МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №4

з дисципліни «Дослідження та проектування комп’ютерних систем»

Виконав:

студент гр. ІО-91мп

Костяніков В. О.

Перевірив:

Каплунов А. В.

Київ 2020

**Завдання**

Завдання Basic:

1. Продивитися всі три приклади (драйвер там однаковий, різні лише системи збирання).
   1. Обрати один з них для подальшої роботи, зібрати і виконати insmod та rmmod на платі BBB (або емуляторі QEMU).
2. Модифікувати модуль, додавши до нього параметр типу uint, який визначає, скільки разів має бути надрукований рядок “Hello, world!”. Значення параметра за умовчанням 1. Якщо значення параметра 0 або знаходиться між 5 і 10, надрукувати попередження і продовжити роботу. Якщо значення параметра більше 10, то функція ініціалізації повинна надрукувати повідомлення про помилку і повернути значення -EINVAL (модуль не має завантажити взагалі).
3. Додати опис параметра. Подивитися його командою modinfo.
4. Виконати insmod/rmmod модуля на платі BBB без параметра у командному рядку, зі значеннями параметра 0, довільним між 1 і 10, довільним більше 10. Після котрогось із insmod подивитися значення встановленого параметра (каталог /sys/module/hello/parameters)

Завдання Advanced:

1. Розділити проект на два модулі, hello1 та hello2.
   1. Модуль hello1 повинен експортувати функцію print\_hello(), яку використовуватиме модуль hello2 (параметр кількості викликів функції перенести у модуль hello2).
2. Заголовковий файл hello1.h, який використовуватимуть обидва модулі, винести у підкаталог inc, який додати у систему збирання так, щоб файли \*.c могли використовувати директиву #include "hello1.h" лише з іменем файлу, без шляху (див. ccflags-y).
3. Замінити printk на відповідні ситуації pr\_err, pr\_warn, pr\_info (для друку привітання використати pr\_info)
4. Виконати insmod hello1.ko, потім insmod hello2.ko з такими значеннями параметра, щоб отримати всі можливі повідомлення і знайти їх (може знадобитися dmesg, grep).
5. Спробувати завантажити hello2.ko, не завантажуючи hello1.ko, пояснити результат.

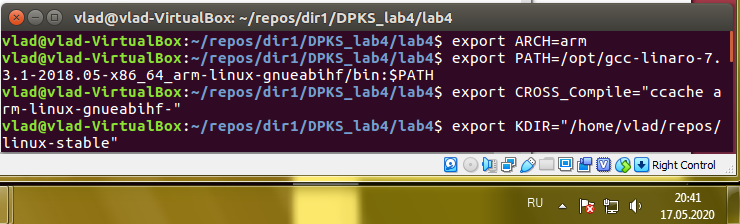
**Результати роботи**

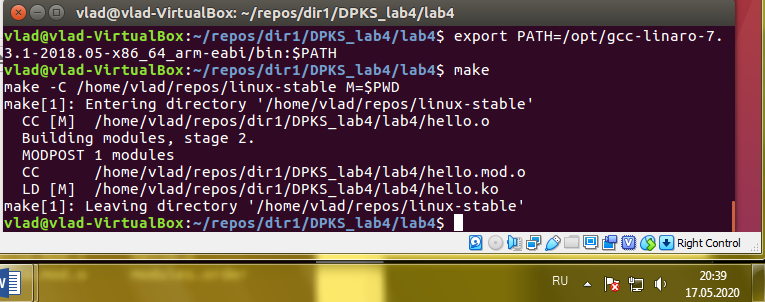
Файли з вихідним кодом, Makefile та усі файли проекту розміщені на GitHub за посиланням:

<https://github.com/Lux-ex-tenebris/DPKS_lab4>

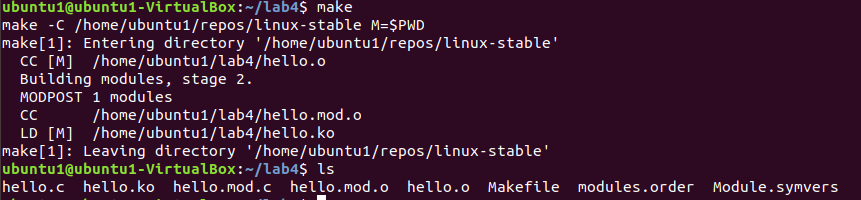
Папка «lab4» містить виконання завдання Basic, «lab4\_advanced» - завдання Advanced.

**Basic**

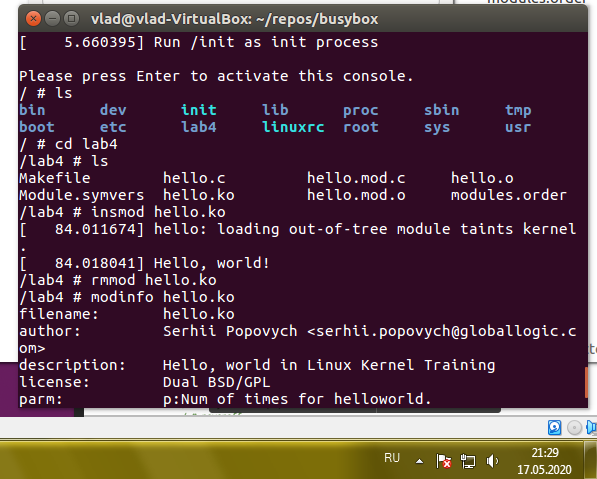




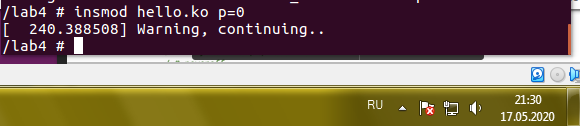
Створення файлів проекту з файлу з вихідним кодом hello.c:



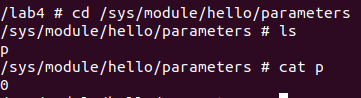
Перевірка insmod hello.ko та modinfo hello.ko:



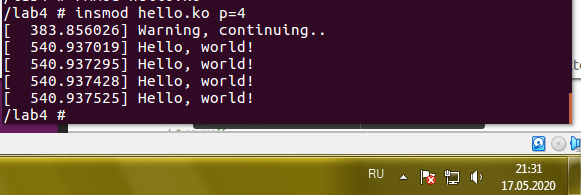
Перевірка модифікованого hello.ko з допомогою insmod (параметр = 0):

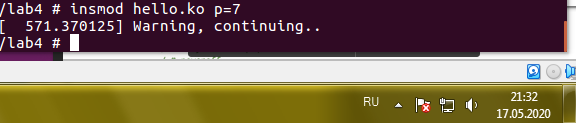


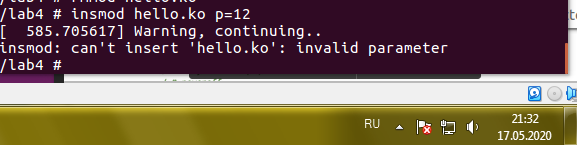
Перевірка значення параметру у директорії:



Перевірка hello.ko з різними параметрами:

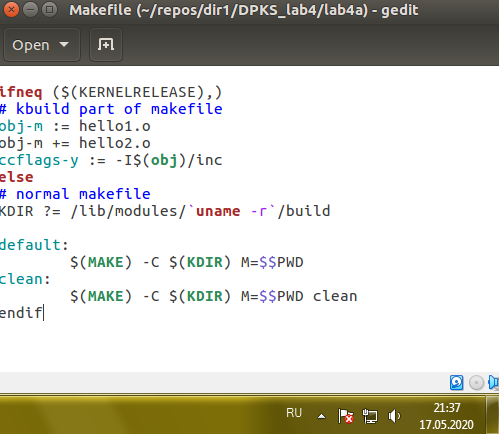




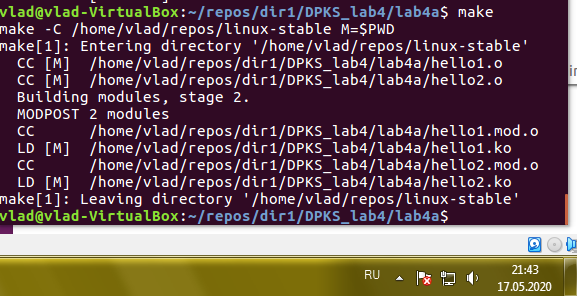


**Advanced**

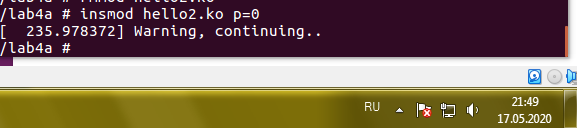
Модифікований Makefile. Додана лінія для ще одного файлу .о, а також зазначення місцезнаходження заголовкового файлу hello1.h.

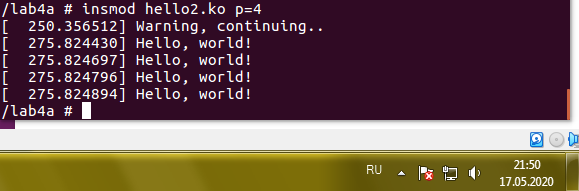


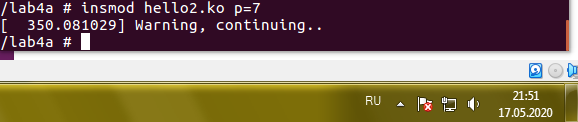
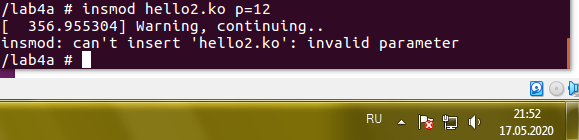
Створення файлів проекту з файлів з вихідним кодом hello1.c та hello2.c та заголовокового файлу hello1.h:



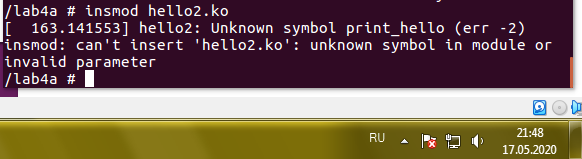
Перевірка роботи функції проекту з різними параметрами:





Перевірка можливості завантаження hello2.ko без завантаження hello1.ko:



Файл не завантажується, оскільки є залежним від hello1.ko, від функції print\_hello, реалізація якої знаходиться в hello1.c, що у hello2.c вказано у цьому рядку:

extern int print\_hello(uint par);