

# УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Операционные системы»

## **Лабораторная работа №2**

*Вариант Procfs: cpu\_timer, syscall\_info*

Студент

*Степанов М. А.*

*P33301*

Преподаватель

*Барсуков И. А.*

Санкт-Петербург, 2021 г.

## Описание задания

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.

Исходный код: <https://github.com/stmikeal/OsiLab2>

## Выполнение

Сначала напишем модуль Procfs для получения структур

(См. custom\_data.c)

Напишем Makefile сразу с запуском модуля:

```
obj-m := custom_data.o
KERNEL_PATH := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)
all:
    make -C $(KERNEL_PATH) M=$(PWD) modules
    sudo insmod ./custom_data.ko
    chmod 777 /proc/custom_data
    make -C $(KERNEL_PATH) M=$(PWD) clean
    rm -f Module.symvers
clean:
    rmmod custom_data
```

Теперь напишем простой скрипт для пользовательского интерфейса

```
import os
def main():
    struct_id = int(input("Choose struct [0 - for cpu_timer, 1 - for syscall_info]: "))
    pid = int(input("Enter pid: "))
    os.system("echo '%d %d' > /proc/custom_data" % (struct_id, pid))
    os.system("cat /proc/custom_data 2>/dev/null")
if __name__ == '__main__':
    main()
```

В моей реализации выводится информация о cpu\_timer:

Есть ли запущенные таймеры, ссылки на сопутствующие структуры для навигации между таймерами.

информация о cpu\_timer:

Стек системных вызовов, текущий системный вызов, указатель на стек вызовов

## Выводы

Во время выполнения лабораторной работы я углубился в работу ядра linux. Написал собственный модуль для ядра (в частности модуль procfs), рассмотрел некоторые сопутствующие задачи структуры ядра.