НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №7**

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

Виконав:

студент 3 курсу

групи ІВ-81

Юхимчук Я. М.

Перевірив:

Каплунов А. В.

Київ 2020 р.

**Посилання на репозиторій:** https://github.com/darkness8129/ak-2-labs/tree/master/lab7

**Лістинг коду:**

/\*

\* Copyright (c) 2017, GlobalLogic Ukraine LLC

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* This product includes software developed by the GlobalLogic.

\* 4. Neither the name of the GlobalLogic nor the

\* names of its contributors may be used to endorse or promote products

\* derived from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY GLOBALLOGIC UKRAINE LLC ``AS IS`` AND ANY

\* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

\* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

\* DISCLAIMED. IN NO EVENT SHALL GLOBALLOGIC UKRAINE LLC BE LIABLE FOR ANY

\* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

\* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND

\* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

\* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

\* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

#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"

#include "asm-generic/bug.h"

MODULE\_AUTHOR("Yaroslav Yukhymchuk IV-81");

MODULE\_DESCRIPTION("AK-2 Lab 5");

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

static uint number = 1;

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

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

struct own\_list\_head {

struct own\_list\_head \*next;

ktime\_t time;

};

static struct own\_list\_head \*head;

static int \_\_init initter(void)

{

uint i = 0;

struct own\_list\_head \*p\_elem;

struct own\_list\_head \*c\_elem;

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

if (number == 0) {

printk(KERN\_WARNING "Parameter is 0\n");

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

printk(KERN\_WARNING "Parameter is between 5 and 10\n");

} else if (number > 10) {

BUG\_ON(number > 10);

}

if (number > 0) {

head = kmalloc(

sizeof(struct own\_list\_head \*),

GFP\_KERNEL

);

head->next = NULL;

head->time = ktime\_get();

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

}

c\_elem = head;

for (i = 1; i < number; i++) {

p\_elem = c\_elem;

c\_elem = kmalloc(

sizeof(struct own\_list\_head \*),

GFP\_KERNEL

);

if (i == 5) {

c\_elem = NULL;

}

c\_elem->next = NULL;

c\_elem->time = ktime\_get();

p\_elem->next = c\_elem;

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

}

return 0;

}

static void \_\_exit exitter(void)

{

struct own\_list\_head \*c1 = head;

struct own\_list\_head \*c2;

while (c1 != NULL) {

c2 = c1;

pr\_info("Time: %lld\n", c1->time);

c1 = c2->next;

kfree(c2);

}

}

module\_init(initter);

module\_exit(exitter);

**Результати:**



