



openEuler 编译内核实验

实验报告

姓 名	邓语苏
学 号	22920212204066
日 期	2024 年 5 月 8 日
学 院	信息学院
课程名称	操作系统

实验报告

目录

1	Tips	1
2	安装编译	1
3	任务一	1
4	任务二	2
5	实验心得	2

1 Tips

1. 在步骤 3 中新建的 output 文件夹是空的很正常
2. 使用 reboot 重启系统
3. 重启系统后要用硬件连接树莓派 (可以选择用网线 or 显示器 + 键盘)
4. 在步骤 4 的网络配置中, 命令改为

代码 1 网络配置

```
insmod ~/kernel/net/rfkill/rfkill.ko
insmod ~/kernel/drivers/net/wireless/broadcom/brcm80211/brcmutil/brcmutil.ko
insmod ~/kernel/net/wireless/cfg80211.ko
insmod ~/kernel/drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.ko
```

5. 同理在 rc.local 中添加代码也需更改

2 安装编译

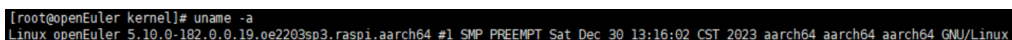


图 1: 旧内核安装成功截图

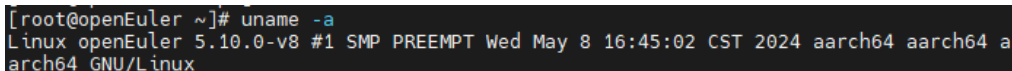


图 2: 新内核安装成功截图

3 任务一

查找相关资料, 解释 hello_world.c 文件中以下代码的含义和作用:

1. MODULE_LICENSE: 在 Linux 内核文档有如下表述:

The sole purpose of this tag is to provide sufficient information whether the module is free software or proprietary for the kernel module loader and for user space tools.

即这个 tag 提供了以便内核模块加载器和用户空间工具判断模块是自由软件还是专有软件的信息。

2. module_param: 是用于定义模块参数的宏, 三个参数分别为变量名、类型、权限。
3. MODULE_PARM_DESC: 用于定义模块参数的描述的宏。描述会在使用 modinfo 命令时显示出来
4. module_init: 用于通知内核初始化模块的时候, 要使用哪个函数进行初始化。
5. module_exit: 用于告诉内核, 当卸载模块时, 需要调用哪个函数。
6. __init: 用于将函数放在可执行文件的 init 段中, 将在初始化时被执行。
7. __exit: 将函数放在 exit 段中, 在模块卸载阶段执行。

内容出处:

1. <https://tldp.org/LDP/lkmpg/2.6/html/x323.html>
2. <https://www.kernel.org/doc/html/latest/process/license-rules.html#id1>

4 任务二

首先, 在文件开始处的三个参数定义进行修改, 写入自己的姓名学号年龄:

代码 2 声明并初始化变量

```
static char* name = "dys";
module_param(name, charp, 0644);
MODULE_PARM_DESC(name, "char* param\n");

static char* id = "4066";
module_param(id, charp, 0644);
MODULE_PARM_DESC(id, "char* param\n");

static int age = 20;
module_param(age, int, 0644);
MODULE_PARM_DESC(age, "int param\n");
```

然后依题意实现两个函数即可。前者直接调用 `printk` 输出, 该函数的使用方式与 `printf` 一致。后者遍历学号、转化为整数并按规则计算:

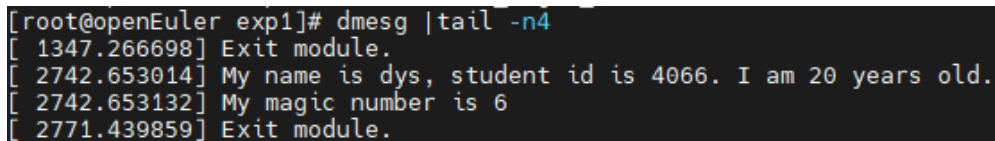
代码 3 `hello_student` 函数

```
void hello_student(char* name, char* id, int age) {
    printk(KERN_ALERT "My name is %s, student id is %s. I am %d years old.\n", name, id, age);
}
```

代码 4 `my_magic_number` 函数

```
void my_magic_number(char* id, int age){
    int i;
    int magic_number=0;
    for (i=0; i<4; i++){
        magic_number+=id[i]-'0';
    }
    magic_number+=age;
    magic_number=magic_number%10;
    printk(KERN_ALERT "My magic number is %d\n", magic_number);
}
```

然后, 使用 `insmod` 安装内核模块, 然后等待一段时间, 使用 `dmesg` 查看输出, 结果如下图



```
[root@openEuler exp1]# dmesg |tail -n4
[ 1347.266698] Exit module.
[ 2742.653014] My name is dys, student id is 4066. I am 20 years old.
[ 2742.653132] My magic number is 6
[ 2771.439859] Exit module.
```

图 3: 结果

5 实验心得

在本次实验中, 我编译并运行了 Linux 内核。在自行编译运行的基础上, 我实现了一个简单的内核模块。了解了 Linux 内核模块的编程并进行了实际体验。在实验中, 我对 Linux 内核编程的细节有了更多把握, 并且注意了 Makefile 中需要使用 Tab 而非空格, 否则会出现错误。