**广州大学学生实验报告**

**开课学院及实验室：**计算机科学与教育软件学院计算机软件实验室 **2020年 9月 23日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与教育软件学院** | **年级/专业/班** | **计科182** | **姓名** | 莫广贤 | **学号** | 1806100104 |
| **实验课程名称** | Unix/linux操作系统分析实验 | | | | | **成绩** |  |
| **实验项目名称** | 文件操作算法 | | | | | **指导老师** | 陶文正 |

**实验三 文件操作算法**

1. **实验目的**

1. 理解Linux中虚拟文件系统的内容

2. 学习编写内核模块的方法

3. 在虚拟文件系统/proc中实现文件操作算法

1. **实验环境**

Ubuntu20.04,源码内核版本：Linux-4.2.6

1. **实验内容**

编写一个内核模块，在/proc文件系统中增加一个目录hello，并在这个目录中增加一个文件world，文件的内容为hello world。

**四、 实验原理 实验中用到的系统调用函数（包括实验原理中介绍的和自己采用的），实验步骤，**

（一）/proc 文件系统简介

/proc 文件系统是Linux上的一种虚拟文件系统，存储的是当前内核运行状态的一系列特殊文件，用户可以通过这些文件查看有关系统硬件及当前正在运行进程的信息，甚至可以更改其中某些文件来改变内核的运行状态。最初开发 /proc 文件系统是为了提供有关系统中进程的信息。但是由于这个文件系统非常有用，因此内核中的很多元素也开始使用它来报告信息或启用动态运行时配置。图 1 是对 /proc 中部分元素进行一次交互查询的结果。它显示的是 /proc 文件系统的根目录中的内容。注意，在左边是一系列数字编号的文件。每个实际上都是一个目录，表示系统中的一个进程。由于在 GNU/Linux 中创建的第一个进程是 init 进程，因此它的 process-id 为 1。然后对这个目录执行一个 ls 命令，这会显示很多文件。每个文件都提供了有关这个特殊进程的详细信息。/proc 中另外一些有趣的文件有：cpuinfo，它标识了处理器的类型和速度；pci，显示在 PCI 总线上找到的设备；modules，标识了当前加载到内核中的模块。

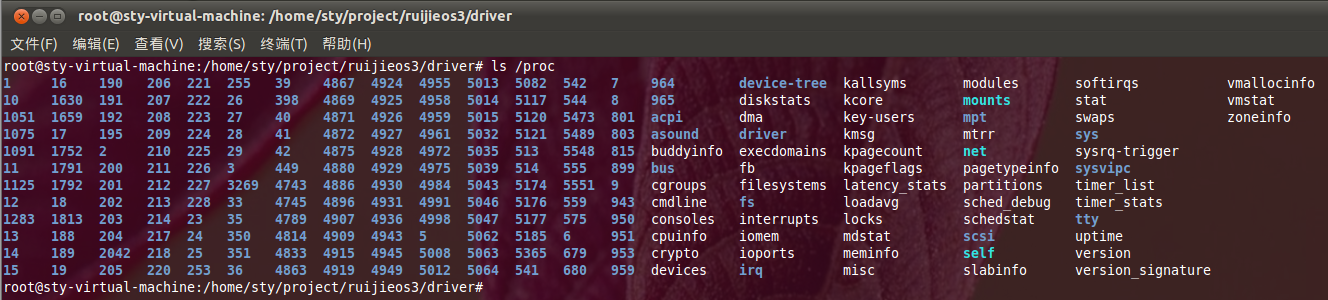
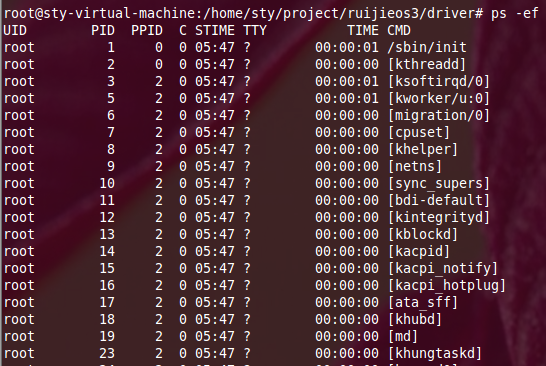
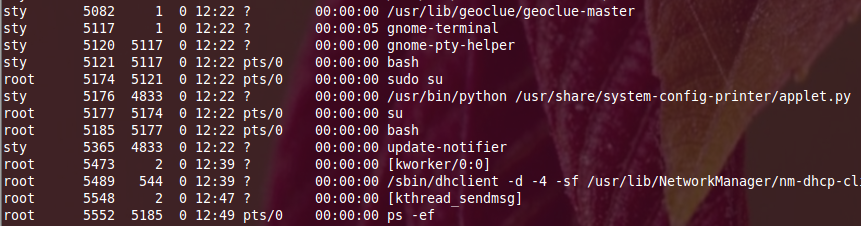


图1





这些文件的解释和意义如下：

cmdline：系统启动时输入给内核命令行参数

cpuinfo：CPU的硬件信息 (型号, 家族, 缓存大小等)

devices：主设备号及设备组的列表，当前加载的各种设备（块设备/字符设备）

dma：使用的DMA通道

filesystems：当前内核支持的文件系统，当没有给 mount(1) 指明哪个文件系统的时候， mount(1) 就依靠该文件遍历不同的文件系统

interrupts ：中断的使用及触发次数，调试中断时很有用

ioports I/O：当前在用的已注册 I/O 端口范围

kcore：该伪文件以 core 文件格式给出了系统的物理内存映象(比较有用)，可以用 GDB 查探当前内核的任意数据结构。该文件的总长度是物理内存 (RAM) 的大小再加上 4KB

kmsg：可以用该文件取代系统调用 syslog(2) 来记录内核日志信息，对应dmesg命令

kallsym：内核符号表，该文件保存了内核输出的符号定义, modules(X)使用该文件动态地连接和捆绑可装载的模块

loadavg：负载均衡，平均负载数给出了在过去的 1、 5,、15 分钟里在运行队列里的任务数、总作业数以及正在运行的作业总数。

locks：内核锁 。

meminfo物理内存、交换空间等的信息，系统内存占用情况，对应df命令。

misc：杂项 。

modules：已经加载的模块列表，对应lsmod命令 。

mounts：已加载的文件系统的列表，对应mount命令，无参数。

partitions：系统识别的分区表 。

slabinfo：sla池信息。

stat：全面统计状态表，CPU内存的利用率等都是从这里提取数据。对应ps命令。

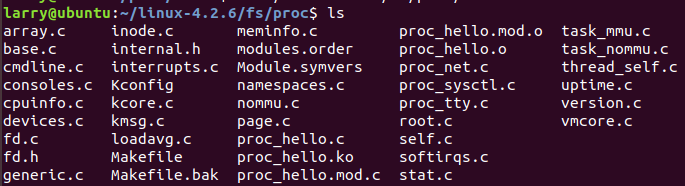
swaps：对换空间的利用情况。

version：指明了当前正在运行的内核版本。

**五、实验中用到的系统调用函数（包括实验原理中介绍的和自己采用的）**

**六、实验步骤**

实现proc系统的代码主要在linux/fs/proc目录下。该目录下的文件如下图所示。



1、创建目录

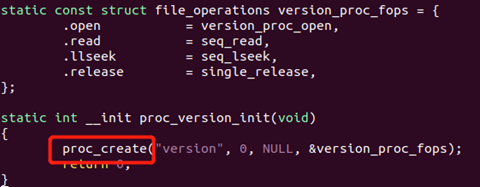
在Linux内核网址https://www.kernel.org/doc/html/latest/查阅相关文档可知，proc中创建目录的函数为proc\_mkdir，该函数原型为：

struct proc\_dir\_entry \*proc\_mkdir(const char \*name, struct proc\_dir\_entry \*parent)；

该函数接受两个参数，分别为要创建的目录的名称以及一个指向该目录的父目录的指针，并返回指向当前目录的指针结构体。

2、创建文件

考虑到在/proc目录下有version和softirqs等文件，且源码中有以上文件的实现，分别为version.c和softirqs.c，则可参考这两个文件的实现进一步实现创建world文件。以version.c为例，部分代码如下图所示。

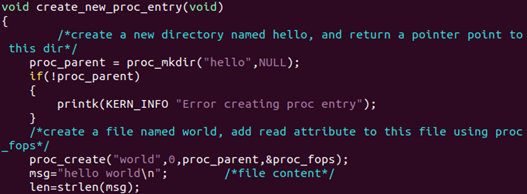


显然，创建version文件的函数为proc\_create，查阅文档可知，该函数原型为：

static inline struct proc\_dir\_entry \*proc\_create(const char \*name, umode\_t mode, struct proc\_dir\_entry \*parent, const struct file\_operations \*proc\_fops)；

该函数接收4个参数，第一个参数为文件名，第二个参数为文件的读写权限，第三个参数为其父目录的结构体指针，第四个参数为文件的读写操作结构体。

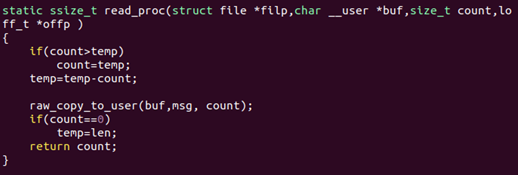
参照/proc/version文件的实现代码和本实验要求，可写出创建world文件的主要代码：



其中，world文件的父目录指针即为先前创建hello目录所返回的指针，proc\_fops定义了该文件所能执行的操作，由于本实验只要求读取文件内容，因此只实现了read属性：



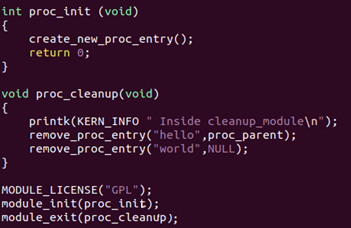
其中，read\_proc为实现读取文件内容的函数指针，其实现如下：



该函数将文件内容通过msg复制给buf，以此实现文件内容的读取。

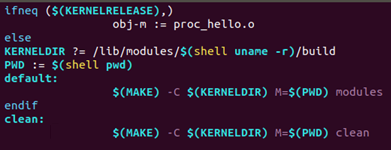
1. 定义模块的初始化和清理函数

其实现如下：



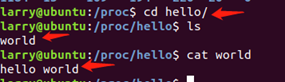
5、编写Makefile

为编译该模块，还需编写Makefile文件，该文件内容如图所示：



6、实验结果

结果如图所示。由图可见，/proc目录下生成了hello目录，且该目录下的world文件的内容为hello world。



7、运行与验证

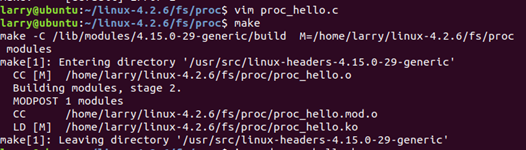
尽管本实验的内核版本为Linux-4.2.6，但是经测试可知，该模块在3.10及以上版本均可使用。

为使用该模块，首先需要下载3.10及以上Linux内核版本并解压，然后将proc\_hello.c文件拷贝至Linux/fs/proc目录下，由于该目录下原本就有Makefile文件，因此要对原Makefile文件备份后再拷贝本实验中的Makefile：

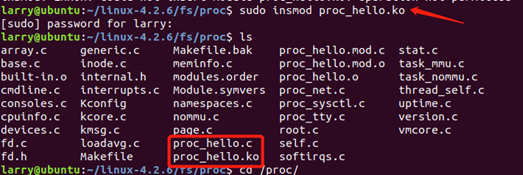
mv Makefile Makefile.bak

mv my\_makefile Makefile

然后，再使用make命令编译模块：

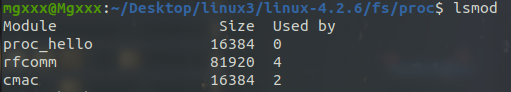


编译成功后，再使用insmod命令安装模块，可以看到该目录下多了一个ko文件，该文件即为编译成功的模块文件。

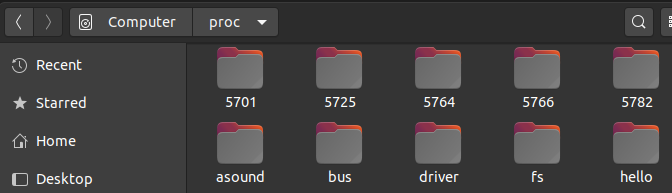
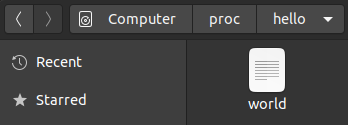


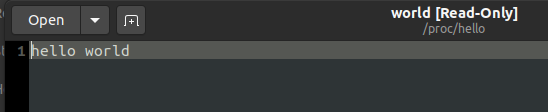
最后切换至/proc目录，即可看到上述结果。

**七、实验结果分析（截屏的实验结果，与实验结果对应的实验分析）**



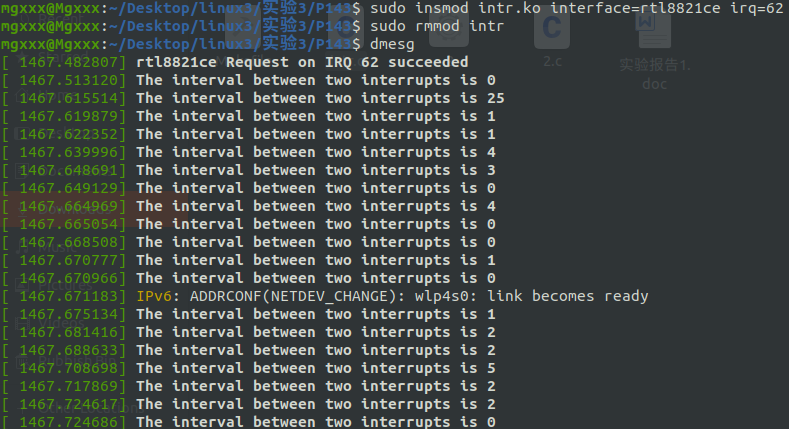
插入编写好的proc\_hello模块后，计算机的proc目录下会出现以下hello目录和world文件，world文件的内容为”hello world”：

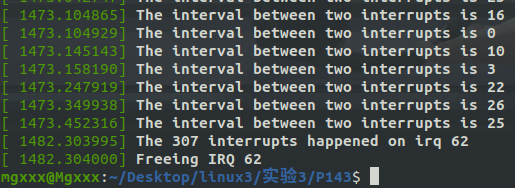
 



实验内容为编写内核模块在/proc增加一个hello目录，并在hello目录下增加一个文件world，内容为“hello world”。在proc\_hello.c中，使用proc\_mkdir来创建hello目录，接着使用proc\_create来创建了world文件，并设置了只能对该文件读。由于是通过proc\_mkdir和proc\_create来创建的目录和文件，所以，需要使用remove\_proc\_entry来删除这些创建的目录和文件。

扩展实验一





上面的两幅图是插入编写的模块后，打印的日志和卸载模块后打印的日志。日志中多次打印了两次中断间隔，可知该设备是有多次中断的。在最后关闭了无线网络设备之后，便不再打印了。因为该设备没有使用，所以就停止了对它的中断，最后卸载了该模块，打印了”freeing irq 62”。（该设备对应的IRQ是62）

**八、实验总结**

通过本次实验，理解了linux下虚拟文件系统的内容，清晰了/proc下的每个文件的意义和功能。又一次编写了内核模块，加深了对内核模块书写的流程。在编写内核模块的时候使用到了对/proc下操作文件的函数，明白了怎样对该目录下文件的进行创建和删除。简单应用了中断来计算网卡设备两次中断之间的间隔，学习到了中断的用处。

**九、实验数据及源代码（学生必须提交自己设计的程序源代码，并有注释，源代码电子版也一并提交），包括思考题的程序。**

源码数据见同目录下的文件夹” 1806100104-莫广贤-实验1-计科182”。