

**Міністерство освіти і науки, молоді та спорту
Національний технічний університет України
“Київський політехнічний інститут ім. І. Сікорського”**

**Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №6

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

Виконав: Студент група ІВ-91мп

Михайленко А. М.

Викладач: Регіда П. Г.

Київ 2020

Хід роботи

Завдання: Ознайомитися із процесом дебагу модуля.

Послідовність виконання роботи

1. Для створення модуля виконаємо такі кроки: створимо директорію lab5, в директорію lab5 кладемо всі необхідні для роботи з модулем файли, виконаємо *export KDIR=/home/alex/repos/linux-stable/*

Виконаємо команди:

- *export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:\$PATH*
- *export CROSS_COMPILE='ccache arm-eabi-'*
- *export ARCH=arm*

2. Виконаємо команду *make*:

```
make -C /home/lara/repos/linux-stable/ M=$PWD
make[1]: Entering directory '/home/lara/repos/linux-stable'
CC [M] /home/lara/repos/busybox/_install/lab06/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/lara/repos/busybox/_install/lab06/hello.mod.o
LD [M] /home/lara/repos/busybox/_install/lab06/hello.ko
make[1]: Leaving directory '/home/lara/repos/linux-stable'
```

3. Створимо архів CPIO для *rootfs* та архівуємо його з GZip:

```
~/repos/busybox/_install$ find . | cpio -o -H newc | gzip > ../rootfs.cpio.gz
129035 blocks
```

Завдання Basic:

Викличемо функцію *BUG_ON()* у файлі *hello.c* і перевіримо виведення помилки на етапі формування четвертого елементу.

```
static int __init hello_init(void)
{
    unsigned int tick;
    struct timer *ptr;

    if (ticks == 0 || (ticks >= 5 && ticks <= 10))
        pr_warn("Warning\n");

    BUG_ON(ticks > 10);

    pr_debug("In the beginning...");
    for (tick = 0; tick < ticks; ++tick) {
        ptr = kmalloc(sizeof(*ptr), GFP_KERNEL);
```

```

    ptr->start = ktime_get();
    pr_debug("Ooohhhhhh! /tick %d/\n", tick);
    ptr->end = ktime_get();
    list_add(&ptr->list, &l_head);
}
pr_debug("In the end.....");

return 0;
}

```

Makefile:

```

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

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

```

Тестування завдання Basic:

```
/lab06 # insmod hello.ko ticks=14
[ 49.482381] hello: loading out-of-tree module taints kernel.
[ 49.507435] -----[ cut here ]-----
[ 49.508960] kernel BUG at /home/lara/repos/busybox/_install/lab06/hello.c:34!
[ 49.514602] Internal error: Oops - BUG: 0 [#1] SMP ARM
[ 49.515404] Modules linked in: hello(0+)
[ 49.518176] CPU: 0 PID: 63 Comm: insmod Tainted: G          0      4.19.114 #2
[ 49.519270] Hardware name: Generic DT based system
[ 49.521440] PC is at hello_init+0x38/0x1000 [hello]
[ 49.522949] LR is at do_one_initcall+0x54/0x208
[ 49.525276] pc : [<bf005038>]   lr : [<c0302d70>]   psr: 200f0013
[ 49.525834] sp : c9089db0 ip : c9100ac0 fp : 00000000
[ 49.526317] r10: bf002040 r9 : c1704c48 r8 : 00000000
[ 49.526865] r7 : bf005000 r6 : fffffe00 r5 : c1704c48 r4 : bf002000
[ 49.527439] r3 : 0000000e r2 : 00000009 r1 : 00003f86 r0 : 00000000
[ 49.528110] Flags: nzCv IRQs on Mode SVC_32 ISA ARM Segment none
[ 49.529481] Control: 10c5387d Table: 490dc06a DAC: 00000051
[ 49.531481] Process insmod (pid: 63, stack limit = 0x(ptrval))
[ 49.532234] Stack: (0xc9089db0 to 0xc908a000)
[ 49.532934] 9da0:                                c1888140 c1704c48 fffffe00 bf005000
[ 49.535260] 9dc0: 00000000 c1704c48 bf002040 c0302d70 db10c000 00000000 00210d00 c9089ddc
[ 49.537156] 9de0: c1704c48 c909d1c0 00000000 bf002088 bf002088 e0c93fff ffe00000 fffff000
[ 49.538751] 9e00: 8040003f c909d980 dbcfaf00 6dc64400 dbcfaf00 c909d1c0 bf002040 6dc64400
[ 49.540206] 9e20: bf002040 00000002 c9100a40 00000002 c9100980 c03d6160 00000001 c03d8638
[ 49.542823] 9e40: c9089f30 c9089f30 00000002 c9100940 00000002 c03d8654 bf00204c 00007fff
[ 49.544315] 9e60: bf002040 c03d5760 00000001 c03d4c1c bf002088 bf001118 bf00222c bf002170
[ 49.547849] 9e80: c139e018 c0f09750 c1231a60 c1704c48 c1708ec4 c90aa280 fffff000 e0800000
[ 49.549107] 9ea0: c90aa280 c909d1c0 00000000 00000000 00000000 00000000 00000000 00000000
[ 49.551325] 9ec0: 6e72656b 00006c65 00000000 00000000 00000000 00000000 00000000 00000000
[ 49.552287] 9ee0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 6dc64400
[ 49.555910] 9f00: 00000080 00001d08 00000000 e0c92d08 0012cf58 c1704c48 0011b1f8 fffffe00
[ 49.557040] 9f20: 00000051 c03d8b10 00000000 00000000 e0c812b6 e0c813c0 e0c81000 00011d08
[ 49.558231] 9f40: e0c92538 e0c92360 e0c8e89c 00003000 00003040 bf002018 00000001 00000000
[ 49.559280] 9f60: 00001764 0000002f 00000030 0000001a 00000000 00000010 00000000 6dc64400
[ 49.560262] 9f80: 000f411e 0011b1f8 b6f44950 00011d08 00000080 c0301204 c9088000 00000080
[ 49.561218] 9fa0: 000f411e c0301000 0011b1f8 b6f44950 0011b250 00011d08 0011b1f8 00000000
[ 49.563461] 9fc0: 0011b1f8 b6f44950 00011d08 00000080 00000001 bee1fe90 001086c4 000f411e
[ 49.564459] 9fe0: bee1fb48 bee1fb38 0003b270 b6dfe1b0 600f0010 0011b250 00000000 00000000
[ 49.566322] [<bf005038>] (hello_init [hello]) from [<c0302d70>] (do_one_initcall+0x54/0x208)
[ 49.570742] [<c0302d70>] (do_one_initcall) from [<c03d6160>] (do_init_module+0x64/0x214)
[ 49.572057] [<c03d6160>] (do_init_module) from [<c03d8654>] (load_module+0x22dc/0x2628)
[ 49.572893] [<c03d8654>] (load_module) from [<c03d8b10>] (sys_init_module+0x170/0x1c4)
[ 49.573758] [<c03d8b10>] (sys_init_module) from [<c0301000>] (ret_fast_syscall+0x0/0x54)
[ 49.575316] Exception stack(0xc9089fa8 to 0xc9089ff0)
[ 49.575637] 9fa0:                                0011b1f8 b6f44950 0011b250 00011d08 0011b1f8 00000000
[ 49.576142] 9fc0: 0011b1f8 b6f44950 00011d08 00000080 00000001 bee1fe90 001086c4 000f411e
[ 49.580929] 9fe0: bee1fb48 bee1fb38 0003b270 b6dfe1b0
[ 49.581546] Code: eb4e59e6 e5943000 e353000a 9a000000 (e7f001f2)
[ 49.583512] ---[ end trace 5cbcaaef856db255 ]---
```