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 ІВ-93

Кочерга Андрій

Перевірив:

Нікольський С.С.

Київ 2021 р.

**Лістинг програми:**

**Hello.c**

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <linux/ktime.h>

#include <linux/slab.h>

#include <linux/list.h>

struct list\_item {

ktime\_t time;

struct list\_head list\_node;

};

static struct list\_head my\_list\_head = LIST\_HEAD\_INIT(my\_list\_head);

MODULE\_AUTHOR("Kocherga Andry");

MODULE\_DESCRIPTION("Hello world Lab6");

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

static uint amount = 1;

module\_param(amount, uint, 0444);

MODULE\_PARM\_DESC(amount, "Amount of outputting Hello worlds");

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

{

uint i = 0;

struct list\_item \*tail;

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

if (amount == 0 || (amount >= 5 && amount <= 10))

pr\_warn("WARNING\nThe parameter is between 5 and 10 or 0");

BUG\_ON(amount > 10);

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

{

tail = kmalloc(sizeof(struct list\_item), GFP\_KERNEL);

if (i == 5)

tail = 0;

if(ZERO\_OR\_NULL\_PTR(tail))

goto error;

tail->time = ktime\_get();

list\_add\_tail(&(tail->list\_node), &my\_list\_head);

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

}

return 0;

error:

{

struct list\_item \*md, \*tmp;

pr\_err("kmalloc out of memory");

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

list\_del(&md->list\_node);

kfree(md);

}

BUG();

return -ENOMEM;

}

}

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

{

struct list\_item \*md, \*tmp;

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

pr\_info("time: %lld\n", md->time);

list\_del(&md->list\_node);

kfree(md);

}

BUG\_ON(!list\_empty(&my\_list\_head));

}

module\_init(hello\_init);

module\_exit(hello\_exit);

**Makefile**

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <linux/ktime.h>

#include <linux/slab.h>

#include <linux/list.h>

struct list\_item {

ktime\_t time;

struct list\_head list\_node;

};

static struct list\_head my\_list\_head = LIST\_HEAD\_INIT(my\_list\_head);

MODULE\_AUTHOR("Kocherga Andry");

MODULE\_DESCRIPTION("Hello world Lab6");

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

static uint amount = 1;

module\_param(amount, uint, 0444);

MODULE\_PARM\_DESC(amount, "Amount of outputting Hello worlds");

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

{

uint i = 0;

struct list\_item \*tail;

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

if (amount == 0 || (amount >= 5 && amount <= 10))

pr\_warn("WARNING\nThe parameter is between 5 and 10 or 0");

BUG\_ON(amount > 10);

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

{

tail = kmalloc(sizeof(struct list\_item), GFP\_KERNEL);

if (i == 5)

tail = 0;

if(ZERO\_OR\_NULL\_PTR(tail))

goto error;

tail->time = ktime\_get();

list\_add\_tail(&(tail->list\_node), &my\_list\_head);

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

}

return 0;

error:

{

struct list\_item \*md, \*tmp;

pr\_err("kmalloc out of memory");

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

list\_del(&md->list\_node);

kfree(md);

}

BUG();

return -ENOMEM;

}

}

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

{

struct list\_item \*md, \*tmp;

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

pr\_info("time: %lld\n", md->time);

list\_del(&md->list\_node);

kfree(md);

}

BUG\_ON(!list\_empty(&my\_list\_head));

}

module\_init(hello\_init);

module\_exit(hello\_exit);

**Скріншот виконання:**

