

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

Лабораторна робота №5

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

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

Київ 2020 р.

Завдання

Ознайомитися з реалізацією зв'язних списків в ядрі OS Linux та з часовими відмітками.

Послідовність виконання роботи

1. Створюємо модуль. Для цього у файловій системі створюємо директорию lab5, 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/lab5$ ll
total 20
drwxr-xr-x  2 anna anna 4096 May 23 08:06 ./
drwxr-xr-x 15 anna anna 4096 May 23 06:45 ../
-rw-r--r--  1 anna anna   46 May 23 07:02 Kbuild
-rw-r--r--  1 anna anna  136 May 23 06:46 Makefile
-rw-r--r--  1 anna anna 1500 May 23 08:06 module4.c

anna@ubuntu:~/lab3/busybox/_install/lab5$ export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:$PATH
anna@ubuntu:~/lab3/busybox/_install/lab5$ export CROSS_COMPILE='ccache arm-eabi-'
anna@ubuntu:~/lab3/busybox/_install/lab5$ export ARCH=arm
anna@ubuntu:~/lab3/busybox/_install/lab5$ export KDIR=/home/anna/lab3/linux-stable/
```

2. Виконуємо команду make для збирання модулю:

```
anna@ubuntu:~/lab3/busybox/_install/lab5$ 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/lab5/module4.o
Building modules, stage 2.
MODPOST 1 modules
LD [M] /home/anna/lab3/busybox/_install/lab5/module4.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. Перевіримо Linux kernel coding style:

```
anna@ubuntu:~/lab3/busybox/_install/lab5$ ~/lab3/linux-stable/scripts/checkpatch.pl -f module4.c
WARNING: Missing or malformed SPDX-License-Identifier tag in line 1
#1: FILE: module4.c:1:
+#include <linux/init.h>

total: 0 errors, 1 warnings, 74 lines checked

NOTE: For some of the reported defects, checkpatch may be able to
mechanically convert to the typical style using --fix or --fix-inplace.

module4.c has style problems, please review.

NOTE: If any of the errors are false positives, please report
them to the maintainer, see CHECKPATCH in MAINTAINERS.
```

5. Виконання завдання Basic:

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

#include <linux/ktime.h>
#include <linux/slab.h>

static LIST_HEAD(lab5_list_head);

struct time_keeper
{
    ktime_t time_before;
    struct list_head time_list;
};

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

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

static unsigned int repeats = 1;

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

static int __init module4_init(void)
{
    if (repeats > 10)
    {
        pr_err("Cannot repeat more than 10 times\n");
        return -EINVAL;
    }
}
```

```

    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;
}

static void __exit module4_exit(void)
{
    struct list_head *p;
    struct list_head *n;
    struct time_keeper *curr;

    pr_info("Module 3 exit\n");

    list_for_each_safe(p, n, &lab5_list_head)
    {
        curr = list_entry(p, struct time_keeper, time_list);
        pr_info("Before function time: %lld(ns).\n", curr->time_before);
        list_del(p);
        kfree(curr);
    }
}

module_init(module4_init);
module_exit(module4_exit);

MODULE_AUTHOR("Anna Doroshenko");
MODULE_DESCRIPTION("Training to work with list");
MODULE_LICENSE("Dual BSD/GPL");

```

6. Протестуємо роботу модуля:

```

/lab5 # insmod module4.ko repeats=4
[ 81.383741] Hello there!
[ 81.384167] Hello there!
[ 81.384496] Hello there!
[ 81.384875] Hello there!
/lab5 # rmmod module4
[ 83.594854] Module 3 exit
[ 83.595289] Before function time: 81355205984(ns).
[ 83.595968] Before function time: 81354828800(ns).
[ 83.596576] Before function time: 81354498880(ns).
[ 83.597240] Before function time: 81354060592(ns).

```

7. Виконання завдання Advanced:

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

static LIST_HEAD(lab5_list_head);

struct time_keeper
{
    ktime_t time_before;
    ktime_t time_after;
    struct list_head time_list;
};

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

    for (repeat = 0; repeat < repeats; repeat++)
    {
        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, S_IRUGO);
MODULE_PARM_DESC(repeats, "How many hello to print");

static int __init module4_init(void)
{
    if (repeats > 10)
    {
        pr_err("Cannot repeat more than 10 times\n");
        return -EINVAL;
    }
}
```

```

    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;
}

static void __exit module4_exit(void)
{
    struct list_head *p;
    struct list_head *n;
    struct time_keeper *curr;

    pr_info("Module 3 exit\n");

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

module_init(module4_init);
module_exit(module4_exit);

MODULE_AUTHOR("Anna Doroshenko");
MODULE_DESCRIPTION("Training to work with list");
MODULE_LICENSE("Dual BSD/GPL");

```

8. Протестуємо роботу модулів:

```

/lab5 # rmmod module4
[ 73.082724] Module 3 exit
[ 73.083164] Time needed for printing is: 331920(ns).
[ 73.084315] Time needed for printing is: 431728(ns).

```