Создание и загрузка простого модуля

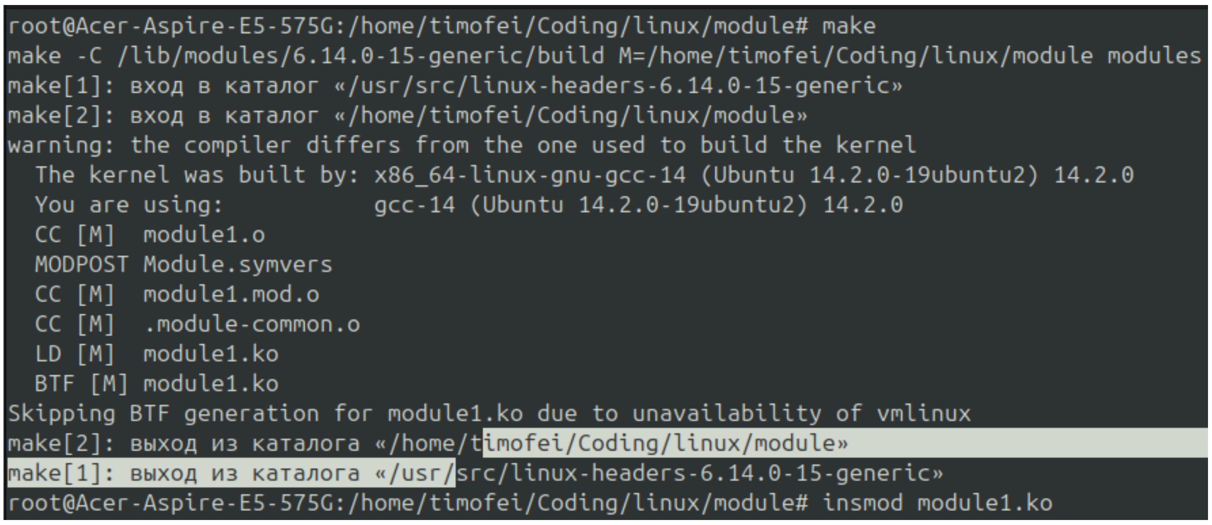
1. Код модуля и Makefile

```c
#include <linux/module.h>
#include <linux/kernel.h>

int init_module(void){
   pr_info("My 'module1' module loaded!!!!\n");
   return 0;
}

void cleanup_module(void){
   pr_info("My 'module1' module unloaded!!!!\n");
}

MODULE_LICENSE("GPL");
```

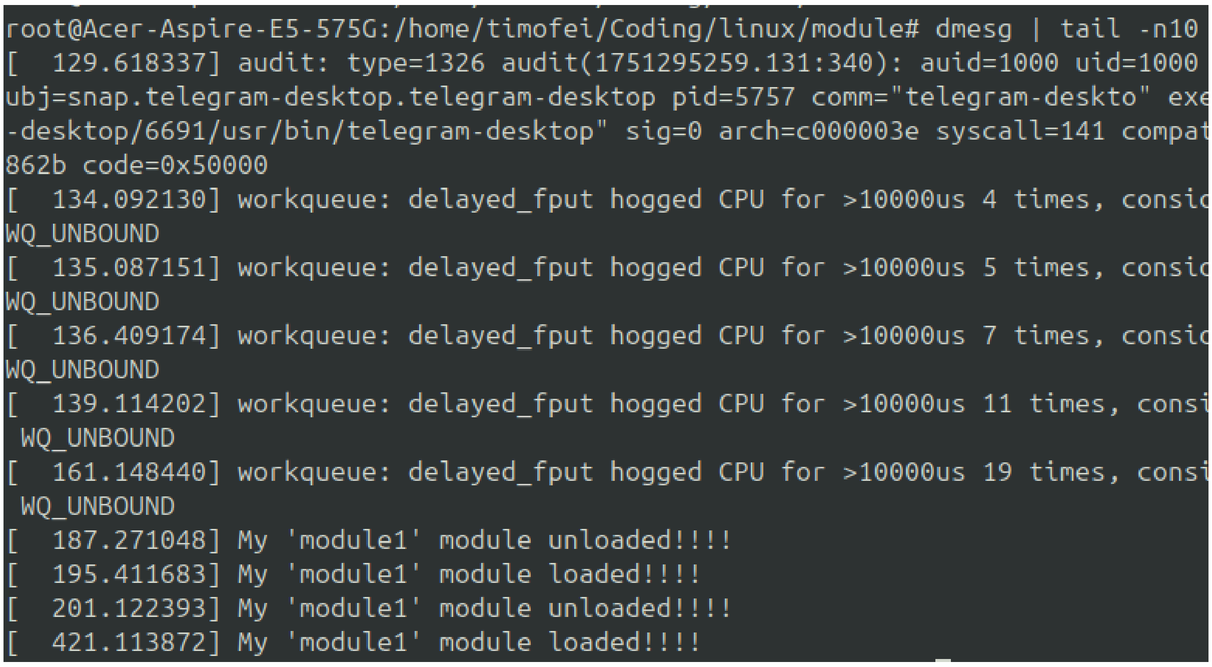
Код модуля содержит функции инициализации и очистки с выводом сообщения в системный журнал.

```makefile
obj-m += module1.o
#ccflags-y += -g -DDEBUG

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

2. Сборка модуля и загрузка

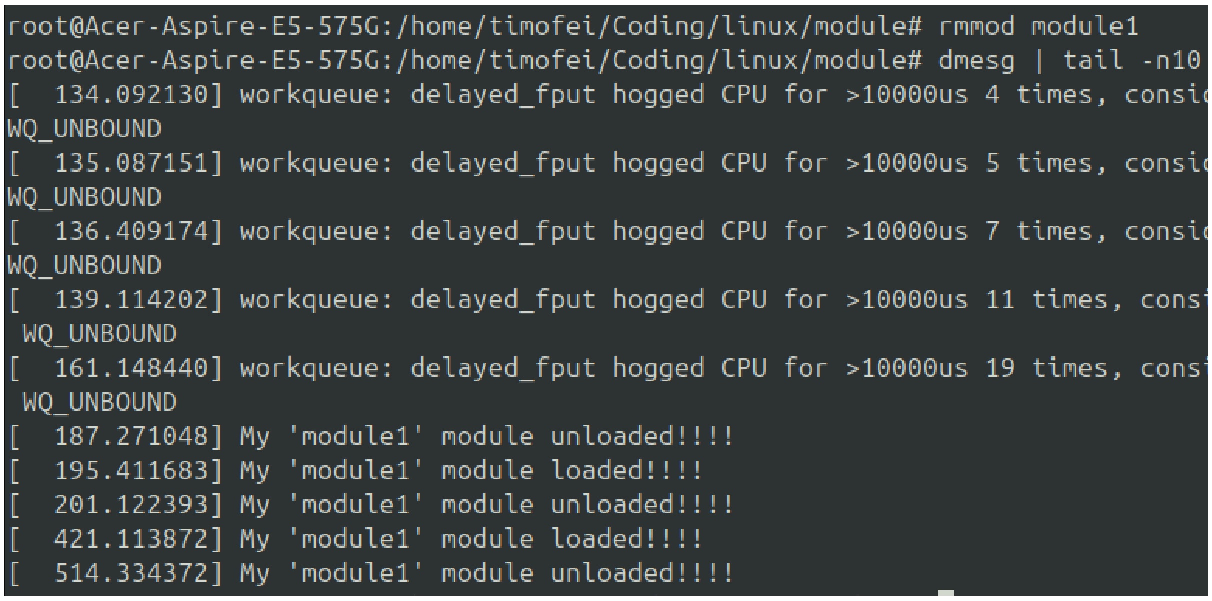


```
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# make
make -C /lib/modules/6.14.0-15-generic/build M=/home/timofei/Coding/linux/module modules
make[1]: вход в каталог «/usr/src/linux-headers-6.14.0-15-generic»
make[2]: вход в каталог «/home/timofei/Coding/linux/module»
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: x86_64-linux-gnu-gcc-14 (Ubuntu 14.2.0-19ubuntu2) 14.2.0
  You are using:           gcc-14 (Ubuntu 14.2.0-19ubuntu2) 14.2.0
  CC [M]  module1.o
  MODPOST Module.symvers
  CC [M]  module1.mod.o
  CC [M]  .module-common.o
  LD [M]  module1.ko
  BTF [M] module1.ko
Skipping BTF generation for module1.ko due to unavailability of vmlinux
make[2]: выход из каталога «/home/timofei/Coding/linux/module»
make[1]: выход из каталога «/usr/src/linux-headers-6.14.0-15-generic»
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# insmod module1.ko
```



```
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# dmesg | tail -n10
[  129.618337] audit: type=1326 audit(1751295259.131:340): auid=1000 uid=1000
ubj=snap.telegram-desktop.telegram-desktop pid=5757 comm="telegram-deskto" exe
-desktop/6691/usr/bin/telegram-desktop" sig=0 arch=c000003e syscall=141 compat
862b code=0x50000
[  134.092130] workqueue: delayed_fput hogged CPU for >10000us 4 times, consic
WQ_UNBOUND
[  135.087151] workqueue: delayed_fput hogged CPU for >10000us 5 times, consic
WQ_UNBOUND
[  136.409174] workqueue: delayed_fput hogged CPU for >10000us 7 times, consic
WQ_UNBOUND
[  139.114202] workqueue: delayed_fput hogged CPU for >10000us 11 times, consi
 WQ_UNBOUND
[  161.148440] workqueue: delayed_fput hogged CPU for >10000us 19 times, consi
 WQ_UNBOUND
[  187.271048] My 'module1' module unloaded!!!!
[  195.411683] My 'module1' module loaded!!!!
[  201.122393] My 'module1' module unloaded!!!!
[  421.113872] My 'module1' module loaded!!!!
```

3. Выгрузка модуля



```
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# rmmod module1
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# dmesg | tail -n10
[  134.092130] workqueue: delayed_fput hogged CPU for >10000us 4 times, consic
WQ_UNBOUND
[  135.087151] workqueue: delayed_fput hogged CPU for >10000us 5 times, consic
WQ_UNBOUND
[  136.409174] workqueue: delayed_fput hogged CPU for >10000us 7 times, consic
WQ_UNBOUND
[  139.114202] workqueue: delayed_fput hogged CPU for >10000us 11 times, consi
 WQ_UNBOUND
[  161.148440] workqueue: delayed_fput hogged CPU for >10000us 19 times, consi
 WQ_UNBOUND
[  187.271048] My 'module1' module unloaded!!!!
[  195.411683] My 'module1' module loaded!!!!
[  201.122393] My 'module1' module unloaded!!!!
[  421.113872] My 'module1' module loaded!!!!
[  514.334372] My 'module1' module unloaded!!!!
```

- Модуль module1 был корректно скомпилирован, загружен и выгружен.
- Сообщения из модуля были успешно записаны в системный лог.
- Отсутствуют ошибки загрузки модуля или конфликты версий.
- Работа с модулем соответствует стандартному процессу разработки и тестирования простых модулей ядра Linux.

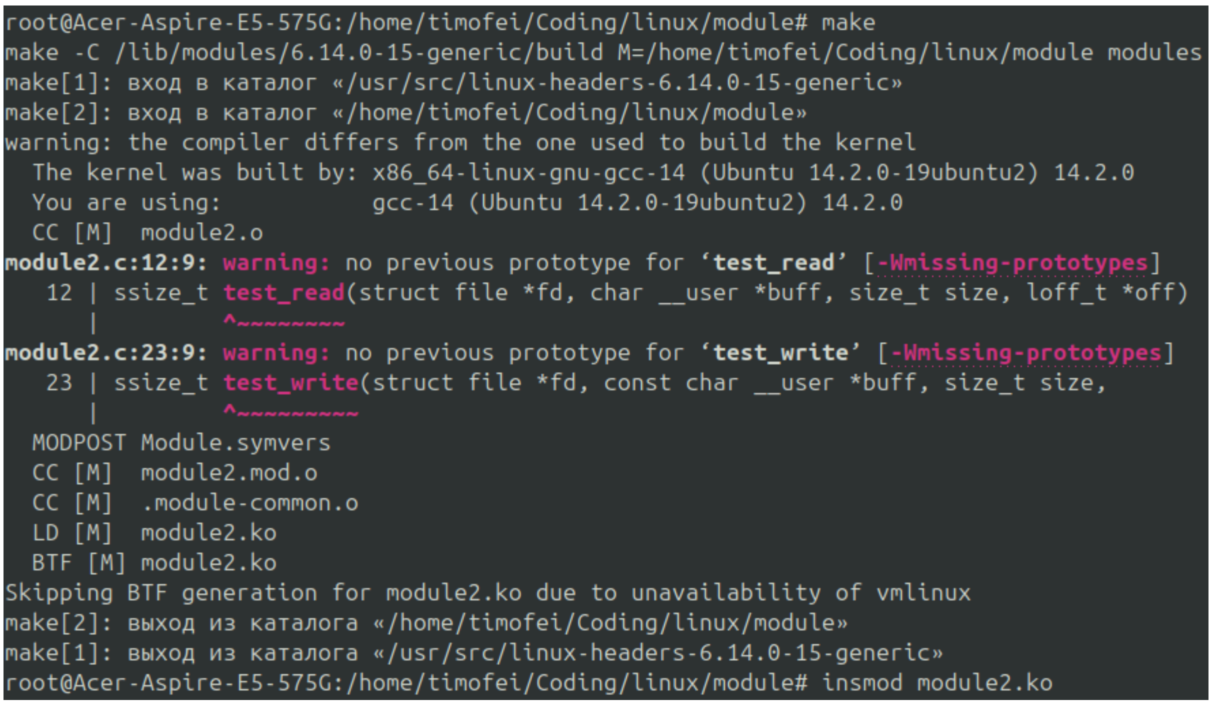
Создание файла устройств и модуля для файла устройств

1.

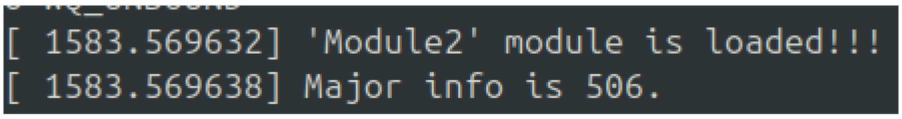
```c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/rwlock.h>
#include <linux/uaccess.h>

static int major = 0;

static rwlock_t lock;
static char test_string[15] = "Hello!\0\n";

ssize_t test_read(struct file *fd, char __user *buff, size_t size,
loff_t *off)
{
   size_t rc;

   read_lock(&lock);
   rc = simple_read_from_buffer(buff, size, off, test_string, 15);
   read_unlock(&lock);

   return rc;
}

ssize_t test_write(struct file *fd, const char __user *buff, size_t
size,
          loff_t *off)
{
   size_t rc = 0;
   if (size > 15) {
       return -EINVAL;
   }

   write_lock(&lock);
   rc = simple_write_to_buffer(test_string, 15, off, buff, size);
   write_unlock(&lock);

   return rc;
}

static struct file_operations fops = { .owner = THIS_MODULE,
                      .read = test_read,
                       .write = test_write };

int init_module(void)
{
   pr_info("'Module2' module is loaded!!!\n");
   rwlock_init(&lock);
   major = register_chrdev(major, "module2", &fops);

   if (major < 0) {
       return major;
   }

   pr_info("Major info is %d.\n", major);

   return 0;
}

void cleanup_module(void)
{
   pr_info("'Module2' module is unloaded!!!\n");
   unregister_chrdev(major, "module2");
}

MODULE_LICENSE("GPL");
```

```
obj-m += module2.o
#ccflags-y += -g -DDEBUG

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

2. Сборка модуля и загрузка

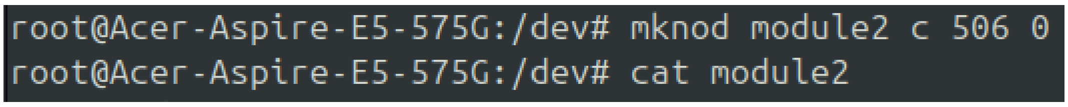


```
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# make
make -C /lib/modules/6.14.0-15-generic/build M=/home/timofei/Coding/linux/module modules
make[1]: вход в каталог «/usr/src/linux-headers-6.14.0-15-generic»
make[2]: вход в каталог «/home/timofei/Coding/linux/module»
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: x86_64-linux-gnu-gcc-14 (Ubuntu 14.2.0-19ubuntu2) 14.2.0
  You are using:           gcc-14 (Ubuntu 14.2.0-19ubuntu2) 14.2.0
  CC [M]  module2.o
module2.c:12:9: warning: no previous prototype for ‘test_read’ [-Wmissing-prototypes]
   12 | ssize_t test_read(struct file *fd, char __user *buff, size_t size, loff_t *off)
      |         ^~~~~~~~~
module2.c:23:9: warning: no previous prototype for ‘test_write’ [-Wmissing-prototypes]
   23 | ssize_t test_write(struct file *fd, const char __user *buff, size_t size,
      |         ^~~~~~~~~~
  MODPOST Module.symvers
  CC [M]  module2.mod.o
  CC [M]  .module-common.o
  LD [M]  module2.ko
  BTF [M] module2.ko
Skipping BTF generation for module2.ko due to unavailability of vmlinux
make[2]: выход из каталога «/home/timofei/Coding/linux/module»
make[1]: выход из каталога «/usr/src/linux-headers-6.14.0-15-generic»
root@Acer-Aspire-E5-575G:/home/timofei/Coding/linux/module# insmod module2.ko
```

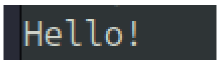


```
[ 1583.569632] 'Module2' module is loaded!!!
[ 1583.569638] Major info is 506.
```

3. создание файла module2 в /dev

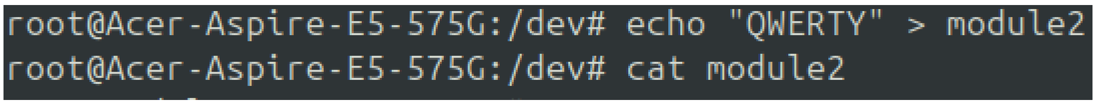


```
root@Acer-Aspire-E5-575G:/dev# mknod module2 c 506 0
root@Acer-Aspire-E5-575G:/dev# cat module2
```

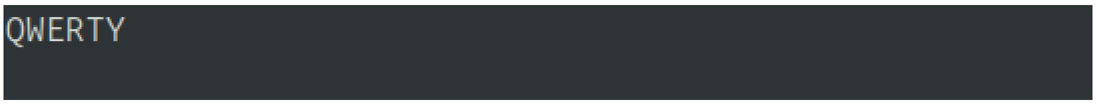


```
Hello!
```

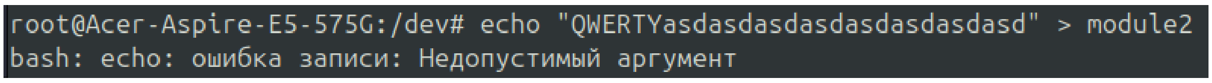
4. Запись в файл и переполнение



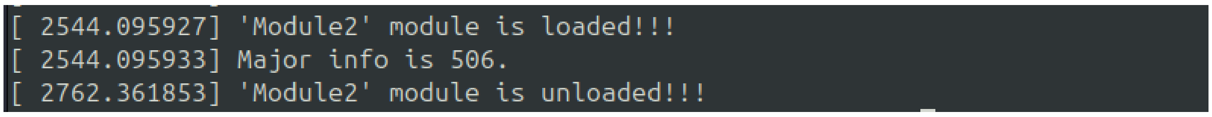
```
root@Acer-Aspire-E5-575G:/dev# echo "QWERTY" > module2
root@Acer-Aspire-E5-575G:/dev# cat module2
```



```
QWERTY
```



```
root@Acer-Aspire-E5-575G:/dev# echo "QWERTYasdasdasdasdasdasdasd" > module2
bash: echo: ошибка записи: Недопустимый аргумент
```



```
[ 2544.095927] 'Module2' module is loaded!!!
[ 2544.095933] Major info is 506.
[ 2762.361853] 'Module2' module is unloaded!!!
```

Файловые системы proc и sys

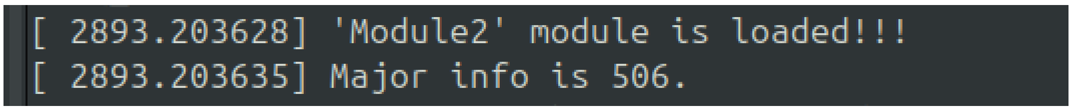
1. Код

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/rwlock.h>
#include <linux/proc_fs.h>
#include <linux/sysfs.h>
#include <linux/string.h>
#include <linux/kobject.h>

static int major = 0;
static struct proc_dir_entry *test = NULL;
static struct kobject *test_kobj = NULL;
static rwlock_t lock;
static char test_string[15] = "Hello!\0\n";

ssize_t test_read(struct file *fd, char __user *buff, size_t size,
loff_t *off)
{
  size_t rc;

  read_lock(&lock);
  rc = simple_read_from_buffer(buff, size, off, test_string, 15);
  read_unlock(&lock);

  return rc;
}

ssize_t test_write(struct file *fd, const char __user *buff, size_t
size,
        loff_t *off)
{
  size_t rc = 0;
  if (size > 15) {
     return -EINVAL;
  }

  write_lock_irq(&lock);
  rc = simple_write_to_buffer(test_string, 15, off, buff, size);
  write_unlock_irq(&lock);
  return rc;
}

ssize_t test_proc_read(struct file *fd, char __user *buff, size_t size,
            loff_t *off)
{
  size_t rc;

  rc = simple_read_from_buffer(buff, size, off, test_string, 15);

  return rc;
}

ssize_t test_proc_write(struct file *fd, const char __user *buff,
size_t size,
        loff_t *off)
{
  size_t rc;

  rc = simple_write_to_buffer(test_string, 15, off, buff, size);

  return rc;
}

static ssize_t string_show(struct kobject *kobj, struct kobj_attribute
*attr,
           char *buff)
{
  size_t rc = 0;
  memcpy(buff, test_string, 15);
  rc = strlen(test_string);
  return rc;
}

static ssize_t string_store(struct kobject *kobj, struct kobj_attribute
*attr,
           const char *buff, size_t count)
{
  size_t rc = 0;
  memcpy(test_string, buff, count);
  rc = strlen(buff);
  return rc;
}
```

```c
static struct file_operations fops = { .owner = THIS_MODULE,
                  .read = test_read,
                  .write = test_write };

static const struct proc_ops pops = {
  .proc_read = test_proc_read,
  .proc_write = test_proc_write,
};

static struct kobj_attribute string_attribute =
  __ATTR(test_string, 0644, string_show, string_store);

static struct attribute *attrs[] = {
  &string_attribute.attr,
  NULL,
};

static struct attribute_group attr_group = {
  .attrs = attrs,
};

int init_module(void)
{
  int retval = 0;
  pr_info("'Module2' module is loaded!!!\n");
  rwlock_init(&lock);
  major = register_chrdev(major, "module2", &fops);

  if (major < 0) {
     return major;
  }
  pr_info("Major info is %d.\n", major);

  test = proc_create("module2", 0666, NULL, &pops);

  test_kobj = kobject_create_and_add("kobject_test", kernel_kobj);

  if (!test_kobj) {
     return -ENOMEM;
  }

  retval = sysfs_create_group(test_kobj, &attr_group);
  if (retval) {
     kobject_put(test_kobj);
     return retval;
  }

  return 0;
}

void cleanup_module(void)
{
  proc_remove(test);
  kobject_put(test_kobj);
  unregister_chrdev(major, "module2");
  pr_info("'Module2' module is unloaded!!!\n");
}

MODULE_LICENSE("GPL");
```

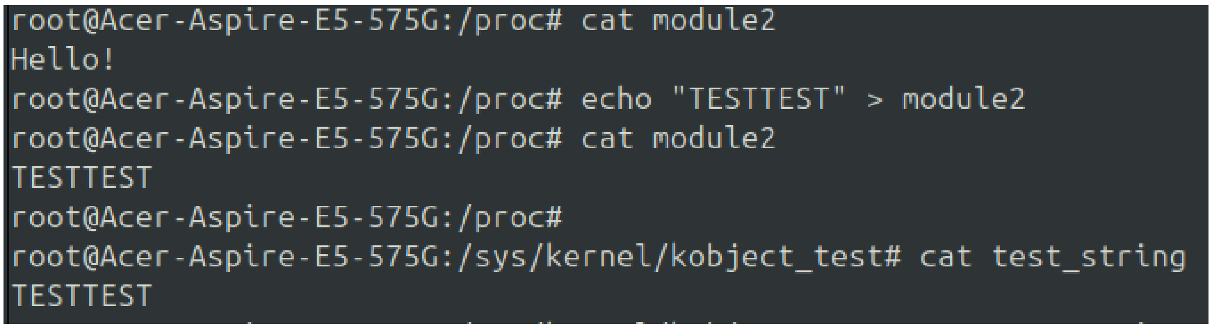
Создан файл /proc/module2 через proc_create().
Создан kobject с именем kobject_test в /sys/kernel/.

2.



```
[ 2893.203628] 'Module2' module is loaded!!!
[ 2893.203635] Major info is 506.
```

3. Результат



```
root@Acer-Aspire-E5-575G:/proc# cat module2
Hello!
root@Acer-Aspire-E5-575G:/proc# echo "TESTTEST" > module2
root@Acer-Aspire-E5-575G:/proc# cat module2
TESTTEST
root@Acer-Aspire-E5-575G:/proc#
root@Acer-Aspire-E5-575G:/sys/kernel/kobject_test# cat test_string
TESTTEST
```