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 №5

з дисципліни «Дослідження та проектування комп’ютерних систем - 2»

Виконав:

студент гр. ІО-91мн

Русінов В.В.

Перевірив:

Каплунов А.В.

Київ 2020

**Завдання**

Завдання Basic:

Оголосити структуру даних для розміщення у списку, яка крім елемента struct list\_head містить поле типу ktime\_t

(include/linux/ktime.h у вашому репозиторії linux-stable).

Створити статичну змінну голови списку.

Перед кожним друком привітання виділити пам'ять для екземпляра оголошеної структури, занести в неї поточний час ядра, отриманий функцією ktime\_get().

У функції hello\_exit() пройти по списку і надрукувати час кожної події в наносекундах, вилучити елемент списку і звільнити виділену пам'ять. Приклад проходу по списку з вилученням елемента є у appendix.

На даному етапі досить виділяти пам'ять викликом

ptr = kmalloc(sizeof(\*ptr), GFP\_KERNEL);

і звільнити її викликом

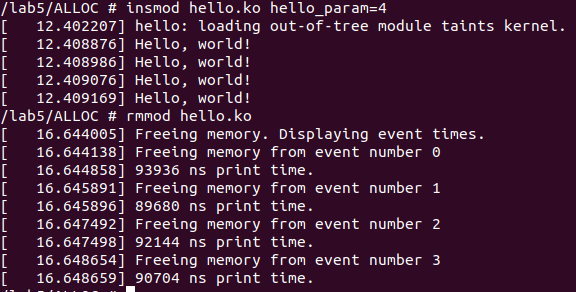
kfree(ptr);

Докладніше у додатку Allocating Memory.

Завдання Advanced:

У цьому завданні 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.

**Приклад виконання програми**



**Програмний код**

**hello.c**

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <linux/types.h>

#include <linux/list.h>

#include <linux/rculist.h>

#include <linux/slab.h>

// MODULE DATA FOR `modinfo`

MODULE\_AUTHOR("Volodymyr Rusinov <beta@zone05.net>");

MODULE\_DESCRIPTION("Module that plays with memory.");

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

// HELLO PARAMETER

static uint hello\_param = 1;

module\_param(hello\_param, uint, S\_IRUGO); // visibility: world, immutable

MODULE\_PARM\_DESC(hello\_param, "An unsigned int, describes how many times \"Hello world\" will be said.");

// STRUCT

static LIST\_HEAD(list\_node\_head);

struct hello\_alloc{

struct list\_head node\_head;

int event\_num;

ktime\_t event\_time;

ktime\_t kill\_time;

};

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

{

if(hello\_param == 0 || (hello\_param >= 5 && hello\_param <= 10))

printk(KERN\_EMERG "WARNING! Invalid argument. Continuing job...");

else if(hello\_param > 10) {

printk(KERN\_EMERG "ERROR! Invalid arguemnt. Exiting job...");

return -EINVAL;

}

else {

// declaring variables here, since code isn't as pretty as it used to be

int i = 0;

ktime\_t start, end;

struct hello\_alloc\* alloc\_list; // if doesn't work, move kmalloc here

for(i = 0; i < hello\_param; i++){

start = ktime\_get();

printk(KERN\_EMERG "Hello, world!\n");

end = ktime\_get();

alloc\_list = (struct hello\_alloc\*) kmalloc(sizeof(struct hello\_alloc), GFP\_KERNEL);

\*alloc\_list = (struct hello\_alloc){

.event\_num = i,

.event\_time = start,

.kill\_time = end

};

list\_add\_tail(&alloc\_list->node\_head, &list\_node\_head);

}

}

return 0;

}

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

{

struct hello\_alloc \*md, \*tmp;

printk(KERN\_EMERG "Freeing memory. Displaying event times.");

list\_for\_each\_entry\_safe(md, tmp, &list\_node\_head, node\_head) {

printk(KERN\_EMERG "Freeing memory from event number %d", md->event\_num);

pr\_info("%lld ns print time.\n", (long long int) (md->kill\_time - md->event\_time));

list\_del(&md->node\_head);

kfree(md);

}

BUG\_ON(!list\_empty(&list\_node\_head));

printk(KERN\_EMERG "Done. Unloading the module...");

}

module\_init(hello\_init);

module\_exit(hello\_exit);

**Висновки**

Під час виконання лабораторної роботи, було розроблено модуль, який виконує функцію виділення пам’яті, для структури даних, яка містить інформацію про час виводу у консоль запису. Було здобуто навички із створення спискових структур даних для мови С для написання модулів ядра Linux.