Создаем файл test3.c со следующим содержанием:

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

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

static ssize_t test_proc_read(struct file *fd, char __user *buff, size_t
size, loff_t *off)
{
	size_t rc = 0;
	rc = simple_read_from_buffer(buff, size, off, test_string, 15);
	return rc;
}

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

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

static ssize_t test_store(struct kobject *kobj, struct kobj_attribute
*attr, const char *buf, size_t count)
{
	size_t rc = 0;
	if (count > 15)
		return -EINVAL;
	memcpy(test_string, buf, count);
```

```
	rc = strlen(buf);
	return rc;
}

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,
0664, test_show, test_store);
static struct attribute *attrs[] = {
	&string_attribute.attr,
	NULL,
};
static struct attribute_group attr_group = {
	.attrs = attrs,
};

int init_module(void)
{
	int ret = 0;
	pr_info("TEST MODULE IS LOADED!\n");
	test = proc_create("test3", 0666, NULL, &pops);
	test_kobj = kobject_create_and_add("kobj_test3", kernel_kobj);

	if (!test_kobj)
		return -ENOMEM;

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

	return 0;
}

void cleanup_module(void)
{
	proc_remove(test);
	kobject_put(test_kobj);
	pr_info("TEST MODULE IS UNLOADED!");
}

MODULE_LICENSE("GPL");
```

Создаем Makefile:

```
obj-m += test3.o

all:
	make -C /lib/modules/6.8.0-79-generic/build M=$(PWD) modules
clean:
	make -C /lib/modules/6.8.0-79-generic/build M=$(PWD) clean
```

Собираем модуль:

```
computer@computer-HP-Laptop-15-bw0xx:~/modules/3$ make
make -C /lib/modules/6.8.0-79-generic/build M=/home/computer/modules/3 modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-79-generic'
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
  You are using:           gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04.2) 12.3.0
  CC [M]  /home/computer/modules/3/test3.o
  MODPOST /home/computer/modules/3/Module.symvers
  CC [M]  /home/computer/modules/3/test3.mod.o
  LD [M]  /home/computer/modules/3/test3.ko
  BTF [M] /home/computer/modules/3/test3.ko
Skipping BTF generation for /home/computer/modules/3/test3.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-79-generic'
```

Устанавливаем:

```
computer@computer-HP-Laptop-15-bw0xx:~/modules/3$ sudo insmod test3.ko
[sudo] password for computer:
```

Заходим в каталог proc и проверяем наличие файла:

```
computer@computer-HP-Laptop-15-bw0xx:/proc$ ls -la | grep test
-rw-rw-rw-    1 root             root                             0 сен 28 22:14 test3
computer@computer-HP-Laptop-15-bw0xx:/proc$
```

Выводим строку на экран:

```
computer@computer-HP-Laptop-15-bw0xx:/proc$ cat test3
Hello!
```

Записываем новую строку:

```
computer@computer-HP-Laptop-15-bw0xx:/proc$ sudo echo "Goodbye!" > test3
[sudo] password for computer:
```

Снова выводим на экран:

```
computer@computer-HP-Laptop-15-bw0xx:/proc$ cat test3
Goodbye!
computer@computer-HP-Laptop-15-bw0xx:/proc$
```

Затем заходим в каталог sys и проверяем наличие каталога с файлом:

```
computer@computer-HP-Laptop-15-bw0xx:/sys/kernel$ ls | grep test3
kobj_test3
computer@computer-HP-Laptop-15-bw0xx:/sys/kernel$
```

```
computer@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3$ ls
test_string
computer@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3$
```

Выводим строку на экран:

```
computer@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3$ cat test_string
Goodbye!
```

Войдем в систему под суперпользователем и запишем новую строку:

```
root@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3# echo "Hello again!" > test_string
root@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3# cat test_string
Hello again!
root@computer-HP-Laptop-15-bw0xx:/sys/kernel/kobj_test3#
```