静态编译

在drivers/char添加hello.c

#include <linux/init.h>

#include <linux/module.h>

static int hello\_init(void)

{

printk(KERN\_ALERT"Hello,World\n");

return 0;

}

static void hello\_exit(void)

{

printk(KERN\_ALERT"Googbye,World\n");

return 0;

}

module\_init(hello\_init)

module\_exit(hello\_exit)

MODULE\_LICENSE("DualBSD/GPL");

MODULE\_AUTHOR("yuming");

MODULE\_DESCRIPTION("my module");

MODULE\_ALIAS("ming");

在drivers/char中的Kconfig文件中添加

config HELLO\_DRIVER

bool "my hello"

default y

在drivers/char中的Makefile文件中添加

obj-$(CONFIG\_HELLO\_DRIVER) += hello.o

重新编译内核

注意：在make menuconfig 命令后找到你配置的模块

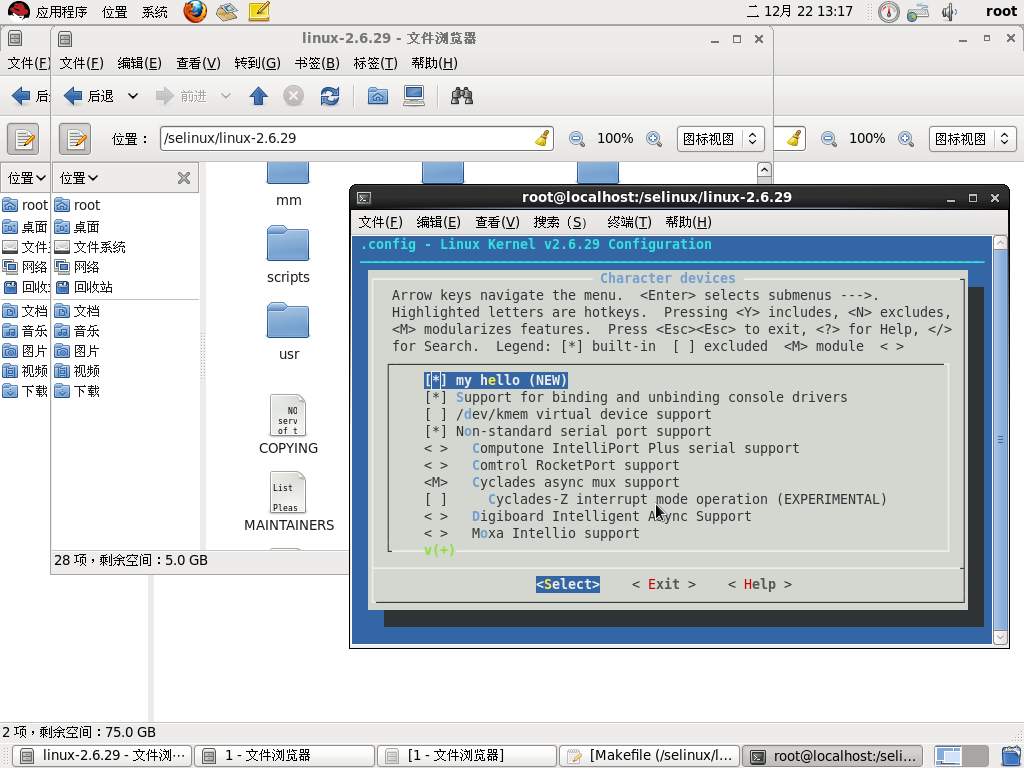
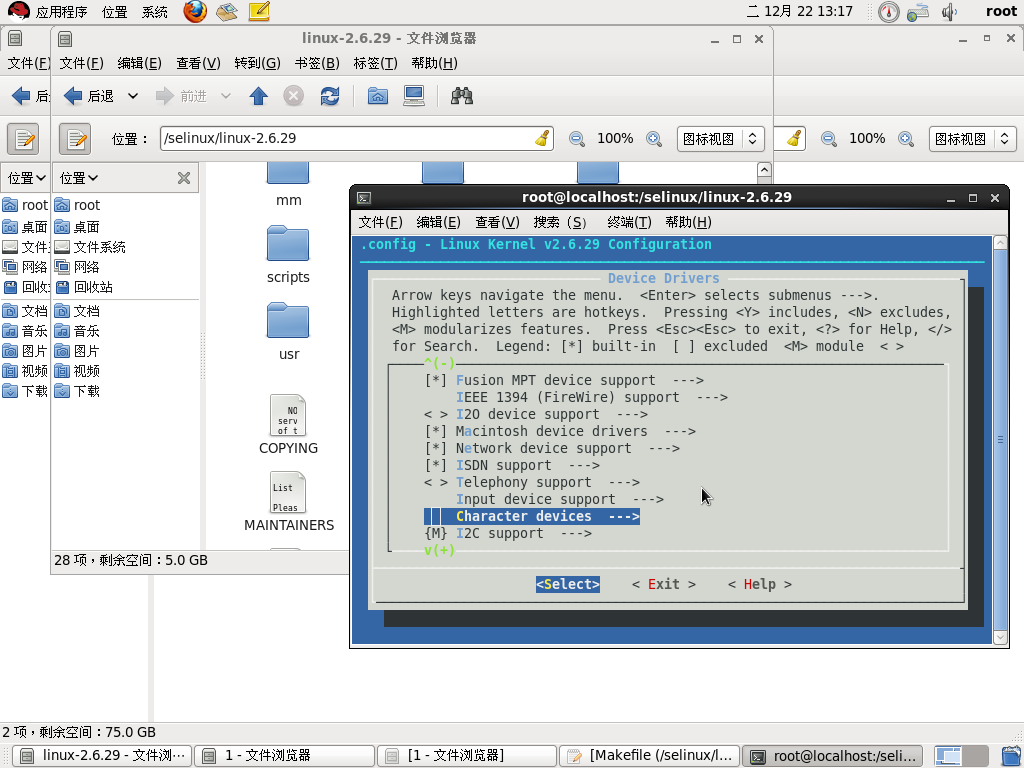
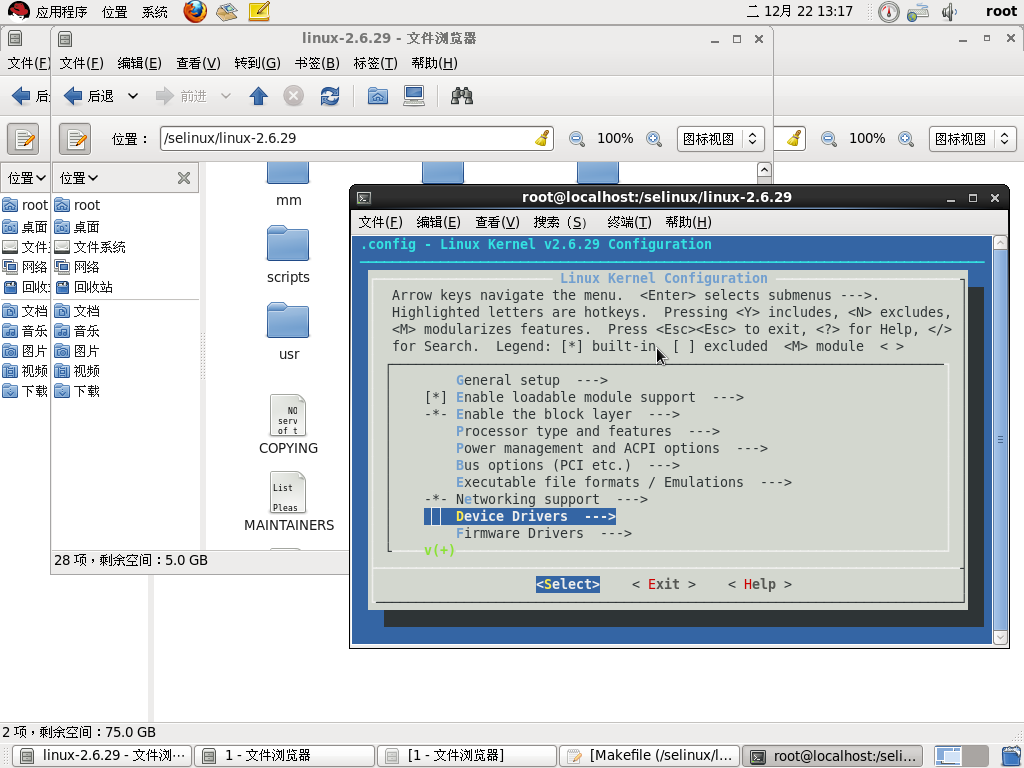
实验一得步骤：

先解压下载好的内核文件

   xz -d linux-2.6.32.71.tar.xz

     tar xvf  linux-2.6.32.71.tar

make menuconfig会生成 .config的配置文件



1、将/boot/config-2.6.18-53.e15 拷贝到Linux2.6.29内核目录下

2、make menuconfig 退出

3、make bzImage(20分钟）

4、make modules编译模块（编译选择为M的选项）

5、安装模块

make modules\_install

1. 制作文件系统

mkinitrd initrd-2.6.29 2.6.29

(/lib/modules/下必须有内核版本）会产生initrd-2.6.29

1. 内核安装过程

cp /arch/x86/boot/bzImage /boot/vmlinuz-2.6.29

cp /linux2.6.29/initrd-2.6.29 /boot/

cp是拷贝命令，你也可以手动操作拷贝

8、修改kconfig文件（控制grub的选项）

vim /etc/grub.conf

title my-linux

kernel /vmlinuz-2.6.29 ro \*\*\*

initrd /initrd-2.6.29

进入编译好的系统，输入dmesg查看是否在日志输出是否输出Hello,World