

## Сопроводительная документация

---

# Содержание

---

- [Описание](#)
- [Сборка](#)
- [Загрузка и выгрузка модуля](#)
- [Интерфейс драйвера](#)
  - [Символьное устройство](#)
  - [Атрибуты в sysfs](#)
- [Проверка работы устройств](#)
  - [Системные вызовы read/write](#)
  - [Системный вызов mmap](#)
  - [Атрибуты sysfs](#)
- [Авторство и лицензия](#)
  - [Автор](#)
  - [Лицензия](#)

# Описание

---

В данном модуле ядра Linux реализуется драйвер для абстрактного устройства. Тип драйвера: `platform_driver`.

При инициализации модуля создаётся 3 устройства типа `platform_device`.

При зондировании драйвер выделяет для устройства страницу памяти (4Кб), символьное устройство `cdev` и атрибуты в файловой системе `sysfs`.

Интерфейс драйвера и демонстрация работы описаны в соответствующих разделах этой документации.

# Сборка

---

Для сборки модуля необходимо выполнить `make`. Будет создано множество файлов в директории `/module`, среди которых необходимый для загрузки модуля `pseud.ko`. Также будет вызвана команда `make` для `/test/Makefile`, в результате чего будет создан исполняемый файл `test_mmap`.

Для очистки сборки выполнить `make clean`.

Собрать модуль и тесты можно отдельными командами: `make module` и `make test`.

Заголовочные файлы `linux-headers` можно явно указать следующим образом:

```
$ make HEADERS=/lib/modules/5.15.0-46-generic/build
```

# Загрузка и выгрузка модуля

---

Для загрузки модуля выполнить:

```
$ cd module
$ sudo insmod pseud.ko
```

Вывод команды `sudo dmesg`:

```
[12320.756172] pseud: Init
[12320.757445] pseud pseud.0: created
[12320.757617] pseud pseud.1: created
```

```
[12320.757724] pseud pseud.2: created
[12320.757777] pseud registered with major number 243
```

Вывод команды `lsmod | grep pseud`:

```
pseud                20480  0
```

Вывод команды `ls /dev/pseud*`:

```
/dev/pseud_0  /dev/pseud_1  /dev/pseud_2
```

Вывод команды `ls /sys/class/pseud/pseud_*`:

```
/sys/class/pseud/pseud_0:
dev device power subsystem uevent

/sys/class/pseud/pseud_1:
dev device power subsystem uevent

/sys/class/pseud/pseud_2:
dev device power subsystem uevent
```

Вывод команды `ls /sys/class/pseud/pseud_0/device`:

```
address driver driver_override modalias power pseud
subsystem uevent value
```

Вывод команды `ls /sys/devices/platform/pseud.*`:

```
/sys/devices/platform/pseud.0:
address driver driver_override modalias power subsystem
uevent value

/sys/devices/platform/pseud.1:
address driver driver_override modalias power subsystem
uevent value

/sys/devices/platform/pseud.2:
address driver driver_override modalias power subsystem
uevent value
```

Чтобы выгрузить модуль надо выполнить:

```
$ sudo rmmod pseud
```

Вывод команды `sudo dmesg`:

```
[12785.474032] pseud: Exit  
[12785.474218] pseud pseud.2: removed  
[12785.475501] pseud pseud.1: removed  
[12785.476303] pseud pseud.0: removed  
[12785.476363] platform pseud.0: released  
[12785.476385] platform pseud.1: released  
[12785.476398] platform pseud.2: released
```

Команды `lsmod | grep pseud`, `ls /dev/pseud*` и `ls /sys/class/pseud/` теперь не должны давать результата.

# Интерфейс драйвера

---

Для доступа к области данных устройства драйвер предоставляет символьное устройство и атрибуты в файловой системе sysfs.

## Символьное устройство

Драйвер создаёт символьное устройство вида `/dev/pseud_{id}`, где `id` - индекс с нумерацией, начинающейся с нуля. Чтение и запись памяти абстрактного устройства осуществляются с помощью следующих файловых операций:

- `read` - чтение из устройства,
- `write` - запись в устройство,
- `mmap` - отображение устройства в память.

Также реализованы операции `open`, `close`, `lseek`.

## Атрибуты в sysfs

В `sysfs` создаются атрибуты `address` и `value`, через которые осуществляется доступ к памяти.

- `address` - значение, представляющее байтовое смещение внутри области памяти устройства.
- `value` - байт (значение `0..255`), находящийся по смещению `address` в области памяти устройства.

Для каждого атрибута реализованы операции `show` и `store`.

# Проверка работы устройств

---

В каждом из следующих разделов приводятся команды для проверки записи и чтения области памяти устройства из пространства пользователя.

## Системные вызовы read/write

```
$ echo "hello, world!" | sudo tee /dev/pseud_0
$ sudo cat /dev/pseud_0
hello, world!
```

Вывод `sudo dmesg`:

```
[13561.913132] pseud_open: pseud_0 (major 243, minor 0)
[13561.913173] pseud_write: pseud_0 (written 14 bytes)
[13561.913179] pseud_release: pseud_0
[13586.450551] pseud_open: pseud_0 (major 243, minor 0)
[13586.450563] pseud_read: pseud_0 (read 4096 bytes)
[13586.450583] pseud_read: pseud_0 (read 0 bytes)
[13586.450591] pseud_release: pseud_0
```

## Системный вызов mmap

Для этой проверки понадобится программа `/test/test_mmap`, которая собирается при вызове `make` или `make test`. В ней производится отображение `/dev/pseud_1`, записывается "hello, world!" в начало области памяти, "goodbye, world!" в конец, и с соответствующими смещениями в области памяти вызывается `printf`.

```
$ sudo ./test/test_mmap
hello, world!
goodbye, world!
$ sudo head -c 5 /dev/pseud_1
hello
$ sudo tail -c 5 /dev/pseud_1
rld!
```

Вывод `sudo dmesg`:

```
[14307.475815] pseud_open: pseud_1 (major 243, minor 1)
[14307.475832] pseud_mmap: pseud_1
[14307.475995] pseud_release: pseud_1
[14319.665780] pseud_open: pseud_1 (major 243, minor 1)
[14319.665791] pseud_read: pseud_1 (read 5 bytes)
[14319.665798] pseud_release: pseud_1
[14324.784483] pseud_open: pseud_1 (major 243, minor 1)
[14324.784496] pseud_llseek: pseud_1 (new pos: 4091)
[14324.784503] pseud_read: pseud_1 (read 5 bytes)
[14324.784505] pseud_read: pseud_1 (read 0 bytes)
[14324.784510] pseud_release: pseud_1
```

## Атрибуты sysfs

```
$ echo "10" | sudo tee /sys/devices/platform/pseud.2/address
$ cat /sys/devices/platform/pseud.2/value
0
$ echo "255" | sudo tee /sys/devices/platform/pseud.2/value
$ cat /sys/devices/platform/pseud.2/value
255
```

## Авторство и лицензия

---

### Автор

Copyright (c) 2022 Доленко Дмитрий <[dolenko.dv@yandex.ru](mailto:dolenko.dv@yandex.ru)>

### Лицензия

Отдельные файлы лицензированы под лицензией MIT. Однако, при связывании модуля с ядром Linux получается модуль ядра Linux, который имеет двойную лицензию MIT/GPLv2 (см. прилагаемый файл LICENSE).