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 ІО-82

Мартинюк Н.О.

Перевірив:

Каплунов А. В.

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> |
|  |

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

|  |
| --- |
| MODULE\_DESCRIPTION("AK\_Lab7"); |
|  |

|  |
| --- |
| MODULE\_AUTHOR("Nazar Martyniuk IO-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; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| static int print\_hello(uint count) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| int i; |
|  |

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

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

|  |
| --- |
|  |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| temp\_head1 = head; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BUG\_ON(count > 10); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (count == 0) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| pr\_warn("c = 0"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else if (count >= 5 && count <= 10) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| pr\_warn("5 < c < 10"); |
|  |

|  |
| --- |
| } |
|  |

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

|  |
| --- |
| { |
|  |

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

|  |
| --- |
| if (i == 11) |
|  |

|  |
| --- |
| temp\_head1 = NULL; |
|  |

|  |
| --- |
| if (ZERO\_OR\_NULL\_PTR(temp\_head1)) |
|  |

|  |
| --- |
| goto clear; |
|  |

|  |
| --- |
| 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; |
|  |

|  |
| --- |
| return 0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| clear: |
|  |

|  |
| --- |
| pr\_err("No memory left\n"); |
|  |

|  |
| --- |
| clear\_my\_list(); |
|  |

|  |
| --- |
| return -ENOMEM; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

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

|  |
| --- |
| { |
|  |

|  |
| --- |
| pr\_info("Hello1 started"); |
|  |

|  |
| --- |
| pr\_info(""); |
|  |

|  |
| --- |
| 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("------------Start------------"); |
|  |

|  |
| --- |
| while(temp\_first != NULL) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| pr\_debug("Working time: %lld", temp\_first->post\_time - temp\_first->time); |
|  |

|  |
| --- |
| temp\_second = temp\_first; |
|  |

|  |
| --- |
| temp\_first=temp\_second->next; |
|  |

|  |
| --- |
| kfree(temp\_second); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| pr\_debug("------------End------------"); |
|  |

|  |
| --- |
| pr\_info("Hello1 finished"); |
|  |

|  |
| --- |
| pr\_info(""); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| module\_init(hello1\_init); |
|  |

module\_exit(hello1\_exit);

**hello2.c**

|  |
| --- |
| #include <linux/init.h> |
|  |

|  |
| --- |
| #include <linux/module.h> |
|  |

|  |
| --- |
| #include <hello1.h> |
|  |

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

|  |
| --- |
| MODULE\_DESCRIPTION("AK\_Lab7"); |
|  |

|  |
| --- |
| MODULE\_AUTHOR("Nazar Martyniuk IO-82"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| static uint count = 1; |
|  |

|  |
| --- |
| module\_param(count, uint, 0); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| static int hello\_init(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| return print\_hello(count); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| static void hello\_exit(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| printk(KERN\_ALERT "Module removed\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| module\_init(hello\_init); |
|  |

module\_exit(hello\_exit);

**hello1.h**

|  |
| --- |
| #include <linux/ktime.h> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| struct my\_list\_head { |
|  |

|  |
| --- |
| struct my\_list\_head \*next; |
|  |

|  |
| --- |
| ktime\_t time; |
|  |

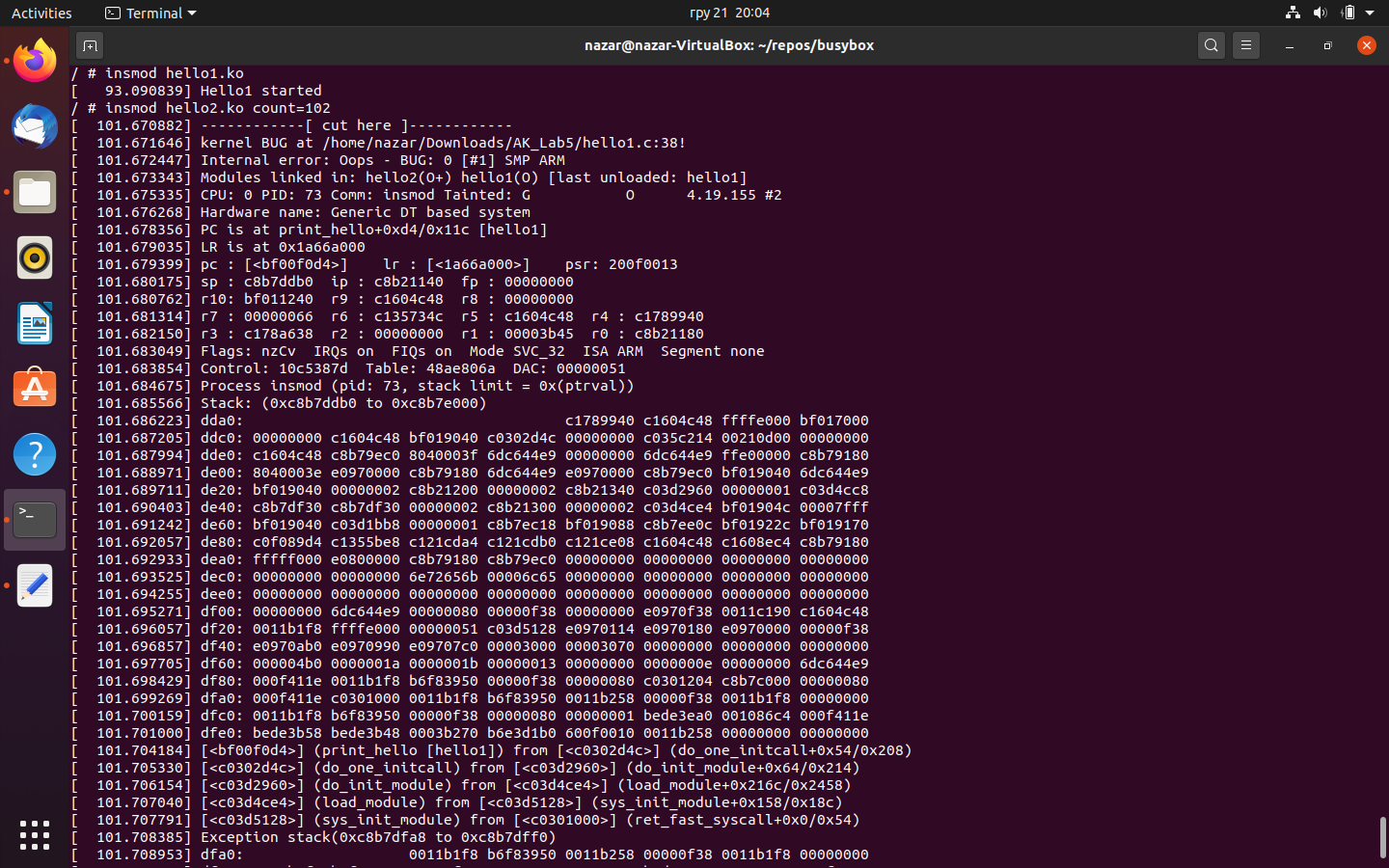
|  |
| --- |
| ktime\_t post\_time; |
|  |

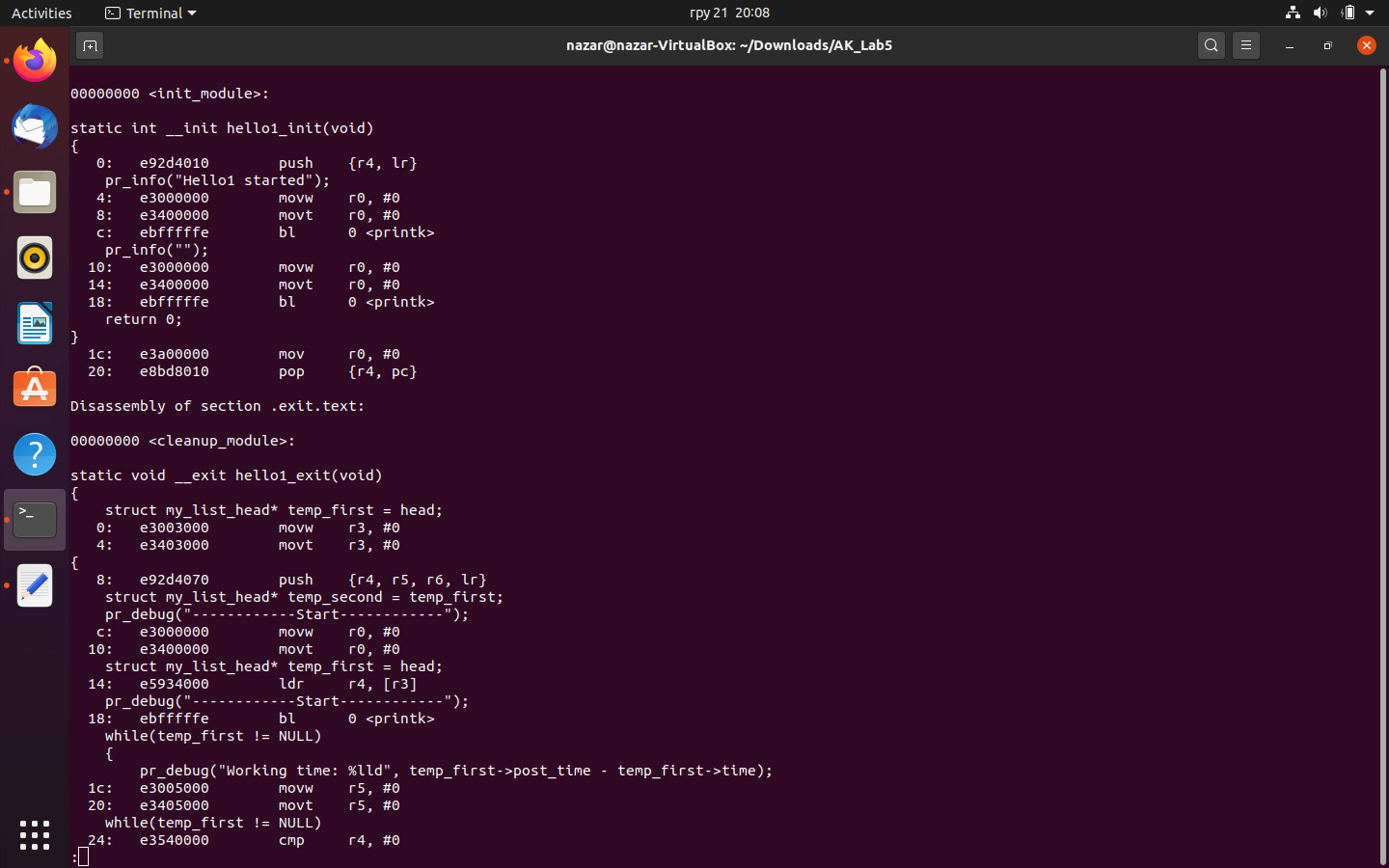
|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

static int print\_hello(uint count);

**Скріншоти виконання:**

****

****