中南大學
CENTRAL SOUTH UNIVERSITY

Linux 内核模块编程
实验报告

学生姓名	王雅蓉
学　　号	8206200602
专业班级	计科 2003
指导教师	沈海澜
学　　院	计算机学院
完成时间	2023 年 06 月 11 日

计算机学院

2023 年 06 月

目录

第 1 章 实验概述

1.1 实验目的

（1）通过模块编程实验的编写设计，加深对 Linux 内核的理解；

（2）明确系统内存空间与用户内存空间的区别；

（3）通过对模块加载、卸载等操作的学习，掌握内核编程的方法；

1.2 准备知识

（1）了解内核模块的概念和特点。

（2）掌握内核编程的相关知识

（3）掌握内核模块的结构、编译和加载、初始化与清理函数、模块的卸载等方法。

1.3 实验内容

设计一个模块，该模块的功能是列出系统中所有内核进程的程序名、PID 号和进程状态。本实验总共完成了以下内容：

（1）编写模块获取系统内所有进程，遍历进程输出他们的程序名、PID 和进程状态

（2）编写 Makefile 文件编译模块

（3）试挂载该模块并移除该模块

1.4 实验要求

（1）掌握 Linux 环境下应用程序的编辑、运行、调试方法

（2）完成实验内容要求实现的任务，并记录实验过程

（3）撰写实验报告，包括实验目的、实验内容、实验要求、实验代码及运行截图、实验心得等。

第 2 章 实验设计

2.1 模块设计

该模块要实现的功能需要获取当前系统进程。实验中采用 Linux 内核中常用的数据结构进程控制块 task_struct 来获取各个系统进程的信息。采用内核宏定义 for_each_process(task)对进程链中的进程进行遍历。遍历过程中通过 task_struct 控制块的内部信息获取进程消息：task->comm, task->pid, task->stats，分别表示进程的进程名、PID、进程状态。核心代码实现如下：

```
1.     struct task_struct *task;
2.
3.         printk(KERN_INFO "List all tasks in the system:\n");
4.      for_each_process(task)
5.         {
6.             printk(KERN_INFO "Process name:%s\tPID:%d\tState:%ld\n",
   task->comm, task->pid, task->stats);
7.         }
8.         return 0;
```

同时在模块中要定义模块装入时、卸载时执行的函数，模块装入时执行的函数实现主要功能，包含以上代码。模块卸载时只需定义卸载函数并用 module_exit 调用即可。核心代码实现如下：

```
1.     static void __exit list_exit(void)
2.     {
3.         printk(KERN_INFO "Goodbye!\n");
4.     }
5.
6.      module_init(list_init);
7.      module_exit(list_exit);
```

2.2 Makefile 文件编写

Makefile 可以简单的认为是一个工程文件的编译规则，描述了整个工程的编译和链接等规则。编译整个工程需要涉及到的，在 Makefile 中都可以进行描述。该文件需要 make 指令解释，简化编译过程里所下达的命令，当执行 make 时，make 会在当前目录下搜寻 Makefile 文本文件，执行对应的操作。本程序由于要执行的模块功能较为单一，所以 Makefile 文件中的内容也较为简单。

第 3 章 代码及运行结果

3.1 research4.c

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/sched/signal.h>
MODULE_LICENSE("GPL");
static int __init list_init(void)
{
    struct task_struct *task;

    printk(KERN_INFO "List all tasks in the system:\n");
    for_each_process(task)
    {
        printk(KERN_INFO "Process name:%s\tPID:%d\tState:%ld\n",
task->comm, task->pid, task->stats);
    }
    return 0;
}

static void __exit list_exit(void)
{
    printk(KERN_INFO "Goodbye!\n");
}

    module_init(list_init);
    module_exit(list_exit);

MODULE_AUTHOR("Your name");
MODULE_DESCRIPTION("List all tasks in the system");
```

3.2 Makefile

```
obj-m += research4.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

3.3 运行结果

```
felicia@felicia-virtual-machine:~/Linux_research/research3/research4$ make
make -C /lib/modules/5.19.0-43-generic/build M=/home/felicia/Linux_research/r
esearch3/research4 modules
```

图 3.3（1） 运行结果 1

```
  MODPOST /home/felicia/Linux_research/research3/research4/Module.symvers
  CC [M]  /home/felicia/Linux_research/research3/research4/research4.mod.o
  LD [M]  /home/felicia/Linux_research/research3/research4/research4.ko
  BTF [M] /home/felicia/Linux_research/research3/research4/research4.ko
```

图 3.3（2）运行结果 2

```
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089858] Process name:rcu_gp      PID:3   State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089859] Process name:rcu_par_gp  PID:4   State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089861] Process name:slub_flushwq        PID:5   State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089862] Process name:netns       PID:6   State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089864] Process name:kworker/0:0H        PID:8   State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089865] Process name:mm_percpu_wq        PID:10  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089867] Process name:rcu_tasks_kthre     PID:11  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089868] Process name:rcu_tasks_rude_     PID:12  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089870] Process name:rcu_tasks_trace     PID:13  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089871] Process name:ksoftirqd/0 PID:14  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089872] Process name:rcu_preempt PID:15  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089873] Process name:migration/0 PID:16  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089875] Process name:idle_inject/0       PID:17  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089876] Process name:cpuhp/0     PID:19  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089878] Process name:cpuhp/1     PID:20  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089879] Process name:idle_inject/1       PID:21  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089880] Process name:migration/1 PID:22  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089881] Process name:ksoftirqd/1 PID:23  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089883] Process name:kworker/1:0H        PID:25  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089884] Process name:cpuhp/2     PID:26  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089886] Process name:idle_inject/2       PID:27  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089887] Process name:migration/2 PID:28  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089888] Process name:ksoftirqd/2 PID:29  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089890] Process name:kworker/2:0H        PID:31  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089891] Process name:cpuhp/3     PID:32  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089892] Process name:idle_inject/3       PID:33  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089894] Process name:migration/3 PID:34  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089895] Process name:ksoftirqd/3 PID:35  State:0
Jun 12 22:24:04 felicia-virtual-machine kernel: [ 1391.089896] Process name:kworker/3:0H        PID:37  State:0
```

图 3.3（3）运行结果 3

总结与收获

经过本次 Linux 内核模块编程实验，我解决了课堂上许多有待解决的问题，更加深刻的了解了 Linux 内核模块的作用。在之前也只是对 Makefile 文件的作用了解，但本次实验中真正实操让我意识到其真正的强大功能还有待进一步学习增强。

实验中起初比较令我头疼的就是一直输出不到目标结果，反复调试、修改权限、分析错误结果后终于输出了正确的目标结果，解决了困扰一天的难题。

在未来的 Linux 系统与应用的学习中，我会更加深入学习课堂上没有深入讲解的内容，更注重自身知识的深度与厚度积累，积极探索，认真学习，在今后的实验中精益求精的要求自己。