

# CS3302 Project1

花振东 jackeyhua@sjtu.edu.cn

2025 年 3 月 10 日

## 目录

<b>1 实现内核模块</b>	<b>2</b>
1.1 参数传递 . . . . .	2
1.2 内核模块的初始化 . . . . .	2
1.3 内核模块的清理 . . . . .	2
1.4 内核模块的读 . . . . .	2
1.5 内核模块的写 . . . . .	3
1.6 实验截图 . . . . .	3
1.7 实验心得 . . . . .	3
<b>2 实验用户态函数</b>	<b>4</b>
2.1 解析进程状态 . . . . .	4
2.2 解析命令行参数 . . . . .	4
2.3 main 函数 . . . . .	4
2.4 实验截图 . . . . .	4
<b>3 实验心得</b>	<b>5</b>

# 1 实现内核模块

## 1.1 参数传递

本内核模块需要接收三个参数，一个 `int` 型的操作数，一个 `int` 数组，以及一个 `char*` 型的字符串表示当前的操作。因为 C array 没法获得长度，所以还需要额外的一个 `int` 型的变量来表示数组的长度。

本处使用了 `module_param` 来读取非数组的参数，权限设置成 0，使得这些参数只能被内核模块自己修改。对于数组参数，使用了 `module_param_array`，该宏的原型是

---

```
1 #define module_param_array(name, type, nump, perm)      \
2     module_param_array_named(name, name, type, nump, perm)
```

---

我们按照顺序，先传入数组，再传入类型 (`int`)，接着传入数组长度的参数，最后设置权限为 0。

## 1.2 内核模块的初始化

本实验的要求是创建 `proc` 文件 `/proc/<ID>/calc`，因此需要先通过 `proc_mkdir` 创建 `/proc/<ID>` 目录，如果本步骤无法完成，那么释放 `proc` 资源，并且打 Error log，然后返回。如果能够成功创建子目录，那么就可以通过 `proc_create` 创建 `calc` 文件，权限设置为 0644，表示内核可读写，但是其他均只能读。如果创建失败，那么释放之前分配的所有资源，然后打 Error log，并且返回。反之，则完成了初始化的工作。

## 1.3 内核模块的清理

一个成功初始化的该内核模块一共有两个资源，一个是 `proc` 文件，一个是 `proc/<id>` 目录。先释放 `proc` 文件，然后释放目录，即完成了清理工作。

## 1.4 内核模块的读

实验要求是当用户读取文件时，输出以逗号分隔的数组结果。我的处理逻辑如下，根据操作符的类型，在迭代数组的同时完成对应的操作，同时将结果转成字符串 `append` 至一个临时数组中。在迭代完成后，将该数组从 `kernel` 拷贝到用户态，即完成了读的操作。

在完成的过程中有以下几点需要注意。

首先，因为 `kernel programming` 不允许使用常规头文件，我们需要使用 `strcpy` 和 `snprintf` 来完成字符串的比较和格式化拼接。所以我们该函数被包含在了 `linux/string.h` 中，引用之即可。

其次，由于 C string 是以 `backslash 0` 作为结尾的，我们需要将格式化后的字符串数组的最后一位设置为 `backslash 0`，以防用户读取出错。此外，由于数组的最后一个元素后无需输出逗号，我们可以直接修改格式化数组的倒数第二位，将其从逗号改为 `backslash n`。

## 1.5 内核模块的写

实验要求是当用户传入一个数字后，将其作为新的操作数。我的处理逻辑如下，首先从用户态把 buffer 拷贝到 kernel buffer 之中，然后检查操作数是否是一个合法的整数，如果是则将其赋值给操作数，否则返回错误。

本部分实现较为简单，只需要使用 `kstrtoint` 函数即可。其他的错误处理和一般处理方法一致。

## 1.6 实验截图

```
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ sudo insmod calc.ko operand1=2 operand2=1,2,3,4,5 operator=add
[sudo] password for ubuntu:
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ cat /proc/521021910653/calc
3,4,5,6,7
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ echo 3 > /proc/521021910653/calc
bash: /proc/521021910653/calc: Permission denied
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ echo 3 | sudo tee /proc/521021910653/calc
3
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ cat /proc/521021910653/calc
4,5,6,7,8
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ sudo rmmod calc
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$
```

图 1: Add 操作截图

```
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ sudo insmod calc.ko operand1=2 operand2=1,2,3,4,5 operator=mul
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ cat /proc/521021910653/calc
2,4,6,8,10
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ echo 3 | sudo tee /proc/521021910653/calc
3
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ cat /proc/521021910653/calc
3,6,9,12,15
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$ sudo rmmod calc
ubuntu@ubuntu:~/Desktop/CS3302-2025-Spring/project1/calc$
```

图 2: Mul 操作截图

## 1.7 实验心得

本实验在完成的时候遇到了 `echo 3 > /proc/<id>/calc` 报权限不够的错，说明重定向的时候没有权限写入，该问题是无法通过给 `echo` 加 `sudo` 权限解决的。因此，我在写的时候，直接使用了 `echo 3 | sudo tee /proc/<id>/calc`，`tee` 是一个文件读写的操作，这样就可以成功写入了。



```

PROBLEMS OUTPUT TERMINAL PORTS GITLENS DEBUG CONSOLE
1128 S /usr/libexec/gvfs-goa-volume-monitor
1124 S /usr/libexec/goa-daemon
1133 S /usr/libexec/goa-identity-service
1139 S /usr/libexec/gvfs-afc-volume-monitor
1146 S /usr/libexec/gvfs-gphoto2-volume-monitor
1158 S /usr/libexec/upowerd
1334 S /usr/libexec/packagekitd
1499 S /usr/libexec/colord
1524 S /usr/sbin/spice-vdagentd
4384 I [kworker/2:1-mm_percpu_wq]
7469 S gdm-session-worker [pam/gdm-password]
7484 S /usr/bin/gnome-keyring-daemon --daemonize --login
7488 S /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
7492 S /usr/libexec/gnome-session-binary --session=ubuntu
7532 S /usr/libexec/gnome-session-ctl --monitor
7545 S /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
7549 I [kworker/0:3-events]
7568 S /usr/libexec/at-spi-bus-launcher --launch-immediately
7569 S /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-address 11 --address=unix:path=/run/user/1000/at-spi/bus
7570 S /usr/bin/gnome-shell
7585 S /usr/libexec/xdg-permission-store
7587 S /usr/libexec/gnome-shell-calendar-server
7597 S /usr/libexec/evolution-source-registry
7598 S /usr/bin/pulseaudio --daemonize=no --log-target=journal
7603 S /usr/libexec/dconf-service
7615 S /usr/libexec/evolution-calendar-factory
7623 S /usr/libexec/gvfsd-trash --spawner /1.4/org/gtk/gvfs/exec_spaw/0
7635 S /usr/libexec/evolution-addressbook-factory
7638 S /usr/libexec/at-spi2-registryd --use-gnome-session
7639 S /usr/bin/gjs /usr/share/gnome-shell/org.gnome.Shell.Notifications
7650 S sh -c /usr/bin/ibus-daemon --panel disable ${[ "$XDG_SESSION_TYPE" = "x11" ]}
86 echo "--xim"
7651 S /usr/libexec/gsd-all-ly-settings
7654 S /usr/bin/ibus-daemon --panel disable
1133 S /usr/libexec/goa-identity-service
1139 S /usr/libexec/gvfs-afc-volume-monitor
1146 S /usr/libexec/gvfs-gphoto2-volume-monitor
1158 S /usr/libexec/upowerd
1334 S /usr/libexec/packagekitd
1499 S /usr/libexec/colord
1524 S /usr/sbin/spice-vdagentd
4384 I [kworker/2:1-events]
7469 S gdm-session-worker [pam/gdm-password]
7484 S /usr/bin/gnome-keyring-daemon --daemonize --login
7488 S /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
7492 S /usr/libexec/gnome-session-binary --session=ubuntu
7532 S /usr/libexec/gnome-session-ctl --monitor
7545 S /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
7549 I [kworker/0:3-events]
7568 S /usr/libexec/at-spi-bus-launcher --launch-immediately
7569 S /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-address 11 --address=unix:path=/run/user/1000/at-spi/bus
7570 S /usr/bin/gnome-shell
7585 S /usr/libexec/xdg-permission-store
7587 S /usr/libexec/gnome-shell-calendar-server
7597 S /usr/libexec/evolution-source-registry
7598 S /usr/bin/pulseaudio --daemonize=no --log-target=journal
7603 S /usr/libexec/dconf-service
7615 S /usr/libexec/evolution-calendar-factory
7623 S /usr/libexec/gvfsd-trash --spawner /1.4/org/gtk/gvfs/exec_spaw/0
7635 S /usr/libexec/evolution-addressbook-factory
7638 S /usr/libexec/at-spi2-registryd --use-gnome-session
7639 S /usr/bin/gjs /usr/share/gnome-shell/org.gnome.Shell.Notifications
7650 S sh -c /usr/bin/ibus-daemon --panel disable ${[ "$XDG_SESSION_TYPE" = "x11" ]}
86 echo "--xim"
7651 S /usr/libexec/gsd-all-ly-settings
7654 S /usr/bin/ibus-daemon --panel disable

```

图 4: 截图 2

### 3 实验心得

经过本次实验，我了解如何编写内核模块，并且通过 kernel 提供的接口完成 kernel 和用户态的交互；也学会了以用户态的身份去获取部分内核信息。这是对庞大 linux 内核的一个很好的入门，也是对操作系统的一个很好的实践。

我的实验代码开源在 [GitHub](#) 上。