

**Факультет программной инженерии и компьютерной техники**

**Лабораторная работа №2 по**

**«Операционным Системам»**

**Преподаватель**: Осипов Святослав Владимирович

**Выполнил:** Бейбитхан Халифа

**Группа:** Р33222

**Вариант:** ioctl: page,thread\_struct

**Задание**

Разработать комплекс программ на пользовательском уровне и уровне ярда, 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 между программой пользователя и ядром и целевая структура задается преподавателем. Интерфейс передачи может быть один из следующих:

syscall - интерфейс системных вызовов.

ioctl - передача параметров через управляющий вызов к файлу/устройству.

procfs - файловая система /proc, передача параметров через запись в файл.

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

Целевая структура может быть задана двумя способами:

Именем структуры в заголовочных файлах Linux

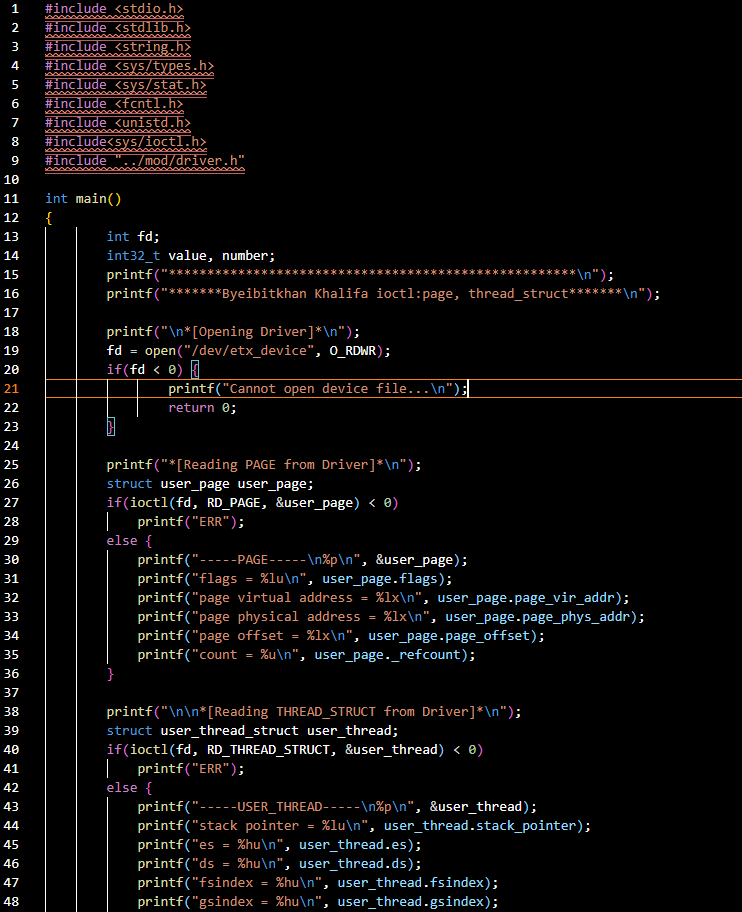
Файлом в каталоге /proc. В этом случае необходимо определить целевую структуру по пути файла в /proc и выводимым данным.

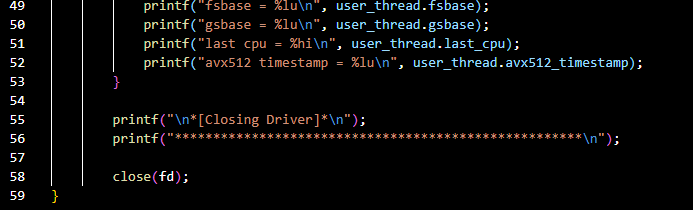
Commands:

Make-Для утилиты make требуется файл Makefile (или makefile), который определяет набор выполняемых задач.

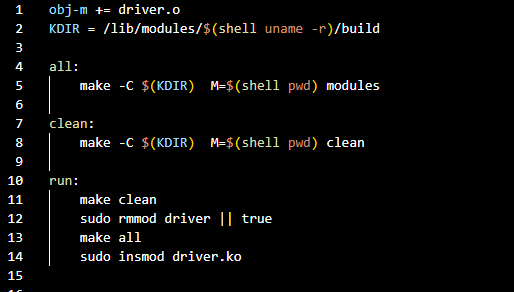
**Выполнение:**

**Main:**

****

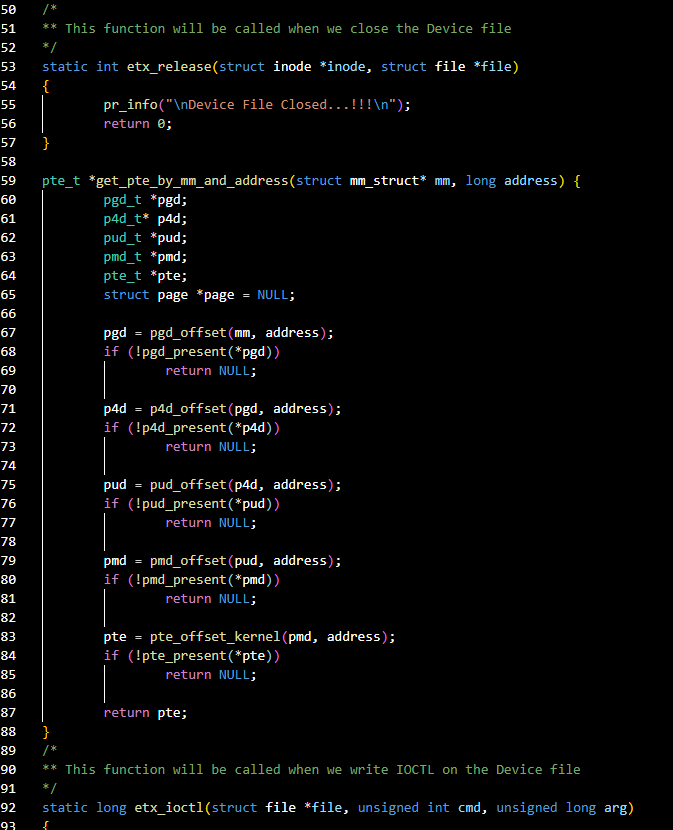
****

**Makefile:**

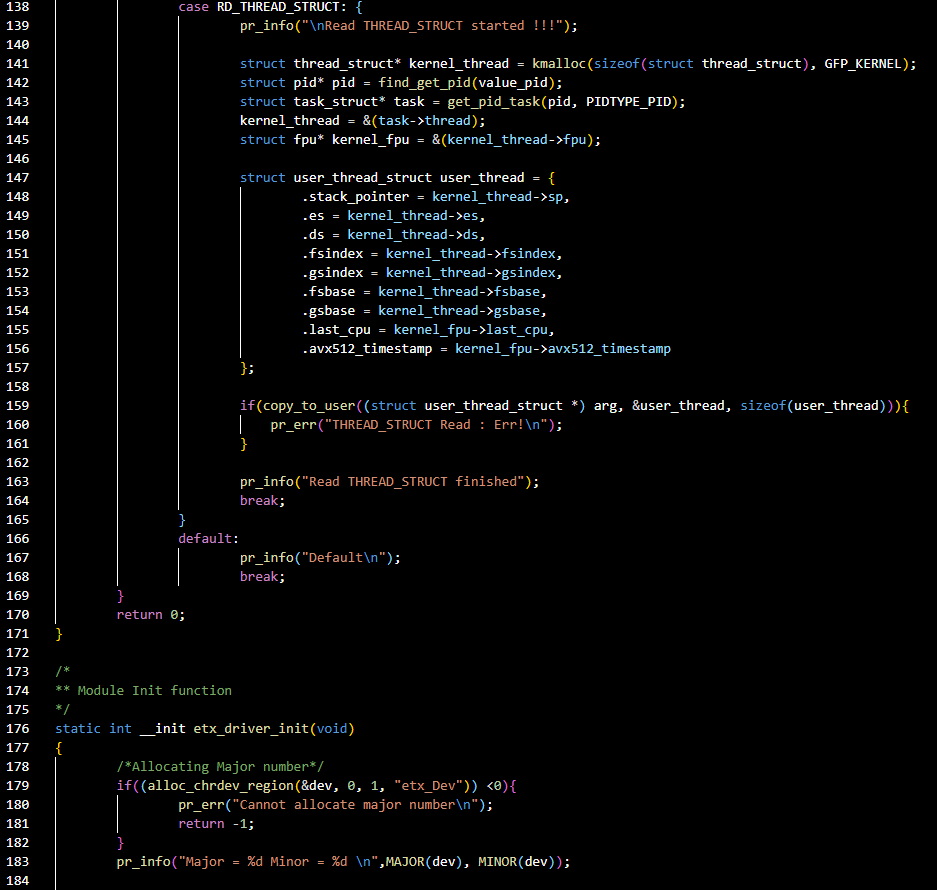
****

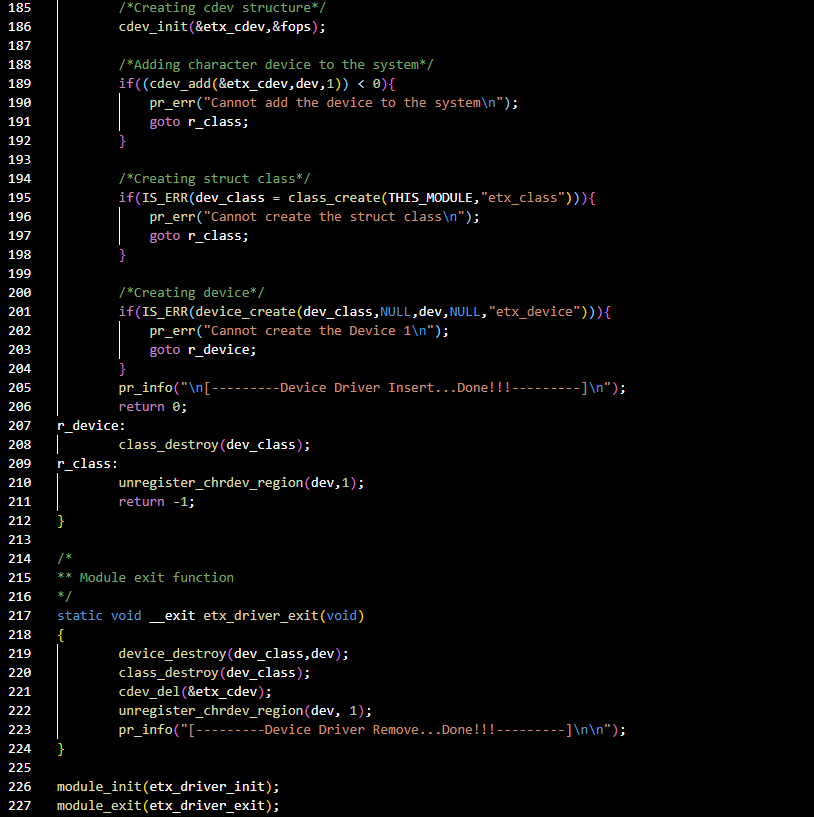
**Driver.c**

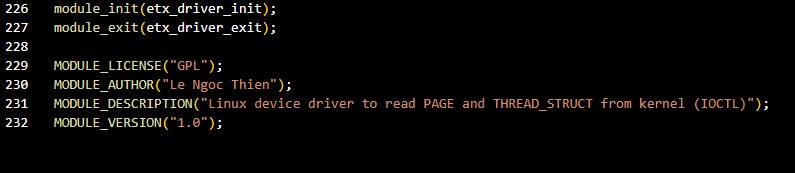
****

****

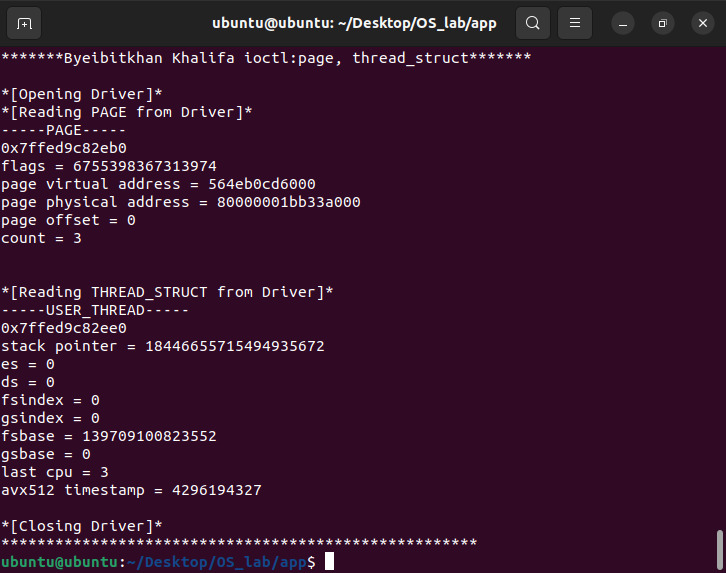
****

****

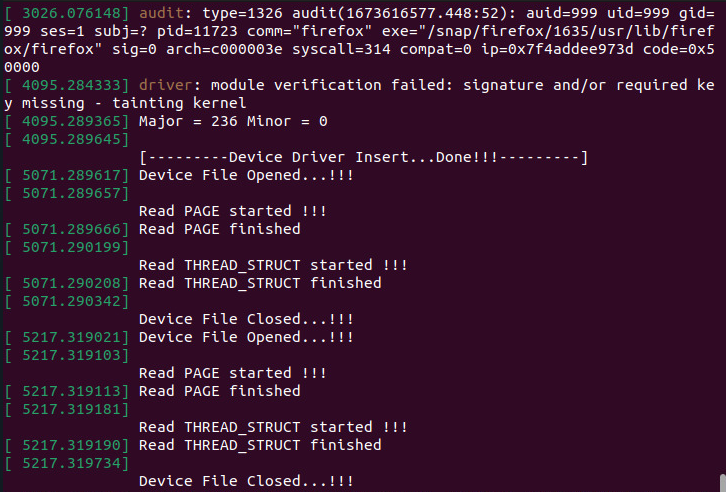
****

****

**Вывод программы:**

****

**Теперь проверьте значение с помощью dmesg: sudo dmesg**

****

**Вывод:**

Во время выполнения лабораторной работы я углубился в работу ядра linux. Работал с интерфейсом передачи (системным вызовом) ioctl, добавил их в ядро и перекомпилировал, а также проверил их на работоспособность.

Чуть-чуть про ioctl:

IOCTL упоминается как управление вводом и выводом, который используется для общения с драйверами устройств. Этот системный вызов доступен в большинстве категорий драйверов. В основном это используется в случае обработки некоторых специфических операций устройства, для которых ядро ​​не имеет системного вызова по умолчанию.

Некоторые приложения ioctl в режиме реального времени: извлечение носителя из дисковода «cd», изменение скорости передачи последовательного порта, регулировка громкости, чтение или запись регистров устройства и т. д. У нас уже есть функция записи и чтения в нашем драйвере устройства. . Но этого недостаточно для всех случаев.