МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ЗВІТ
ПРО ЛАБОРАТОРНУ РОБОТУ №5

Виконав:
Студент групи ІО-11
Гук Д. С.

Перевірив:
Гайдай А. Р.

Київ 2023

Історія виконаних команд:

```
cd
cd AK/Lab5/
vi hello3.c
vi Makefile
// встановлення змінних середовища, які використовуються при збиранні ядра Linux для архітектури ARM
export KDIR=$HOME/repos/linux-stable
export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:$PATH
export CROSS_COMPILE='ccache arm-eabi-'
export ARCH=arm
make
cp hello3.ko ~/repos/busybox/_install/hello1.ko       // Додаємо hello3.ko до директорії BBB
cd
cd repos/busybox/_install
find . | cpio -o -H newc | gzip > ../rootfs.cpio.gz       // Перезбираємо BBB
cd ..
qemu-system-arm -kernel _install/boot/zImage -initrd rootfs.cpio.gz \
-machine virt -nographic -m 512 \
--append "root=/dev/ram0 rw console=ttyAMA0,115200 mem=512M"
ls
insmod hello3.ko myParam=8
rmmod hello3.ko
insmod hello3.ko myParam=11
${CROSS_COMPILE}objdump -dS hello3.o | grep e7f001f2
```

Вихідний код hello3.c:

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/printk.h>
#include <linux/list.h>
#include <linux/ktime.h>
#include <linux/slab.h>

MODULE_AUTHOR("Huk Dmytro");
MODULE_DESCRIPTION("Lab5 Module hello.c");
MODULE_LICENSE("Dual BSD/GPL");

static uint myParam = 1;

// 0444 = S_IRUGO - флаг дозволу на читання параметра
module_param(myParam, uint, 0444);
MODULE_PARM_DESC(myParam, "My description");

struct myStruct {
    struct list_head list;
    ktime_t myTime;
};

// статична зміна голови списку
static LIST_HEAD(myList);
int counter;

static void freeMemory(int message) {
    struct myStruct *ptr, *next;
    list_for_each_entry_safe(ptr, next, &myList, list) {
        if (message == 1) {
            pr_emerg("Time in nanoseconds: %lld\n", ktime_to_ns(ptr->myTime));
        }
        list_del(&ptr->list);
        kfree(ptr);
    }
}

static int __init hello_init(void)
{
    BUG_ON(myParam > 10);

    if (myParam == 0 || (myParam <= 10 && myParam >= 5)) {
        pr_emerg("Warning: %u\n", myParam);
    }

    counter = 0;
    while (counter < myParam) {
        struct myStruct *ptr = kmalloc(sizeof(*ptr), GFP_KERNEL);
        // Примусово встановлення ptr на 0 для 9-го елементу
        if (counter == 9) {
            ptr = NULL;
        }
        if (!ptr) {
            pr_err("kmalloc() returned 0\n");
            freeMemory(0);
```

```
                BUG();
        }
        ptr->myTime = ktime_get();
        list_add_tail(&ptr->list, &myList);
        pr_emerg("Hello world!\n");
        counter += 1;
    }

    return 0;
}

static void __exit hello_exit(void)
{
    freeMemory(1);
}

module_init(hello_init);
module_exit(hello_exit);
```

Вихідний код Makefile зроблений по apendix1:

```
ifneq ($(KERNELRELEASE),)
# kbuild part of makefile
obj-m := hello3.o
ccflags-y += -g                 # add debugging info
else
# normal makefile
KDIR ?= /lib/modules/`uname -r`/build

default:
        $(MAKE) -C $(KDIR) M=$$PWD
        cp hello3.ko hello3.ko.unstripped
        $(CROSS_COMPILE)strip -g hello3.ko       # strip only debugging info

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

%.s %.i: %.c                    # just use make hello.s instead of objdump
        $(MAKE) -C $(KDIR) M=$$PWD $@

endif
```

Скріншоти виконання та перевірки

Вхідний параметр 8:

```
/ # insmod hello1.ko
[  108.034318] hello1: loading out-of-tree module taints kernel.
/ # insmod hello2.ko myParam=8
[  120.121231] Calling print_hello() from hello2...
[  120.122257] Warning: 8
[  120.122454] Hello world!
[  120.122800] Hello world!
[  120.123575] Hello world!
[  120.124121] Hello world!
[  120.124389] Hello world!
[  120.124653] Hello world!
[  120.124797] Hello world!
[  120.125046] Hello world!
/ # rmmod hello2.ko
/ # rmmod hello1.ko
[  184.535990] Time in nanoseconds: 290384
[  184.538083] Time in nanoseconds: 772272
[  184.544546] Time in nanoseconds: 514800
[  184.544972] Time in nanoseconds: 264704
[  184.545606] Time in nanoseconds: 261568
[  184.545761] Time in nanoseconds: 145552
[  184.545924] Time in nanoseconds: 243664
[  184.546283] Time in nanoseconds: 596960
```

Вхідний параметр 11 (максимально можливий - 10):

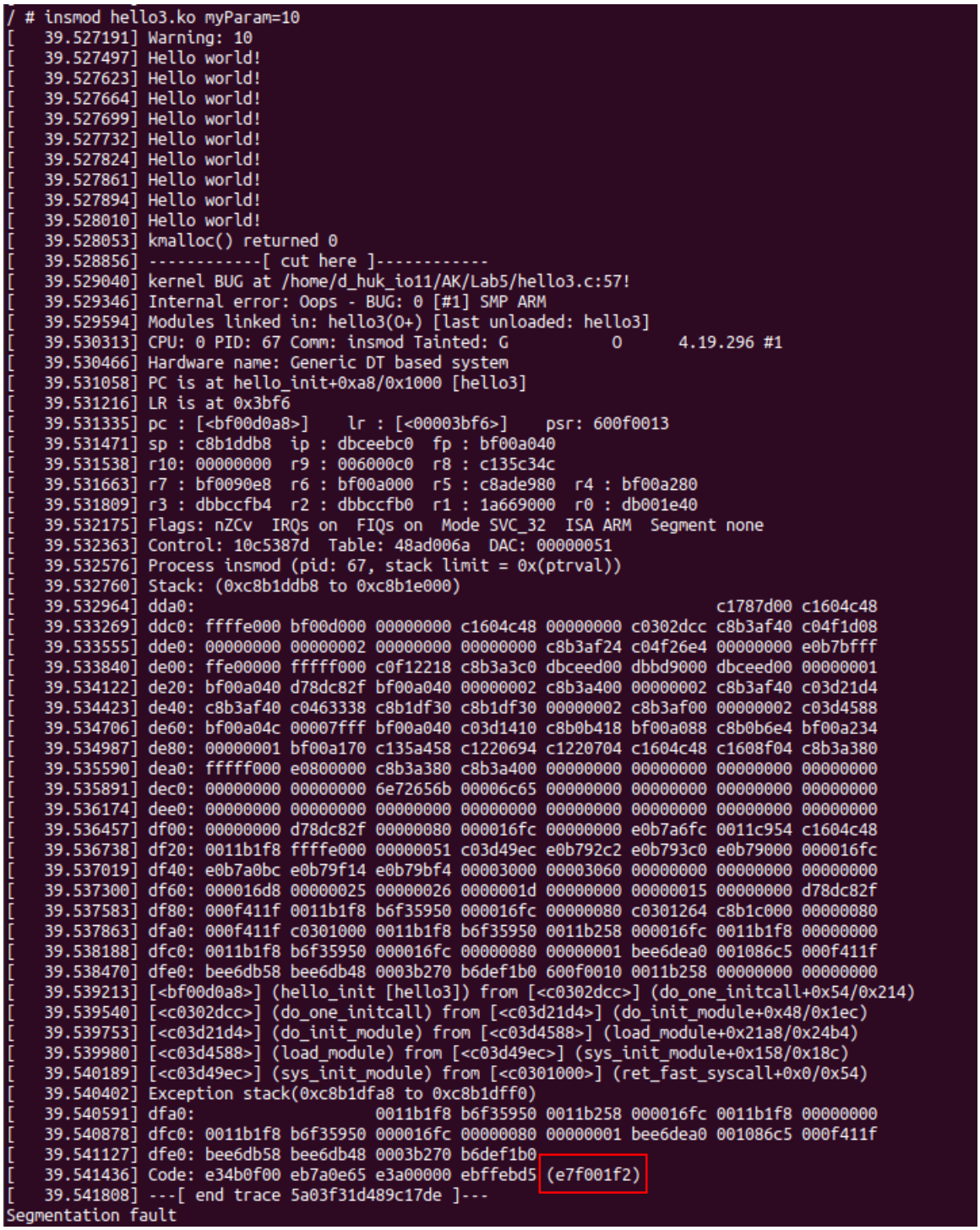
```
/ # insmod hello3.ko myParam=11
[   72.044499] ------------[ cut here ]------------
[   72.044709] kernel BUG at /home/d_huk_io11/AK/Lab5/hello3.c:41!
[   72.044986] Internal error: Oops - BUG: 0 [#1] SMP ARM
[   72.045281] Modules linked in: hello3(O+) [last unloaded: hello3]
[   72.045659] CPU: 0 PID: 67 Comm: insmod Tainted: G           O      4.19.296 #1
[   72.045832] Hardware name: Generic DT based system
[   72.046437] PC is at hello_init+0x18/0x1000 [hello3]
[   72.046800] LR is at do_one_initcall+0x54/0x214
[   72.046932] pc : [<bf00d018>]    lr : [<c0302dcc>]    psr: 200f0013
[   72.047222] sp : c8ad1db8  ip : c8ae0400  fp : bf00a040
[   72.047416] r10: 00000000  r9 : c1604c48  r8 : 00000000
[   72.047492] r7 : bf00d000  r6 : bf00a000  r5 : c1604c48  r4 : c1787d00
[   72.047574] r3 : 00000000  r2 : d7077925  r1 : 0000000b  r0 : 00000000
[   72.047744] Flags: nzCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
[   72.047841] Control: 10c5387d  Table: 48b5806a  DAC: 00000051
[   72.048147] Process insmod (pid: 67, stack limit = 0x(ptrval))
[   72.048322] Stack: (0xc8ad1db8 to 0xc8ad2000)
[   72.048529] 1da0:                                                       c1787d00 c1604c48
[   72.048782] 1dc0: ffffe000 bf00d000 00000000 c1604c48 00000000 c0302dcc c8ae0f00 c04f1d08
[   72.049070] 1de0: 00000000 00000002 00000000 00000000 c8ae0ee4 c04f26e4 00000000 e0b7bfff
[   72.049318] 1e00: ffe00000 fffff000 c0f12218 c8ae0480 dbceecc0 dbbd9000 dbceecc0 00000001
[   72.049489] 1e20: bf00a040 d7077925 bf00a040 00000002 c8ae0440 00000002 c8ae0f00 c03d21d4
[   72.049822] 1e40: c8ae0f00 c0463338 c8ad1f30 c8ad1f30 00000002 c8ae0ec0 00000002 c03d4588
[   72.050067] 1e60: bf00a04c 00007fff bf00a040 c03d1410 c8b4b418 bf00a088 c8b4b6e4 bf00a234
[   72.050314] 1e80: 00000001 bf00a170 c135a458 c1220694 c1220704 c1604c48 c1608f04 c8ae0400
[   72.050546] 1ea0: fffff000 e0800000 c8ae0400 c8ae0440 00000000 00000000 00000000 00000000
[   72.050775] 1ec0: 00000000 00000000 6e72656b 00006c65 00000000 00000000 00000000 00000000
[   72.050947] 1ee0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[   72.051299] 1f00: 00000000 d7077925 00000080 000016fc 00000000 e0b7a6fc 0011c954 c1604c48
[   72.051532] 1f20: 0011b1f8 ffffe000 00000051 c03d49ec e0b792c2 e0b793c0 e0b79000 000016fc
[   72.051762] 1f40: e0b7a0bc e0b79f14 e0b79bf4 00003000 00003060 00000000 00000000 00000000
[   72.051998] 1f60: 000016d8 00000025 00000026 0000001d 00000000 00000015 00000000 d7077925
[   72.052401] 1f80: 000f411f 0011b1f8 b6fb5950 000016fc 00000080 c0301264 c8ad0000 00000080
[   72.052638] 1fa0: 000f411f c0301000 0011b1f8 b6fb5950 0011b258 000016fc 0011b1f8 00000000
[   72.052865] 1fc0: 0011b1f8 b6fb5950 000016fc 00000080 00000001 bec7fea0 001086c5 000f411f
[   72.053095] 1fe0: bec7fb58 bec7fb48 0003b270 b6e6f1b0 600f0010 0011b258 00000000 00000000
[   72.054497] [<bf00d018>] (hello_init [hello3]) from [<c0302dcc>] (do_one_initcall+0x54/0x214)
[   72.054733] [<c0302dcc>] (do_one_initcall) from [<c03d21d4>] (do_init_module+0x48/0x1ec)
[   72.054901] [<c03d21d4>] (do_init_module) from [<c03d4588>] (load_module+0x21a8/0x24b4)
[   72.055534] [<c03d4588>] (load_module) from [<c03d49ec>] (sys_init_module+0x158/0x18c)
[   72.057988] [<c03d49ec>] (sys_init_module) from [<c0301000>] (ret_fast_syscall+0x0/0x54)
[   72.058160] Exception stack(0xc8ad1fa8 to 0xc8ad1ff0)
[   72.058376] 1fa0:                   0011b1f8 b6fb5950 0011b258 000016fc 0011b1f8 00000000
[   72.058621] 1fc0: 0011b1f8 b6fb5950 000016fc 00000080 00000001 bec7fea0 001086c5 000f411f
[   72.058923] 1fe0: bec7fb58 bec7fb48 0003b270 b6e6f1b0
[   72.059426] Code: e34b6f00 e5961008 e351000a 9a000000 (e7f001f2)
[   72.059812] ---[ end trace f7ea3aa01a9f5b47 ]---
Segmentation fault
```

Спроба force remove:

```
/ # rmmod -f hello3.ko
rmmod: remove 'hello3': Device or resource busy
```

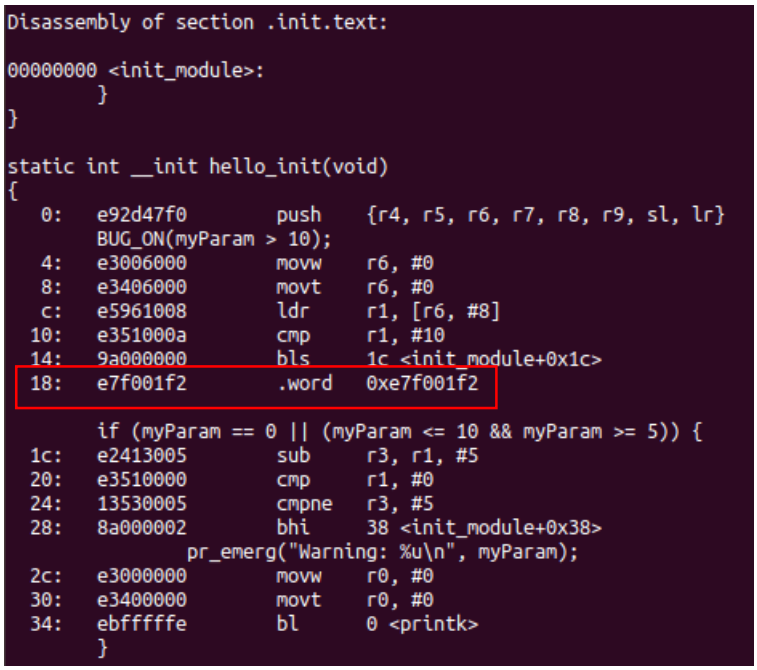
Для вирішення цієї проблеми допомагає перезапуск емулятора.

Вхідний параметр 10 (Помилка на 9-му елементі):



Помилки OOPS, або "Out Of Kernel Space", можуть виникнути в результаті серйозної помилки в коді ядра або в модулях ядра, які призводять до некоректної поведінки або навіть до аварійного завершення роботи системи. У нашому випадку – це навмисно створені помилки за допомогою BUG() та BUG_ON().

Дизасемблерний код:



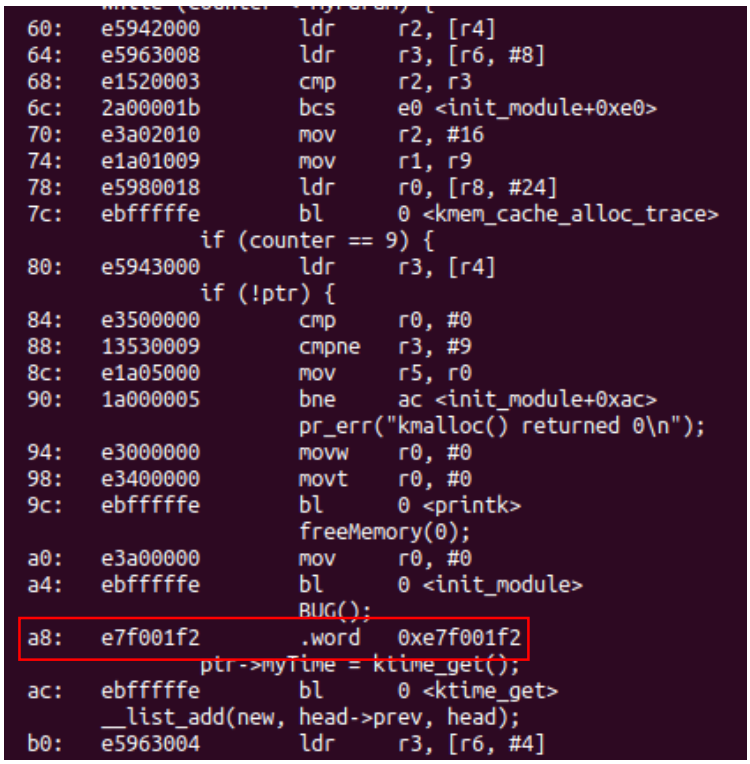
```
Disassembly of section .init.text:

00000000 <init_module>:
        }
}

static int __init hello_init(void)
{
   0:   e92d47f0        push    {r4, r5, r6, r7, r8, r9, sl, lr}
        BUG_ON(myParam > 10);
   4:   e3006000        movw    r6, #0
   8:   e3406000        movt    r6, #0
   c:   e5961008        ldr     r1, [r6, #8]
  10:   e351000a        cmp     r1, #10
  14:   9a000000        bls     1c <init_module+0x1c>
  18:   e7f001f2        .word   0xe7f001f2

        if (myParam == 0 || (myParam <= 10 && myParam >= 5)) {
  1c:   e2413005        sub     r3, r1, #5
  20:   e3510000        cmp     r1, #0
  24:   13530005        cmpne   r3, #5
  28:   8a000002        bhi     38 <init_module+0x38>
                pr_emerg("Warning: %u\n", myParam);
  2c:   e3000000        movw    r0, #0
  30:   e3400000        movt    r0, #0
  34:   ebffffe         bl      0 <printk>
        }
```

…



```
  60:   e5942000        ldr     r2, [r4]
  64:   e5963008        ldr     r3, [r6, #8]
  68:   e1520003        cmp     r2, r3
  6c:   2a00001b        bcs     e0 <init_module+0xe0>
  70:   e3a02010        mov     r2, #16
  74:   e1a01009        mov     r1, r9
  78:   e5980018        ldr     r0, [r8, #24]
  7c:   ebffffe         bl      0 <kmem_cache_alloc_trace>
                if (counter == 9) {
  80:   e5943000        ldr     r3, [r4]
                if (!ptr) {
  84:   e3500000        cmp     r0, #0
  88:   13530009        cmpne   r3, #9
  8c:   e1a05000        mov     r5, r0
  90:   1a000005        bne     ac <init_module+0xac>
                        pr_err("kmalloc() returned 0\n");
  94:   e3000000        movw    r0, #0
  98:   e3400000        movt    r0, #0
  9c:   ebffffe         bl      0 <printk>
                        freeMemory(0);
  a0:   e3a00000        mov     r0, #0
  a4:   ebffffe         bl      0 <init_module>
                        BUG();
  a8:   e7f001f2        .word   0xe7f001f2
                ptr->myTime = ktime_get();
  ac:   ebffffe         bl      0 <ktime_get>
        __list_add(new, head->prev, head);
  b0:   e5963004        ldr     r3, [r6, #4]
```

Посилання на репозиторій github:

https://github.com/Reiny24/AK-2-IO-11-Huk-Lab-works/tree/Lab-5