НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №6**

*з дисципліни* ***«****Архітектура комп’ютерів 2****»***

Виконав:

студент 3 курсу

групи ІО-83

ДяченкоВ.Г.

Перевірив:

Нікольський С.С.

Київ 2020 р.

**Хід роботи:**

1. Ознайомився зі скриптом checkpatch.pl .
2. Оголосив структуру даних для розміщення у списку, створив статичну змінну голови списку.
3. Виділив пам'ять для екземпляра оголошеної структури, заніс в неї поточний час ядра.
4. Надрукував час кожної події в наносекундах, вилучив елемент списку і звільнив виділену пам'ять.
5. Модифікував hello1.c згідно поставленого завдання.

**Лістинг програми:**

***hello.c***

#include <linux/init.h>

#include <linux/module.h>

#include <linux/moduleparam.h>

#include <linux/printk.h>

#include <linux/types.h>

#include <linux/slab.h>

#include <linux/ktime.h>

struct personal\_list\_head {

struct personal\_list\_head \*next;

ktime\_t time;

};

MODULE\_AUTHOR("Vlad Diachenko <vlad.diachenko@ukr.net>");

MODULE\_DESCRIPTION("Hello world in Linux Kernel Training");

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

static struct personal\_list\_head \*header;

static uint countOfRepeats = 1;

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

MODULE\_PARM\_DESC(countOfRepeats, "How many times to print hello world");

static int \_\_init initter(void)

{

uint i = 0;

struct personal\_list\_head \*first\_variable, \*second\_variable;

printk(KERN\_INFO "countOfRepeats: %d\n", countOfRepeats);

if(countOfRepeats == 0) {

printk(KERN\_WARNING "Warning: repeats number is 0");

}else if(countOfRepeats >=5 && countOfRepeats <= 10) {

printk(KERN\_WARNING "Warning: repeats number is between 5 and 10");

} else if(countOfRepeats > 10) {

printk(KERN\_ERR "Error: repeats number is greater than 10");

return -EINVAL;

}

header = kmalloc(sizeof(struct personal\_list\_head \*), GFP\_KERNEL);

first\_variable = header;

for(i = 0; i < countOfRepeats; i++){

first\_variable->next = kmalloc(sizeof(struct personal\_list\_head), GFP\_KERNEL);

first\_variable->time = ktime\_get();

pr\_info("Hello world!");

second\_variable = first\_variable;

first\_variable = first\_variable->next;

}

if (countOfRepeats != 0) {

kfree(second\_variable->next);

second\_variable->next = NULL;

}

printk(KERN\_INFO "countOfRepeats: %d\n", countOfRepeats);

return 0;

}

static void \_\_exit exitter(void)

{

struct personal\_list\_head \*variable;

while (header != NULL && countOfRepeats != 0) {

variable = header;

pr\_info("Time : %lld", variable->time);

header = variable->next;

kfree(variable);

}

if (countOfRepeats == 0) {

pr\_info("Time 0 because no printing was used");

}

pr\_info("");

}

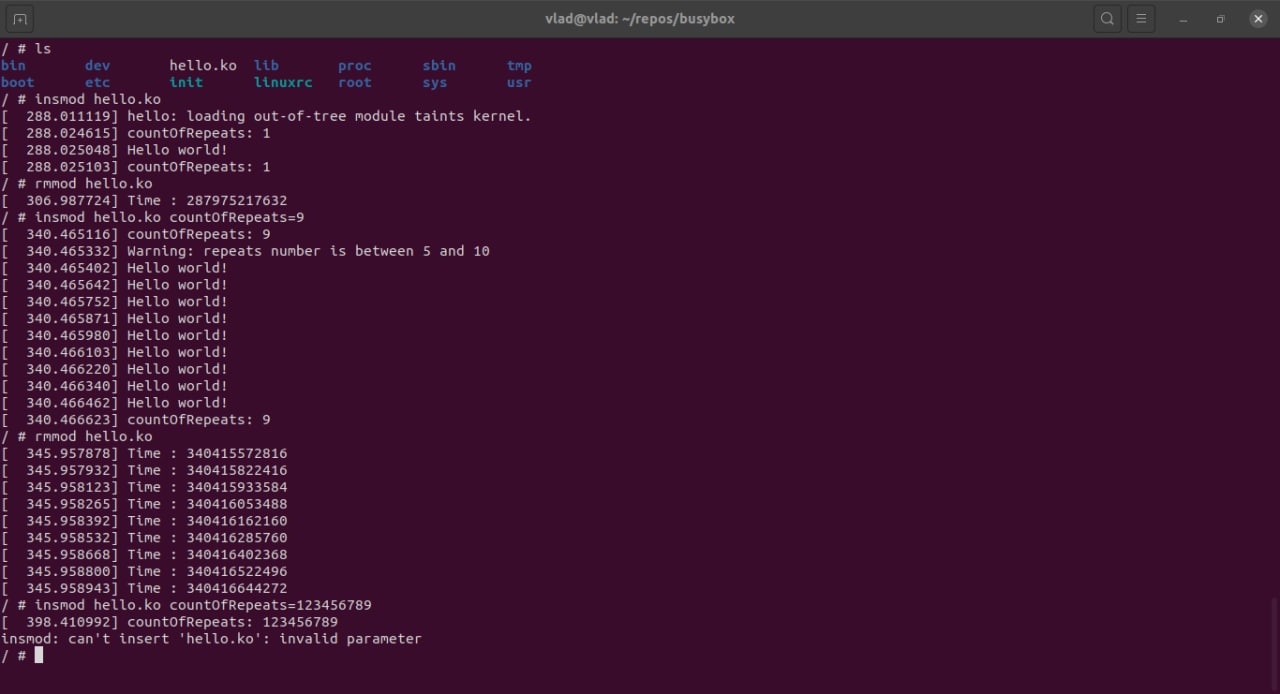
module\_init(initter);

module\_exit(exitter);

**Посилання на репозиторій**

*https://github.com/Ananasik-bet/AK\_2\_Lab6*

**Скріншот виконання, тестування всіх необхідних випадків:**

****