Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

Лабораторна робота 5

З дисципліни «Архітектура компʼютерів 2»

ВИКОНАВ:

студент ІІІ курсу ФІОТ

Групи ІО-11

Гончаров П.О.

Київ-2023

**Лабораторна робота 5**

**Завдання:**

I. Розділити проект на два модулі, hello1 та hello2.

A. Модуль hello1 повинен експортувати функцію print\_hello(),

яку використовуватиме модуль hello2 (параметр кількості

викликів функції перенести у модуль hello2).

B. Вся робота зі списком виконується в модулі hello1. Також

додати ще одне поле типу ktime\_t і засікати час до та після

виклику функції друку, а на вивантаженні модуля надрукувати

час, який пішов на кожен друк.

II. 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

&quot;hello1.h&quot; лише з іменем файлу, без шляху (див. ccflags-y).

III. Замінити printk\* на відповідні ситуації pr\_err, pr\_warn, pr\_info (для

друку привітання використати pr\_info)

IV. Виконати insmod hello1.ko, потім insmod hello2.ko з такими

значеннями параметра, щоб отримати всі можливі повідомлення і

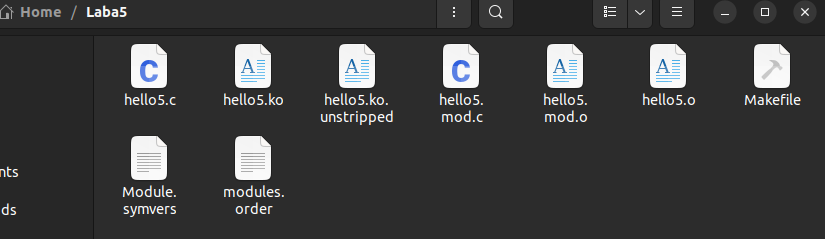
знайти їх (може знадобитися dmesg, grep).

V. Спробувати завантажити hello2.ko, не завантажуючи hello1.ko,

пояснити результат.

**link на гітхаб:**

**Хід роботи**

****

**Hello5.c**

// SPDX-License-Identifier: GPL-2.0+

#include <linux/init.h>

#include <linux/module.h>

#include <linux/moduleparam.h>

#include <linux/printk.h>

#include <linux/types.h>

#include <linux/ktime.h>

#include <linux/slab.h>

#include <linux/list.h>

struct time\_measurement {

struct list\_head list;

ktime\_t start;

ktime\_t end;

};

MODULE\_AUTHOR("IO-11 Goncharov");

MODULE\_DESCRIPTION("Laba 5");

MODULE\_LICENSE("GPL");

static LIST\_HEAD(time\_list);

static uint counter = 1;

module\_param(counter, uint, 0444);

MODULE\_PARM\_DESC(counter, "A counter parameter");

static int \_\_init my\_hello\_init(void)

{

uint idx = 0;

struct time\_measurement \*current\_record;

pr\_info("Counter: %d\n", counter);

BUG\_ON(counter >= 6);

for (idx = 0; idx < counter; idx++) {

current\_record = kmalloc(sizeof(struct time\_measurement), GFP\_KERNEL);

if (idx == 5)

current\_record = NULL;

if (ZERO\_OR\_NULL\_PTR(current\_record))

goto error\_handle;

current\_record->start = ktime\_get();

pr\_info("Hello!\n");

current\_record->end = ktime\_get();

list\_add\_tail(&current\_record->list, &time\_list);

}

return 0;

error\_handle:

{

struct time\_measurement \*temp\_record, \*safe\_record;

pr\_err("Out of memory...\n");

list\_for\_each\_entry\_safe(temp\_record, safe\_record, &time\_list, list) {

list\_del(&temp\_record->list);

kfree(temp\_record);

}

BUG();

return -ENOMEM;

}

}

static void \_\_exit my\_hello\_exit(void)

{

struct time\_measurement \*temp\_record, \*safe\_record;

list\_for\_each\_entry\_safe(temp\_record, safe\_record, &time\_list, list) {

pr\_info("Execution Time: %lld",

temp\_record->end - temp\_record->start);

list\_del(&temp\_record->list);

kfree(temp\_record);

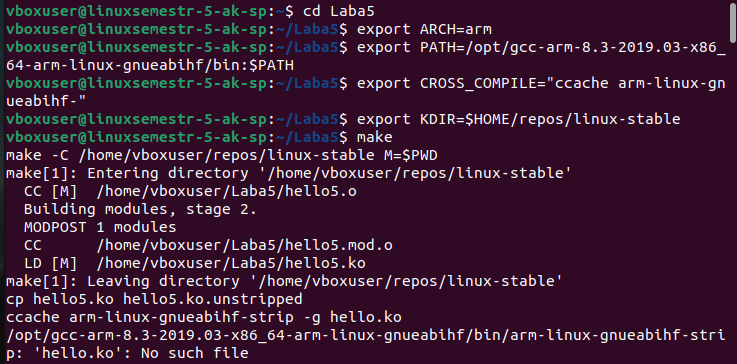
}

BUG\_ON(!list\_empty(&time\_list));

}

module\_init(my\_hello\_init);

module\_exit(my\_hello\_exit);

****

$ cd ~/repos/busybox

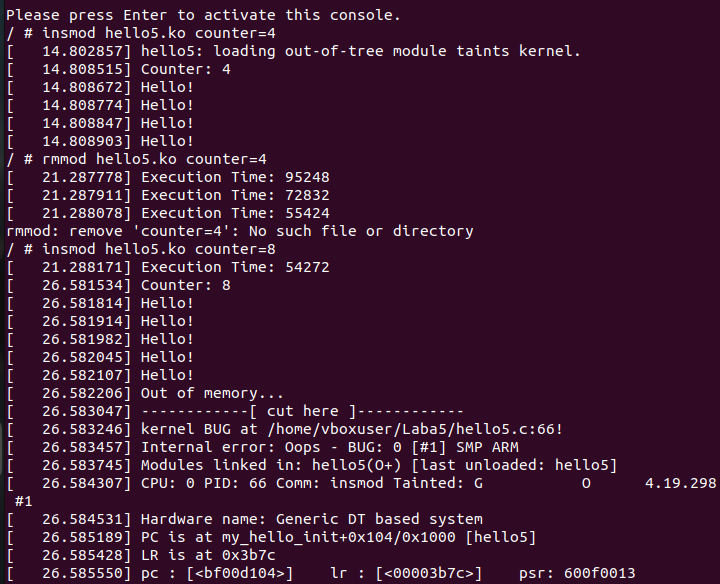
$ cd \_install

$ find . | cpio -o -H newc | gzip > ../rootfs.cpio.gz

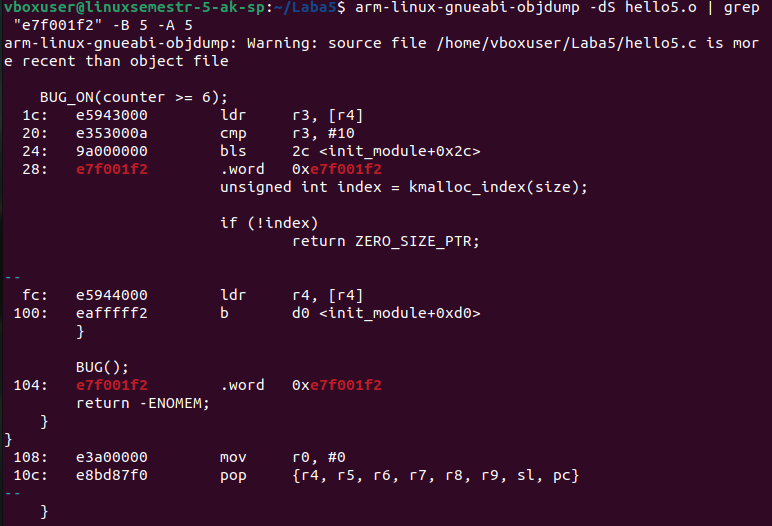
$ cd ..

Після запуску QEMU

Як бачимо програма правильно але так як в коді ми добавили BUG\_ON(counter >= 6); то коли ми визвали програму при counter = 8 вивело помилку

****

****

****