

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Поволжский государственный университет телекоммуникаций и
информатики»

Факультет информатики и вычислительной техники

Кафедра При

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

“Kernel API — работа с процессами и памятью”

Дисциплина: Разработка драйверов

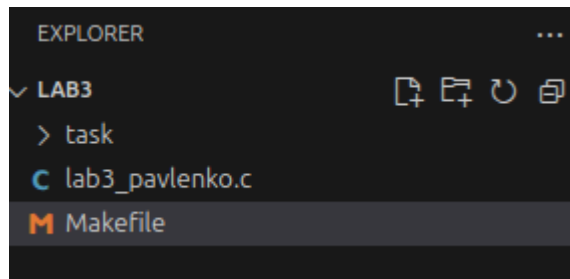
Выполнил: студент
При-21 Павленко Я.И.

Самара 2024

Цель работы: Знакомство с приемами программирования ОС GNU/Linux, создание простого модуля ядра системы, работа с процессами и памятью.

Вариант 14

Задание 1. В рабочем каталоге пользователя создать каталог lab3. В данном каталоге создать файл исходного кода модуля ядра с именем lab3_ivanov.c (вместо ivanov вставить свою фамилию транслитерацией).



Задание 2. На основе результата выполнения предыдущей лабораторной работы, составить программу модуля ядра системы с одним параметром (int). Данный параметр должен использоваться для передачи значения PID процесса системы. Необходимо создать структуру task_info, содержащую поле для PID процесса, и для временной метки. При загрузке модуля необходимо выделить память для экземпляра данной структуры с помощью kmalloc(). При выгрузке, модуль должен выводить строку, содержащую данный номер процесса и количество системных тиков, прошедших с момента загрузки системы. Также, при выгрузке, необходимо очистить память с помощью функции kfree().

```
C lab3_pavlenko.c x Makefile
C lab3_pavlenko.c
1  #include <linux/init.h>
2  #include <linux/module.h>
3  #include <linux/kernel.h>
4  #include <linux/sched.h>
5  #include <linux/sched/signal.h>
6  #include <linux/slab.h>
7  #include <linux/jiffies.h>
8  #include <linux/moduleparam.h>
9
10
11  MODULE_LICENSE("GPL");
12  MODULE_AUTHOR("Yaroslav Pavlenko");
13  MODULE_DESCRIPTION("Lab3, task 2: memory + PID");
14  MODULE_VERSION("0.01");
15
16
17  static int pid = 1;
18  module_param(pid, int, S_IRUGO);
19  MODULE_PARM_DESC(pid, "Process ID to record");
20
21
22  struct task_info {
23      pid_t t_pid;
24      unsigned long timestamp;
25  };
26
27
28  static struct task_info *ti1;
29
30
31  static struct task_info *task_info_alloc_pavlenko(int pid)
32  {
33      struct task_info *ti;
34
```

```
C lab3_pavlenko.c x Makefile
C lab3_pavlenko.c
30
31  static struct task_info *task_info_alloc_pavlenko(int pid)
32  {
33      struct task_info *ti;
34
35      ti = kmalloc(sizeof(*ti), GFP_KERNEL);
36
37
38      if (ti == NULL)
39          return NULL;
40
41      ti->t_pid = pid;
42      ti->timestamp = jiffies;
43
44      return ti;
45  }
46
47
48
49  static int __init lab3_init_pavlenko(void)
50  {
51      pr_info("lab3_pavlenko: loading module...\n");
52
53
54      ti1 = task_info_alloc_pavlenko(pid ? pid : current->pid);
55
56
57      if (!ti1) {
58          pr_err("lab3_pavlenko: memory allocation failed!\n");
59          return -ENOMEM;
60      }
61
62
```

```
C lab3_pavlenko.c x M Makefile
C lab3_pavlenko.c
51 {
61 }
62
63 pr_info("lab3_pavlenko: module loaded successfully!\n");
64 return 0;
65 }
66
67 static void __exit lab3_exit_pavlenko(void)
68 {
69     pr_info("lab3_pavlenko: unloading module...\n");
70
71     if (t1) {
72         pr_info("lab3_pavlenko: PID = %d, Timestamp = %lu\n",
73             t1->t_pid, t1->timestamp);
74         kfree(t1);
75         t1 = NULL;
76     }
77
78     pr_info("lab3_pavlenko: module unloaded.\n");
79 }
80
81 module_init(lab3_init_pavlenko);
82 module_exit(lab3_exit_pavlenko);
83
84
85
86
87
```

```
C lab3_pavlenko.c M Makefile x
M Makefile
1
2 obj-m += lab3_pavlenko.o
3
4
5 KDIR := /lib/modules/$(shell uname -r)/build
6 PWD := $(shell pwd)
7
8
9 all:
10     make -C $(KDIR) M=$(PWD) modules
11
12 clean:
13     make -C $(KDIR) M=$(PWD) clean
14
15 test:
16     sudo dmesg -C
17     sudo insmod lab3_pavlenko.ko pid=1
18     sudo rmmod lab3_pavlenko
19     sudo dmesg | tail -n 30
20
21
```

```

iarik01@iarik01:~/RDU/lab3$ make test
sudo dmesg -C
sudo insmod lab3_pavlenko.ko pid=1
sudo rmmod lab3_pavlenko
sudo dmesg | tail -n 30
[14056.911194] lab3_pavlenko: loading module...
[14056.911198] lab3_pavlenko: module loaded successfully!
[14056.997610] lab3_pavlenko: unloading module...
[14056.997613] lab3_pavlenko: PID = 1, Timestamp = 43087240
13
[14056.997615] lab3_pavlenko: module unloaded.
iarik01@iarik01:~/RDU/lab3$

```

Задание 3. В соответствии с таблицей вариантов, модифицировать программу.

№ варианта	Задание
4	<p>Добавить к структуре <code>task_info</code> еще один элемент, строковый тип. При вызове <code>task_info_alloc</code> присвоить этому элементу в качестве значения фамилию (транслитом) студента, выполняющего работу.</p> <p>При загрузке модуля вызвать <code>task_info_alloc</code> для процесса, который будет после следующего за текущим (<code>next_task(next_task(current))->pid</code>) и вывести по нему аналогичную информацию при выгрузке модуля (номер процесса и количество системных тиков, прошедших с момента загрузки системы).</p>

```

C lab3_pavlenko.c x  Makefile
C lab3_pavlenko.c
23 struct task_info {
24     pid_t      t_pid;
25     unsigned long timestamp;
26     char       student[32];
27 };
28
29 static struct task_info *t1;
30
31
32 static pid_t choose_next_next_of_current(void)
33 {
34     pid_t res;
35
36     rcu_read_lock();
37     {
38         struct task_struct *t1 = next_task(current);
39         struct task_struct *t2 = next_task(t1);
40         res = task_pid_nr(t2);
41     }
42     rcu_read_unlock();
43
44     return res;
45 }

```

```
lab3_pavlenko.c x Makefile
lab3_pavlenko.c
65
66 static int __init lab3_init_pavlenko(void)
67 {
68     pr_info("lab3_pavlenko: loading module...\n");
69
70
71     pid_t target = choose_next_next_of_current();
72
73     til = task_info_alloc_pavlenko(target);
74     if (!til) {
75         pr_err("lab3_pavlenko: memory allocation failed!\n");
76         return -ENOMEM;
77     }
78
79     pr_info("lab3_pavlenko: init ok, student=%s, target pid=%d, jiffies(now)=%lu\n",
80           til->student, til->t_pid, jiffies);
81     return 0;
82 }
83
84
85 static void __exit lab3_exit_pavlenko(void)
86 {
87     pr_info("lab3_pavlenko: unloading module...\n");
88
89     if (til) {
90         pr_info("lab3_pavlenko: student=%s, PID=%d, jiffies(now)=%lu\n",
91               til->student, til->t_pid, jiffies);
92         kfree(til);
93         til = NULL;
94     }
95
96     pr_info("lab3_pavlenko: module unloaded.\n");
97 }
98
99 module_init(lab3_init_pavlenko);
```

```
iarik01@iarik01:~/RDU/lab3$ make test
sudo dmesg -C
[sudo] password for iarik01:
sudo insmod lab3_pavlenko.ko pid=1
sudo rmmod lab3_pavlenko
sudo dmesg | tail -n 30
[16080.805424] lab3_pavlenko: loading module...
[16080.805429] lab3_pavlenko: init ok, student=pavlenko, target pid=1, jiffies(now)=4310747897
[16080.890768] lab3_pavlenko: unloading module...
[16080.890772] lab3_pavlenko: student=pavlenko, PID=1, jiffies(now)=4310747981
[16080.890773] lab3_pavlenko: module unloaded.
Show Apps iarik01:~/RDU/lab3$
```