Міністерство освіти та науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Лабораторна робота №6

з дисципліни “Технології програмування для комп'ютерних систем – 3”

Виконала:
студентка групи ІВ-91мн
Дорошенко А.Ю.

Київ 2020 р.

Завдання

Ознайомитися із дебагінгом модулю.

Послідовність виконання роботи

1. Створюємо модуль. Для цього у файловій системі створюємо директорію `lab6`, у яку додаємо усі необхідні файли для роботи з модулем. Виконуємо команди `export` для збирання ядра, а також `export KDIR=/home/anna/lab3/linux-stable/.`

```
anna@ubuntu:~/lab3/busybox/_install/lab6$ ll
total 20
drwxr-xr-x  2 anna anna 4096 May 23 08:54 ./
drwxr-xr-x 16 anna anna 4096 May 23 08:54 ../
-rw-------  1 anna anna   46 May 23 08:54 Kbuild
-rw-------  1 anna anna  136 May 23 08:54 Makefile
-rw-------  1 anna anna 1567 May 23 08:54 module5.c
```

```
anna@ubuntu:~/lab3/busybox/_install/lab6$ export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:$PATH
anna@ubuntu:~/lab3/busybox/_install/lab6$ export CROSS_COMPILE='ccache arm-eabi-'
anna@ubuntu:~/lab3/busybox/_install/lab6$ export ARCH=arm
anna@ubuntu:~/lab3/busybox/_install/lab6$ export KDIR=/home/anna/lab3/linux-stable/
```

2. Виконуємо команду `make` для збирання модулю:

```
anna@ubuntu:~/lab3/busybox/_install/lab6$ make
make -C /home/anna/lab3/linux-stable/ M=$PWD
make[1]: Entering directory '/home/anna/lab3/linux-stable'
  CC [M]  /home/anna/lab3/busybox/_install/lab6/module5.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/anna/lab3/busybox/_install/lab6/module5.mod.o
  LD [M]  /home/anna/lab3/busybox/_install/lab6/module5.ko
make[1]: Leaving directory '/home/anna/lab3/linux-stable'
```

3. Створимо архів `CPIO` для `rootfs` та заархівуємо його за допомогою `GZip`:

```
anna@ubuntu:~/lab3/busybox/_install$ find . | cpio -o -H newc | gzip > ../rootfs.cpio.gz
118689 blocks
```

4. Виконання завдання Basic1:

Замінимо виведення повідомлення та повернення `-EINVAL` для неприпустимого значення параметра викликом функції `BUG_ON()`.

Додаємо навмисне внесення помилки під час формування останнього елементу списку.

До `Kbuild` додаємо прапорець `-g`:

```
# kbuild part of makefile
ccflags-y += -g
obj-m := module4.o
```

Нижче наведено змінений фрагмент коду:

```
static void print_text(unsigned int repeats)
{
        unsigned int repeat;
        struct time_keeper *ptr;

        for (repeat = 0; repeat < repeats; repeat++) {
                if (repeat == repeats - 1)
                        ptr = 0;
                else
                        ptr = kmalloc(sizeof(*ptr), GFP_KERNEL);
                ptr->time_before = ktime_get();
                pr_info("Hello there!\n");
                ptr->time_after = ktime_get();
                list_add(&ptr->time_list, &lab5_list_head);
        }
}

static unsigned int repeats = 1;

module_param(repeats, uint, '0444');
MODULE_PARM_DESC(repeats, "How many hello to print");

static int __init module4_init(void)
{
        BUG_ON(repeats > 10);

        if (repeats >= 5 && repeats <= 10)
                pr_warn("Repeatition from 5 to 10 times\n");

        if (repeats == 0)
                pr_warn("No repeatition\n");

        print_text(repeats);
        return 0;
}
```

5. Протестуємо роботу модуля:

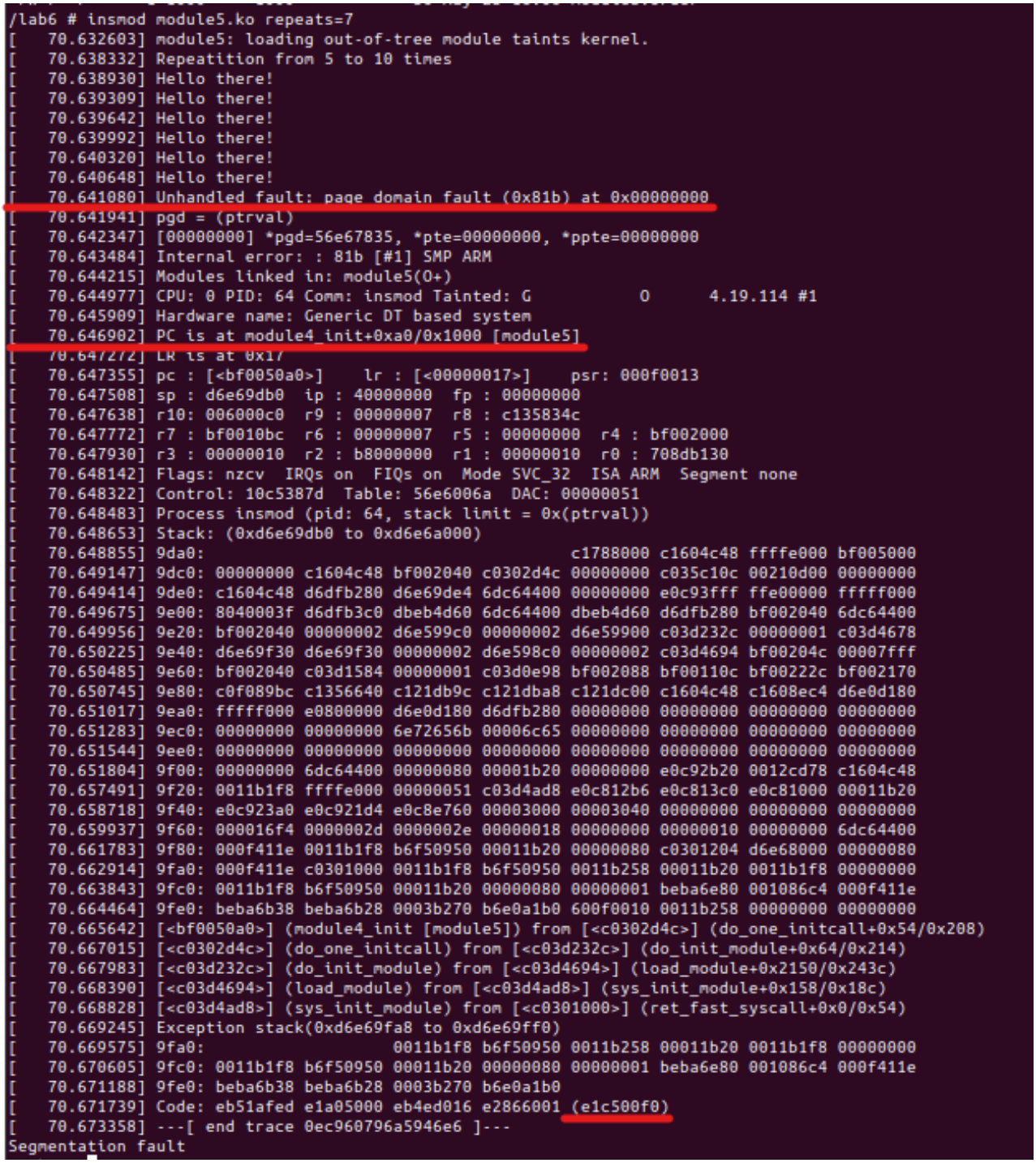
Можна побачити, що завантаження модулю зі значенням параметру, який є більшим за 10 (у прикладі `repeats = 12`), призводить до виконання макросу `BUG_ON`.

```
/lab6 # insmod module5.ko repeats=12
[   36.098361] module5: loading out-of-tree module taints kernel.
[   36.103333] ------------[ cut here ]------------
[   36.104038] kernel BUG at /home/anna/lab3/busybox/_install/lab6/module5.c:40!
[   36.105030] Internal error: Oops - BUG: 0 [#1] SMP ARM
[   36.105793] Modules linked in: module5(O+)
[   36.106611] CPU: 0 PID: 62 Comm: insmod Tainted: G           O      4.19.114 #1
[   36.107619] Hardware name: Generic DT based system
[   36.108701] PC is at module4_init+0x18/0x1000 [module5]
[   36.109532] LR is at do_one_initcall+0x54/0x208
[   36.110112] pc : [<bf005018>]    lr : [<c0302d4c>]    psr: 200f0013
[   36.110926] sp : d6e6bdb0  ip : d6deba40  fp : 00000000
[   36.111056] r10: bf002040  r9 : c1604c48  r8 : 00000000
[   36.111188] r7 : bf005000  r6 : ffffe000  r5 : c1604c48  r4 : bf002000
[   36.111344] r3 : 0000000c  r2 : 6dc64400  r1 : 00003c40  r0 : 00000000
[   36.111550] Flags: nzCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
[   36.111729] Control: 10c5387d  Table: 56e6406a  DAC: 00000051
[   36.111947] Process insmod (pid: 62, stack limit = 0x(ptrval))
[   36.112118] Stack: (0xd6e6bdb0 to 0xd6e6c000)
[   36.112317] bda0:                                     c1788000 c1604c48 ffffe000 bf005000
[   36.112660] bdc0: 00000000 c1604c48 bf002040 c0302d4c 00000000 c035c10c 00210d00 00000000
[   36.112921] bde0: c1604c48 d6dfb280 d6e6bde4 6dc64400 00000000 e0c93fff ffe00000 fffff000
[   36.113177] be00: 8040003f d6dfb3c0 dbeb5a60 6dc64400 dbeb5a60 d6dfb280 bf002040 6dc64400
[   36.113803] be20: bf002040 00000002 d6deb9c0 00000002 d6deb900 c03d232c 00000001 c03d4678
[   36.114064] be40: d6e6bf30 d6e6bf30 00000002 d6deb8c0 00000002 c03d4694 bf00204c 00007fff
[   36.114320] be60: bf002040 c03d1584 00000001 c03d0e98 bf002088 bf00110c bf00222c bf002170
[   36.119668] be80: c0f089bc c1356640 c121db9c c121dba8 c121dc00 c1604c48 c1608ec4 d6e0d180
[   36.121219] bea0: fffff000 e0800000 d6e0d180 d6dfb280 00000000 00000000 00000000 00000000
[   36.122415] bec0: 00000000 00000000 6e72656b 00006c65 00000000 00000000 00000000 00000000
[   36.123526] bee0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[   36.124636] bf00: 00000000 6dc64400 00000080 00001b20 00000000 e0c92b20 0012cd78 c1604c48
[   36.125793] bf20: 0011b1f8 ffffe000 00000051 c03d4ad8 e0c812b6 e0c813c0 e0c81000 00011b20
[   36.126535] bf40: e0c923a0 e0c921d4 e0c8e760 00003000 00003040 00000000 00000000 00000000
[   36.127099] bf60: 000016f4 0000002d 0000002e 00000018 00000000 00000010 00000000 6dc64400
[   36.127659] bf80: 000f411e 0011b1f8 b6fcb950 00011b20 00000080 c0301204 d6e6a000 00000080
[   36.128217] bfa0: 000f411e c0301000 0011b1f8 b6fcb950 0011b258 00011b20 0011b1f8 00000000
[   36.128824] bfc0: 0011b1f8 b6fcb950 00011b20 00000080 00000001 befc3e80 001086c4 000f411e
[   36.129471] bfe0: befc3b38 befc3b28 0003b270 b6e851b0 600f0010 0011b258 00000000 00000000
[   36.130434] [<bf005018>] (module4_init [module5]) from [<c0302d4c>] (do_one_initcall+0x54/0x208)
[   36.131667] [<c0302d4c>] (do_one_initcall) from [<c03d232c>] (do_init_module+0x64/0x214)
[   36.132239] [<c03d232c>] (do_init_module) from [<c03d4694>] (load_module+0x2150/0x243c)
[   36.132675] [<c03d4694>] (load_module) from [<c03d4ad8>] (sys_init_module+0x158/0x18c)
[   36.133129] [<c03d4ad8>] (sys_init_module) from [<c0301000>] (ret_fast_syscall+0x0/0x54)
[   36.133558] Exception stack(0xd6e6bfa8 to 0xd6e6bff0)
[   36.133895] bfa0:                   0011b1f8 b6fcb950 0011b258 00011b20 0011b1f8 00000000
[   36.134475] bfc0: 0011b1f8 b6fcb950 00011b20 00000080 00000001 befc3e80 001086c4 000f411e
[   36.135027] bfe0: befc3b38 befc3b28 0003b270 b6e851b0
[   36.135538] Code: e34b4f00 e5943000 e353000a 9a000000 (e7f001f2)
[   36.136330] ---[ end trace f6ec302c2862f631 ]---
Segmentation fault
```

За допомогою утиліти objdump можна побачити, що значення `PC` та рядку з `BUG_ON` є ідентичними.

```
anna@ubuntu:~/lab3/busybox/_install/lab6$ arm-eabi-objdump -dS module5.ko

module5.ko:     file format elf32-littlearm


Disassembly of section .init.text:

00000000 <init_module>:

module_param(repeats, uint, S_IRUGO);
MODULE_PARM_DESC(repeats, "How many hello to print");

static int __init module4_init(void)
{
   0:   e92d47f0        push    {r4, r5, r6, r7, r8, r9, sl, lr}
        BUG_ON(repeats > 10);
   4:   e3004000        movw    r4, #0
   8:   e3404000        movt    r4, #0
   c:   e5943000        ldr     r3, [r4]
  10:   e353000a        cmp     r3, #10
  14:   9a000000        bls     1c <init_module+0x1c>
  18:   e7f001f2        .word   0xe7f001f2
```

Якщо ж вести значення параметру менше за 10, то при завантаженні модуля (з параметром, наприклад, `repeats=7`), побачимо `null pointer dereference`.



Як результат виконання objdump отримали таке:

```
                    if (!index)
                            return ZERO_SIZE_PTR;

                    return kmem_cache_alloc_trace(kmalloc_caches[index],
50:   e3008000          movw    r8, #0
54:   e3a0a0c0          mov     sl, #192        ; 0xc0
            pr_info("Hello there!\n");
58:   e3007000          movw    r7, #0
5c:   e3408000          movt    r8, #0
60:   e340a060          movt    sl, #96 ; 0x60
64:   e3407000          movt    r7, #0
      for (repeat = 0; repeat < repeats; repeat++) {
68:   e3a06000          mov     r6, #0
6c:   e1590006          cmp     r9, r6
70:   0a000017          beq     d4 <init_module+0xd4>
            if (repeat == repeats - 1)
74:   e2493001          sub     r3, r9, #1
78:   e1530006          cmp     r3, r6
                    ptr = 0;
7c:   03a05000          moveq   r5, #0
            if (repeat == repeats - 1)
80:   0a000004          beq     98 <init_module+0x98>
84:   e3a02018          mov     r2, #24
88:   e1a0100a          mov     r1, sl
8c:   e5980018          ldr     r0, [r8, #24]
90:   ebffffe           bl      0 <kmem_cache_alloc_trace>
94:   e1a05000          mov     r5, r0
            ptr->time_before = ktime_get();
98:   ebffffe           bl      0 <ktime_get>
      for (repeat = 0; repeat < repeats; repeat++) {
9c:   e2866001          add     r6, r6, #1
            ptr->time_before = ktime_get();
a0:   e1c500f0          strd    r0, [r5]
            pr_info("Hello there!\n");
a4:   e1a00007          mov     r0, r7
a8:   ebffffe           bl      0 <printk>
            ptr->time_after = ktime_get();
ac:   ebffffe           bl      0 <ktime_get>
```

6. Виконання завдання Basic2:

У функції `exit` модуля друк вмісту списку змінемо на `pr_debug` і додамо два виклики `pr_debug` до та після друку списку.

Для того, аби побачити зміни при завантаженні та вивантаженні модуля необхідно додати `#define DEBUG` на початку файлу та аби рівень логування був 8, аби виводилися `debug` повідомлення.

Встановлюємо параметр `CONFIG_DYNAMIC_DEBUG` у `~/lab3/linux-stable/fragments/bbb.cfg` та перезбираємо ядро.

```
# --- Networking ---
CONFIG_BRIDGE=y
CONFIG_DYNAMIC_DEBUG=y
```

```
anna@ubuntu:~/lab3/linux-stable$ ./scripts/kconfig/merge_config.sh arch/arm/configs/multi_v7_defconfig fragments/bbb.cfg
Using arch/arm/configs/multi_v7_defconfig as base
Merging fragments/bbb.cfg
Value of CONFIG_AM335X_PHY_USB is redefined by fragment fragments/bbb.cfg:
Previous value: CONFIG_AM335X_PHY_USB=m
New value: CONFIG_AM335X_PHY_USB=y

Value of CONFIG_USB_MUSB_TUSB6010 is redefined by fragment fragments/bbb.cfg:
Previous value: CONFIG_USB_MUSB_TUSB6010=m
New value: CONFIG_USB_MUSB_TUSB6010=y

Value of CONFIG_USB_MUSB_OMAP2PLUS is redefined by fragment fragments/bbb.cfg:
Previous value: CONFIG_USB_MUSB_OMAP2PLUS=m
New value: CONFIG_USB_MUSB_OMAP2PLUS=y

Value of CONFIG_USB_MUSB_HDRC is redefined by fragment fragments/bbb.cfg:
Previous value: CONFIG_USB_MUSB_HDRC=m
New value: CONFIG_USB_MUSB_HDRC=y
```

З’явився каталог `/sys/kernel/debug/dynamic_debug`:

```
anna@ubuntu:~/lab3/busybox/_install/lab6$ sudo ls -la /sys/kernel/debug/dynamic_debug
[sudo] password for anna:
total 0
drwxr-xr-x  2 root root 0 May 14 12:22 .
drwx------ 34 root root 0 May 14 12:22 ..
-rw-r--r--  1 root root 0 May 14 12:22 control
```

Нижче наведено змінений фрагмент коду:

```
static void __exit module5_exit(void)
{
        struct list_head *p;
        struct list_head *n;
        struct time_keeper *curr;

        pr_info("Module 5 exit\n");

        pr_debug("Before printing of the list");
        list_for_each_safe(p, n, &lab5_list_head) {
                curr = list_entry(p, struct time_keeper, time_list);
                pr_debug("Time needed for printing is: %lld(ns).\n",
                                curr->time_after - curr->time_before);
                list_del(p);
                kfree(curr);
        }
        pr_debug("After printing of the list");
}
```

Змінюючи прапорці у `/sys/kernel/debug/dynamic_debug/control` можна змінювати формат повідомлення для всього модулю та для окремих рядків.

```
/lab6 # echo 8 > /proc/sys/kernel/printk
/lab6 # insmod module5.ko repeats=3
[  153.173200] module5: loading out-of-tree module taints kernel.
[  153.179912] Hello there!
[  153.180438] Hello there!
[  153.180764] Hello there!
/lab6 # cat /sys/kernel/debug/dynamic_debug/control | grep module5
/home/anna/lab3/busybox/_install/lab6/module5.c:59 [module5]module5_exit =p "Before printing of the list\012"
/home/anna/lab3/busybox/_install/lab6/module5.c:63 [module5]module5_exit =p "Time needed for printing is: %lld(ns).\012"
/home/anna/lab3/busybox/_install/lab6/module5.c:67 [module5]module5_exit =p "After printing of the list\012"
/lab6 # echo 'file module5.c line 59 +mf' > /sys/kernel/debug/dynamic_debug/cont
rol
/lab6 # echo 'file module5.c line 59 +mf' > /sys/kernel/debug/dynamic_debug[  272.478981] random: fast init done
/lab6 # echo 'file module5.c line 63 -p' > /sys/kernel/debug/dynamic_debug/contr
ol
/lab6 # echo 'file module5.c line 67 =pl' > /sys/kernel/debug/dynamic_debug/cont
rol
/lab6 # cat /sys/kernel/debug/dynamic_debug/control | grep module5
/home/anna/lab3/busybox/_install/lab6/module5.c:59 [module5]module5_exit =pmf "Before printing of the list\012"
/home/anna/lab3/busybox/_install/lab6/module5.c:63 [module5]module5_exit =_ "Time needed for printing is: %lld(ns).\012"
/home/anna/lab3/busybox/_install/lab6/module5.c:67 [module5]module5_exit =pl "After printing of the list\012"
/lab6 # rmmod module5
[  339.839005] Module 5 exit
[  339.839675] module5:module5_exit: Before printing of the list
[  339.840490] 67: After printing of the list
```