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 - 2»

Виконав:

студент гр. ІО-91мн

Глушенок В.О.

Перевірив:

Каплунов А.В.

Київ 2020

**Завдання**

Завдання Basic:

Продивитися всі три приклади (драйвер там однаковий, різні лише системи збирання).

Обрати один з них для подальшої роботи, зібрати і виконати insmod та rmmod на платі BBB (або емуляторі QEMU).

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).

Додати опис параметра. Подивитися його командою modinfo.

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)

Див. $KDIR/include/linux/moduleparam.h, опис макросів module\_param(), MODULE\_PARM\_DESC().

Завдання Advanced:

Розділити проект на два модулі, hello1 та hello2.

Модуль hello1 повинен експортувати функцію print\_hello(), яку використовуватиме модуль hello2 (параметр кількості викликів функції перенести у модуль hello2).

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).

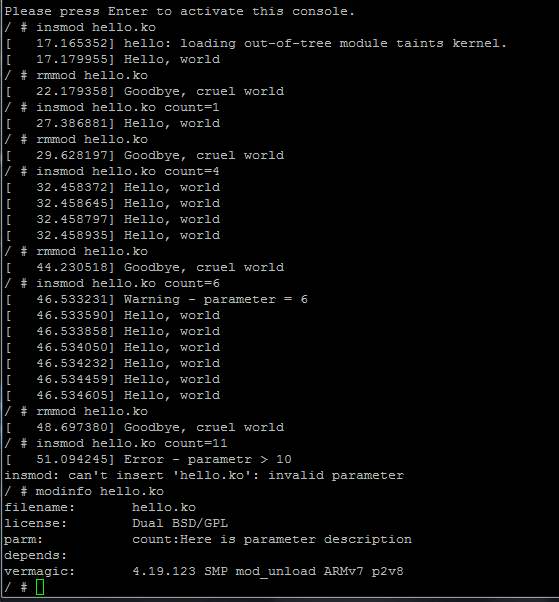
Замінити printk на відповідні ситуації pr\_err, pr\_warn, pr\_info (для друку привітання використати pr\_info)

Виконати insmod hello1.ko, потім insmod hello2.ko з такими значеннями параметра, щоб отримати всі можливі повідомлення і знайти їх (може знадобитися dmesg, grep).

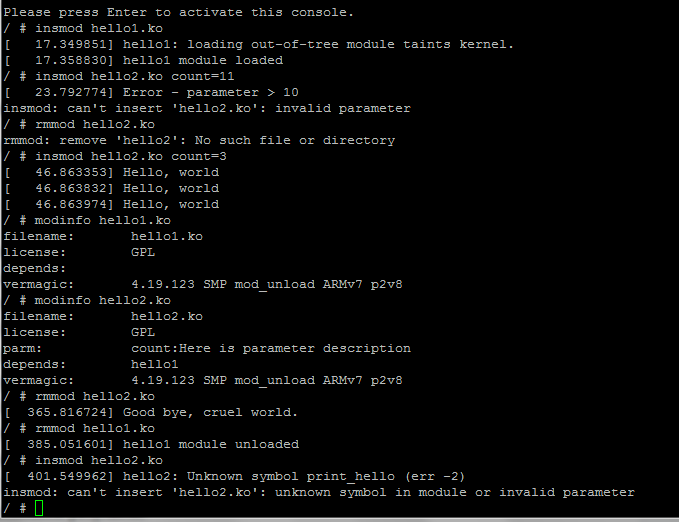
Спробувати завантажити hello2.ko, не завантажуючи hello1.ko, пояснити результат.

**Приклад виконання програми**

Basic:



Advanced:



**Програмний код**

**hello.c**

 #include <linux/init.h>

#include <linux/module.h>

#include <linux/moduleparam.h>

#include <linux/errno.h>

MODULE\_LICENSE("Dual BSD/GPL");

static uint count = 1;

module\_param(count , uint, S\_IRUGO);

MODULE\_PARM\_DESC(count , "Here is parameter description");

static int \_\_init hello\_init(void)

{

if (count > 10)

{

printk(KERN\_ERR "Error - parametr > 10\n");

return -EINVAL;

}

if (!count || (count >= 5 && count <= 10))

printk(KERN\_WARNING "Warning - parameter = %u\n", count );

uint i;

for (i = 0; i < count ; ++i)

printk(KERN\_NOTICE "Hello, world\n");

return 0;

}

module\_init(hello\_init);

static void \_\_exit hello\_exit(void)

{

printk(KERN\_NOTICE "Goodbye, cruel world\n");

}

module\_exit(hello\_exit);

**Makefile**

# If KERNELRELEASE is defined, we've been invoked from the

# kernel build system and can use its language.

ifneq ($(KERNELRELEASE),)

obj-m := hello.o

# Otherwise we were called directly from the command

# line; invoke the kernel build system.

else

KDIR ?= /lib/modules/$(shell uname -r)/build

PWD := $(shell pwd)

default:

$(MAKE) -C $(KDIR) M=$(PWD) modules

Endif

**hello1.c**

#include "hello1.h"

#include <linux/errno.h>

static int \_\_init hello1\_init(void)

{

pr\_info("hello1 module loaded\n");

return 0;

}

module\_init(hello1\_init);

int print\_hello(uint n)

{

if (n > 10)

{

pr\_err("Error - parameter > 10\n");

return -EINVAL;

}

if (!n || (n >= 5 && n <= 10))

pr\_warn("Warning - parameter = %u\n", n);

uint i;

for (i = 0; i < n; ++i)

pr\_info("Hello, world\n");

return 0;

}

EXPORT\_SYMBOL(print\_hello);

static void \_\_exit hello1\_exit(void)

{

pr\_info("hello1 module unloaded\n");

}

module\_exit(hello1\_exit);

**hello2.c**

#include "hello1.h"

#include <linux/moduleparam.h>

static uint count = 1;

module\_param(count , uint, S\_IRUGO);

MODULE\_PARM\_DESC(count , "Here is parameter description");

extern int print\_hello(uint count);

static int \_\_init hello2\_init(void)

{

return print\_hello(count);

}

module\_init(hello2\_init);

static void \_\_exit hello2\_exit(void)

{

pr\_info("Good bye, cruel world.\n");

}

module\_exit(hello2\_exit);

**Makefile**

# If KERNELRELEASE is defined, we've been invoked from the

# kernel build system and can use its language.

ifneq ($(KERNELRELEASE),)

obj-m := hello1.o hello2.o

ccflags-y := -I$(src)/inc

# Otherwise we were called directly from the command

# line; invoke the kernel build system.

else

KDIR ?= /lib/modules/$(shell uname -r)/build

PWD := $(shell pwd)

default:

$(MAKE) -C $(KDIR) M=$(PWD) modules

endif

**Висновки**

Під час виконання лабораторної роботи, було розроблено модуль для лінукса. Протестовано в раніше створеному ядрі лінукса запущеному в емуляторі qemu з різними вхідними параметрами.