到minix并没有被加载  
，我们使用insmod minix.ko命令： [cocobear@cocobear minix]$  
sudo insmod minix.ko [cocobear@cocobear minix]$ cat
```

```
/proc/modules | grep minix minix 28676 0 - Live 0xd0e7d000
```

insmod后我们从上面的信息可以看到minix模块已经被加载，如果不需要使用这个模块我们同样可以很方便的把它卸载：

```
[cocobear@cocobear minix]$ sudo rmmod minix.ko
```

```
[cocobear@cocobear minix]$ cat /proc/modules | grep minix
```

```
[cocobear@cocobear minix]$
```

到此我们顺利的完成了文件系统的编译、安装以及卸载。 BTW：中间遇到了点问题写

了Makefile后输入make提示：“make: Nothing to be done for `default’.”，在网上找到了原因，在make命令前要使用tab，而不是空格，而我的刚好的空格，郁闷，以前就似乎遇到过的。更多优质资料尽在百考试题论坛 百考试题在线题库 linux 认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com