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-82

Кузьмич А.А.

Перевірив:

Нікольський С.С.

Київ 2020 р.

Лістинг **програми:**

hello1**.c**

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <hello1.h>

#include <linux/slab.h>

#define DEBUG

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

MODULE\_DESCRIPTION("AK-1 Lab7");

MODULE\_AUTHOR("Kuzmich IV-82");

EXPORT\_SYMBOL(print\_hello);

static struct my\_list\_head \*head;

void clear\_my\_list(void)

{

struct my\_list\_head\* temp\_first;

struct my\_list\_head\* temp\_second;

temp\_first = head;

while(temp\_first != NULL) {

temp\_second = temp\_first->next;

kfree(temp\_first);

temp\_first = temp\_second;

}

}

int print\_hello(uint count)

{

int i;

struct my\_list\_head \*temp\_head1;

struct my\_list\_head \*temp\_head2;

BUG\_ON(count > 10);

head = kmalloc(sizeof(struct my\_list\_head\*), GFP\_KERNEL);

temp\_head1 = head;

if (count == 0)

{

pr\_warn("WARNING\nvalue = 0\n");

return 0;

}

else if (count >= 5)// && count <= 10)

{

pr\_warn("WARNING\nvalue is in betwen 5 and 10\n");

}

// else if (count > 10)

// {

// pr\_err("ERROR\n value > 10");

// return -EINVAL;

// }

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

{

temp\_head1->next = kmalloc(sizeof(struct my\_list\_head\*), GFP\_KERNEL);

if (i == 9){

temp\_head1 = NULL;

}

// BUG\_ON(temp\_head1 == NULL);

// if (!temp\_head1) {

// clear\_my\_list();

// pr\_err("ERROR\n out of memory");

// return -ENOMEM;

// }

temp\_head1->time = ktime\_get();

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

temp\_head1->post\_time = ktime\_get();

temp\_head2 = temp\_head1;

temp\_head1 = temp\_head1->next;

}

kfree(temp\_head2->next);

temp\_head2->next = NULL;

//pr\_info("%d", sizeof(struct my\_list\_head) \* count / 1024 / 1024);

return 0;

}

static int \_\_init hello1\_init(void)

{

pr\_info("hello1 init");

return 0;

}

static void \_\_exit hello1\_exit(void)

{

struct my\_list\_head\* temp\_first = head;

struct my\_list\_head\* temp\_second = temp\_first;

pr\_debug("Print list start");

while(temp\_first != NULL)

{

pr\_debug("Time spent %lld", temp\_first->post\_time - temp\_first->time);

temp\_second = temp\_first;

temp\_first=temp\_second->next;

kfree(temp\_second);

}

pr\_debug("Print list end");

pr\_info("hello1 exit");

}

module\_init(hello1\_init);

module\_exit(hello1\_exit);

**hello2.c**

#include <linux/init.h>

#include <linux/module.h>

//#include <linux/types.h>

#include <hello1.h>

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

MODULE\_DESCRIPTION("AK-1 Lab7");

MODULE\_AUTHOR("Kuzmich IV-82");

static uint count = 1;

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

static int hello\_init(void)

{

return print\_hello(count);

}

static void hello\_exit(void)

{

printk(KERN\_ALERT "Goodbye, cruel world\n");

}

module\_init(hello\_init);

module\_exit(hello\_exit);

**Makefile**

ccflags-y := -I$(PWD)/inc

ifneq ($(KERNELRELEASE),)

# kbuild part of makefile

obj-m := hello1.o hello2.o

ccflags-y += -g -DDEBUG

else

# normal makefile

KDIR ?= /lib/modules/`uname -r`/build

default:

$(MAKE) -C $(KDIR) M=$$PWD

cp hello1.ko hello1.ko.unstripped

cp hello2.ko hello2.ko.unstripped

$(CROSS\_COMPILE)strip -g hello1.ko

$(CROSS\_COMPILE)strip -g hello2.ko

clean:

$(MAKE) -C $(KDIR) M=$$PWD clean

%.s %.i: %.c

$(MAKE) -C $(KDIR) M=$$PWD $@

endif

**hello1.h**

#include <linux/ktime.h>

struct my\_list\_head {

struct my\_list\_head \*next;

ktime\_t time;

ktime\_t post\_time;

};

int print\_hello(uint count);