

CS353 Project 1

实验要求

1. 编写模块实现以下要求

- 接受三个参数：`operand1` 类型为 `int`，`operand2` 类型为 `int` 数组，`operator` 类型为 `charp`（字符串）。
- 创建 `proc` 文件 `/proc/<你的学号>/calc`。
- 如果 `operator` 为 `add`，那么 `operand2` 的每一个元素都加上 `operand1`，得到结果数组；如果 `operator` 为 `mul`，那么 `operand2` 的每一个元素都乘上 `operand1`，得到结果数组。
- 当读取 `proc` 文件时，输出结果数组，每个元素用逗号分隔。
- 当用户向 `proc` 文件写入一个数字时，这个数字作为新的 `operand1` 重新进行计算。
- 一个例子如下：

```
# sudo insmod calc.ko operand1=2 operand2=1,2,3,4,5 operator=add
# cat /proc/<ID>/calc
3,4,5,6,7
# echo 3 > /proc/<ID>/calc
# cat /proc/<ID>/calc
4,5,6,7,8
# sudo rmmod calc
# sudo insmod calc.ko operand1=2 operand2=1,2,3,4,5 operator=mul
# cat /proc/<ID>/calc
2,4,6,8,10
# echo 3 > /proc/<ID>/calc
# cat /proc/<ID>/calc
3,6,9,12,15
# sudo rmmod calc
```

2. 编写一个程序实现以下要求：

- 从 `/proc` 文件系统中得到系统中的所有进程 `PID` 以及相关信息。
- 输出这些进程的 `PID`，进程状态，进程的命令行参数三列信息。
- `PID` 5字符宽度，右对齐，空格填补空缺；每列信息之间用一个空格分隔。
- 输出效果可以参考命令 `ps -e -ww -o pid:5,state,cmd` 的输出效果。

实验提示

1. 实现模块的读写函数时，若要读取用户缓存内容，要先使用 `copy_from_user` 函数将用户缓存复制到内核空间中；若要写入用户缓存，则使用 `copy_to_user` 将内核空间内容复制过去。这两个函数定义在头文件 `linux/uaccess.h` 中。
2. 模块参数传递用到的宏定义在头文件 `linux/moduleparam.h` 中，注意普通参数和数组参数使用的宏是不同的，可以阅读头文件中的注释来进一步了解这些宏。
3. 虽然在内核模块编程中无法使用 C 标准库，Linux 内核自己实现了大部分标准库中的函数。对本次实验有用的函数大多定义在头文件 `linux/kernel.h` 和 `linux/kstrtox.h` 中。若想使用某个标准库函数，不妨先在网络搜索 Linux 内核是否自带这个函数。

4. 模块退出的时候不要忘了把创建的 proc 文件和文件夹都删除掉。
5. 实现简易 ps 程序所需要的 proc 文件有 `/proc/<PID>/cmdline` 和 `/proc/PID/stat`。对于部分进程，其 `cmdline` 文件为空，此时可输出 `/proc/<PID>/comm` 文件中的内容。
6. 可以从模版代码出发。
7. 可以参考 <https://sysprog21.github.io/lkmpg/#the-proc-file-system> 进行模块编写，参考 <https://man7.org/linux/man-pages/man5/proc.5.html> 了解 proc 中各个文件的作用

实验提交

提交渠道：Canvas

提交文件：学号_project1.zip，源码文件夹 学号_project1_src（所有源代码文件以及Makefile），实验报告学号_project1_report.pdf。

实验报告内容包括但不限于实验过程、实验效果截图、实验心得（实验过程中遇到的困难、解决的方法，或者是值得分享的小技巧）。