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-82

Старшинов К.Ю.

Перевірив:

Нікольський С. С.

Київ 2020 р.

**Репозиторій:** https://github.com/Born2drum/AK2\_Lab6\_Advanced

**Лістинг програми:**

**hello1.c**

// SPDX-License-Identifier: GPL-2-Clause

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <linux/types.h>

#include <linux/slab.h>

#include <linux/ktime.h>

#include <hello1.h>

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

MODULE\_AUTHOR("Starshynov IV-82\n");

MODULE\_DESCRIPTION("AK-2 lab\_6 (advanced task): hello1.c\n");

struct timeit\_list {

struct list\_head node;

ktime\_t before;

ktime\_t after;

};

static struct list\_head head\_node = LIST\_HEAD\_INIT(head\_node);

int print\_hello(uint n)

{

struct timeit\_list \*list, \*tmp;

uint i;

if (n < 0) {

pr\_err("ERROR! n < 0\n");

} else if (n == 0) {

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

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

pr\_warn("WARNING! 5 <= n <= 10\n");

} else if (n > 10) {

pr\_err("ERROR! n > 10\n");

return -EINVAL;

}

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

list = kmalloc(sizeof(struct timeit\_list), GFP\_KERNEL);

if(ZERO\_OR\_NULL\_PTR(list)) goto clean\_up;

list->before = ktime\_get();

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

list->after = ktime\_get();

list\_add\_tail(&list->node, &head\_node);

}

return 0;

clean\_up:

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

list\_del(&list->node);

kfree(list);

}

pr\_err("ERROR! Memory is out\n");

return -1;

}

EXPORT\_SYMBOL(print\_hello);

static int \_\_init init\_hello(void)

{

pr\_info("hello1 init\n");

return 0;

}

static void \_\_exit exit\_hello(void)

{

struct timeit\_list \*list, \*tmp;

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

pr\_info("Time: %lld", list->after - list->before);

list\_del(&list->node);

kfree(list);

}

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

pr\_info("hello1 exit\n");

}

module\_init(init\_hello);

module\_exit(exit\_hello);

**hello2.c**

// SPDX-License-Identifier: GPL-2-Clause

#include <linux/init.h>

#include <linux/module.h>

#include <linux/printk.h>

#include <linux/types.h>

#include <linux/slab.h>

#include <linux/ktime.h>

#include <hello1.h>

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

MODULE\_DESCRIPTION("AK-2 lab\_6 (advanced task): hello2.c\n");

MODULE\_AUTHOR("Starshynov IV-82\n");

static uint n = 1;

module\_param(n, uint, 0);

MODULE\_PARM\_DESC(n, "How many hellos to print\n");

static int \_\_init init\_hello(void)

{

pr\_info("hello2 init\n");

print\_hello(n);

return 0;

}

static void \_\_exit exit\_hello(void)

{

pr\_info("hello2 exit\n");

}

module\_init(init\_hello);

module\_exit(exit\_hello);

**Makefile**

# add files from /inc to path

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

ifneq ($(KERNELRELEASE),)

# kbuild part of makefile

obj-m := hello1.o hello2.o

else

# normal makefile

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

default:

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

clean:

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

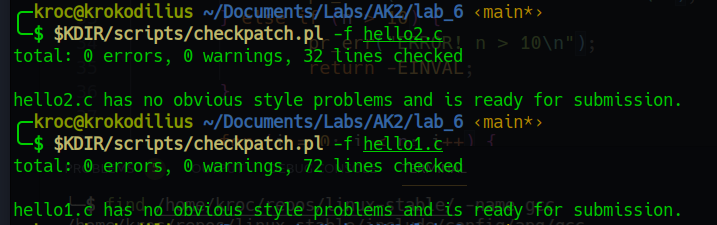
endif

**inc/hello1.h**

#include <linux/types.h>

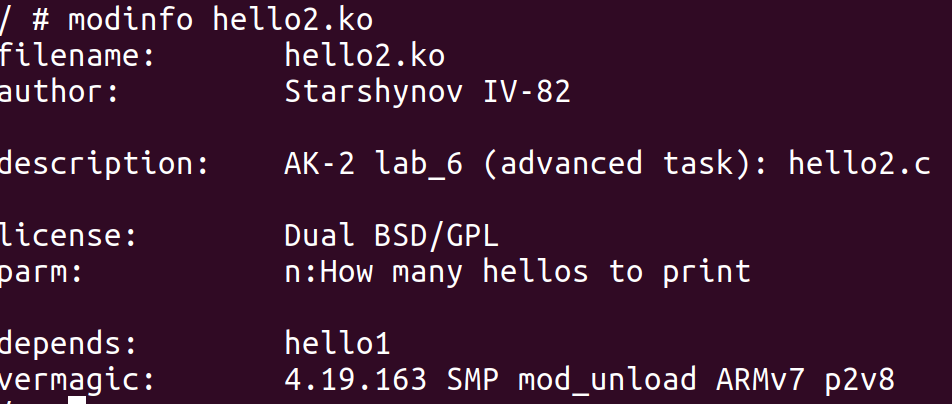
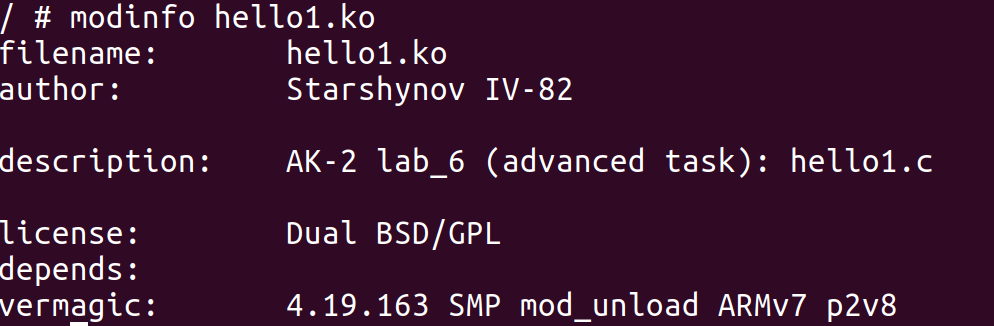
int print\_hello(uint n);

**Відповідність Linux Kernel Coding Style:**



**Скріншоти виконання:**

1. *modinfo*



*2. insmod / rmmod з різними параметрами*

