

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”
Факультет ПИиКТ



ОТЧЁТ
По лабораторной работе №2
По предмету: Операционные системы
Вариант: debugfs: net_device, pci_dev

Студент:
Андрейченко Леонид Вадимович
Группа Р33301

Преподаватель:
Барсуков Илья Александрович

Санкт – Петербург

2022

Задание

Разработать комплекс программ на пользовательском уровне и уровне ядра, который собирает информацию на стороне ядра и передает информацию на уровень пользователя, и выводит ее в удобном для чтения человеком виде. Программа на уровне пользователя получает на вход аргумент(ы) командной строки (не адрес!), позволяющие идентифицировать из системных таблиц необходимый путь до целевой структуры, осуществляет передачу на уровень ядра, получает информацию из данной структуры и распечатывает структуру в стандартный вывод. Загружаемый модуль ядра принимает запрос через указанный в задании интерфейс, определяет путь до целевой структуры по переданному запросу и возвращает результат на уровень пользователя.

Интерфейс передачи: debugfs - отладочная файловая система /sys/kernel/debug, передача параметров через запись в файл.

Целевые структуры: pci_dev, net_device

Выполнение

Исходный код: <https://github.com/buffer404/university/tree/master/year3/Operating%20systems/lab2>

Вывод пользовательской программы

```
leonid@leonid-VirtualBox:~/Desktop/test$ ./user
Hello, select the structure type
1: net_device
2: pci_dev
1
The net_device structure was obtained successfully!
{
  name: lo
  irq: 0
  type: 772
  dma: 0
  mtu: 65536
  tx_queue_len: 1000
  watchdog_timeout: 0
  mem_start: 0
}
```

```
leonid@leonid-VirtualBox:~/Desktop/test$ ./user
Hello, select the structure type
1: net_device
2: pci_dev
2
The pci_dev structure was obtained successfully!
{
  device: 4663
  vendor: 32902
  devfn: 0
  subsystem_device: 0
  subsystem_vendor: 0
}
```

Выводы

Во время выполнения данной лабораторной работы я углубился в работу ядра Linux. Написал собственный модуль для ядра (модуль debugfs), рассмотрел структуры, с которыми я работал.