LAB3 实验报告

小组成员及贡献排序:

林元灿 汪值 吕若皙 陈景昊 木扎拜·塔依尔

一、编译内核

（1）、查看当前系统号

```
ycnalin@ubuntu:/usr/src/linux$ uname -a
Linux ubuntu 4.10.0-37-generic #41~16.04.1-Ubuntu SMP Fri Oct 6 22:42:59 UTC 2017
 x86_64 x86_64 x86_64 GNU/Linux
```

（2）（3）、编译安装 linux 内核

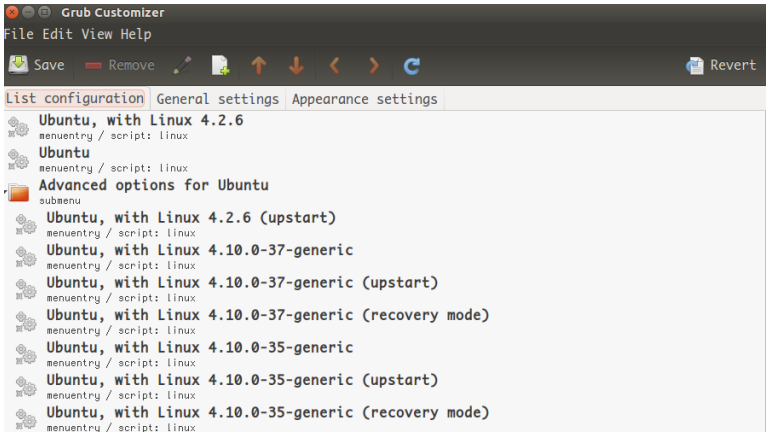
```
su root
tar -xvf linux-4.2.6.tar.xz -C /usr/src/
ln -sv linux-4.2.6 linux
# prerequisites: sudo apt-get install libncurses5-dev
make menuconfig        # config kernel
# clean last compile: make mrproper,make clean
make -j4
make install

```

（4）替换 linux 内核

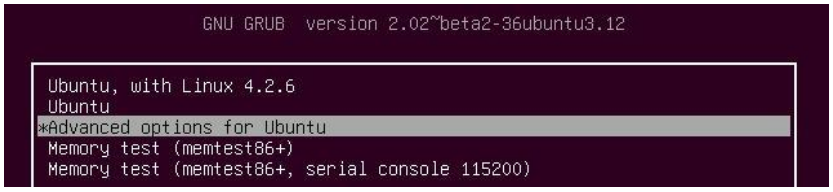
设置 linux 启动配置文件中 **timeout** 值

```
ycnalin@ubuntu:~$ cat -n /boot/grub/grub.cfg | grep timeout
    86    set timeout=30
    88    if [ x$feature_timeout_style = xy ] ; then
    89      set timeout_style=hidden
    90      set timeout=5
    91    # Fallback hidden-timeout code in case the timeout_style feature is
    94      set timeout=5
```

使用 Grub customizer 自动设置启动文件



重启系统按 **Shift** 按键选择内核



查看内核是否更改成功——成功切换为 4.2.6 内核

```
ycnalin@ubuntu:~$ uname -a
Linux ubuntu 4.2.6 #1 SMP Sun Nov 19 16:02:48 CST 2017 x86_64 x86_64 x86_64 GNU/
Linux
```

二、添加系统调用

(1)、添加系统调用声明 位置/include/linux/syscalls.h

```
asmlinkage long sys_hellosys(const char __user *str,unsigned int len);
```

(2)、添加系统调用函数 位置/kernel/sys.c

```
asmlinkage long sys_hellosys(const char __user *str,unsigned int len){
    struct file *filp = NULL,*filp2 = NULL;
    int err = 0;
    loff_t offset = 0;
    unsigned int ret;
    printk("enter_my_syscall");

    filp = filp_open("/tmp/hellosys", O_CREAT | O_RDWR, 0644);
    filp2 = sys_open("/tmp/hellosys2", O_CREAT | O_RDWR, 0644);

    if(filp){
        ret = vfs_write(filp,str,len,&offset);
        filp_close(filp, NULL);
        printk("write done hellosys");
    }
    if(filp2){
        ret = vfs_write(filp2,str,len,&offset);
        filp_close(filp2, NULL);
        printk("write done hellosys");
    }
    if(ret==0) printk("open_file_error");


    return ret;
}
```

(3)、修改系统调用向量表 /arch/x86/entry/syscalls/syscall_64.tbl

```
332  333 64  hellosys    sys_hellosys
```

(4)、重新编译系统，方法同一

(5)、在新内核中测试系统调用，代码如下：

```
#include <sys/syscall.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h> /* for strerror */
#include <errno.h>
#include <stdio.h>

int main(){
    const char test[] = "test syscall hellosys!\n";
    long ret = syscall(333, test , sizeof(test));
    printf("result is %ld\n", ret);
    printf("errno num is %d,%s\n", errno, strerror(errno));
    return 0;
}
```

编译执行效果图，注：代码调用系统调用，在/tmp/下新建文件 hellosys，并向其中打印用户字符串。

```
root@ubuntu: /home/ycnalin/Desktop
root@ubuntu:/home/ycnalin/Desktop# rm /tmp/hellosys
root@ubuntu:/home/ycnalin/Desktop# ./demo
Killed
root@ubuntu:/home/ycnalin/Desktop# cat /tmp/hellosys
test syscall hellosys!
root@ubuntu:/home/ycnalin/Desktop#
```