

**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 №5

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

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/lab05/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/lara/repos/busybox/_install/lab05/hello.mod.o
LD [M] /home/lara/repos/busybox/_install/lab05/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
```

4. Перевіримо **Linux kernel coding style**:

```
WARNING: Missing or malformed SPDX-License-Identifier tag in line 1
#1: FILE: hello.c:1:
+#include <linux/init.h>

WARNING: Symbolic permissions 'S_IRUGO' are not preferred. Consider using octal permissions '0444'
#21: FILE: hello.c:21:
+module_param(ticks, uint, S_IRUGO);

total: 0 errors, 2 warnings, 81 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.

hello.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.
```

Завдання Basic:

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

MODULE_DESCRIPTION("Lab5: linked list");
MODULE_LICENSE("Dual BSD/GPL");

static LIST_HEAD(l_head);

struct timer
{
    ktime_t start;
    ktime_t end;
    struct list_head list;
};

static unsigned int ticks = 1;

module_param(ticks, uint, S_IRUGO);
MODULE_PARM_DESC(ticks, "Number of Ticks to print");

static int shouldWarn(int number)
{
    if (number == 0 || (number >= 5 && number <= 10))
        return 1;
    else
        return 0;
}

static int shouldShowError(int number)
{
    if (number > 10)
        return 1;
    else
        return 0;
}

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

    if (shouldWarn(ticks))
```

```

        pr_warn("Warning\n");

    if (shouldShowError(ticks)) {
        pr_err("Too many ticks\n");
        return -EINVAL;
    }

    for (tick = 0; tick < ticks; ++tick) {
        ptr = kmalloc(sizeof(*ptr), GFP_KERNEL);
        ptr->start = ktime_get();
        pr_info("Oooohhhhh! /tick %d/\n", tick);
        ptr->end = ktime_get();
        list_add(&ptr->list, &l_head);
    }

    return 0;
}

static void __exit hello_exit(void)
{
    struct list_head *pos;
    struct list_head *num;
    struct timer *cur;

    pr_info("Exit module\n");

    list_for_each_safe(pos, num, &l_head)
    {
        cur = list_entry(pos, struct timer, list);
        pr_info("Time taken = %lld ns\n", cur->end - cur->start);
        list_del(pos);
        kfree(cur);
    }
}

module_init(hello_init);
module_exit(hello_exit);

```

Тестування завдання Basic:

```

/lab05 # insmod hello.ko ticks=3
[ 44.906053] hello: loading out-of-tree module taints kernel.
[ 44.916621] Oooohhhhh! /tick 0/
[ 44.917461] Oooohhhhh! /tick 1/
[ 44.923964] Oooohhhhh! /tick 2/
/lab05 # rmmod hello
[ 57.704092] Exit module
[ 57.705174] Start = 44849782432 ns
[ 57.706071] Start = 44843287808 ns
[ 57.706852] Start = 44842432592 ns

```

Завдання Advanced:

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

MODULE_DESCRIPTION("Lab5: linked list");
MODULE_LICENSE("Dual BSD/GPL");

static LIST_HEAD(l_head);

struct timer
{
    ktime_t start;
    ktime_t end;
    struct list_head list;
};

static unsigned int ticks = 1;

module_param(ticks, uint, S_IRUGO);
MODULE_PARM_DESC(ticks, "Number of Ticks to print");

static int shouldWarn(int number)
{
    if (number == 0 || (number >= 5 && number <= 10))
        return 1;
    else
        return 0;
}

static int shouldShowError(int number)
{
    if (number > 10)
        return 1;
    else
        return 0;
}

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

    if (shouldWarn(ticks))
```

```

        pr_warn("Warning\n");

    if (shouldShowError(ticks)) {
        pr_err("Too many ticks\n");
        return -EINVAL;
    }

    for (tick = 0; tick < ticks; ++tick) {
        ptr = kmalloc(sizeof(*ptr), GFP_KERNEL);
        ptr->start = ktime_get();
        pr_info("Oooohhhhh! /tick %d/\n", tick);
        ptr->end = ktime_get();
        list_add(&ptr->list, &l_head);
    }

    return 0;
}

static void __exit hello_exit(void)
{
    struct list_head *pos;
    struct list_head *num;
    struct timer *cur;

    pr_info("Exit module\n");

    list_for_each_safe(pos, num, &l_head)
    {
        cur = list_entry(pos, struct timer, list);
        pr_info("Time taken = %lld ns\n", cur->end - cur->start);
        list_del(pos);
        kfree(cur);
    }
}

module_init(hello_init);
module_exit(hello_exit);

```

Тестування завдання Advanced:

```

/ # cd lab05
/lab05 # insmod hello.ko ticks=3
[ 31.096295] hello: loading out-of-tree module taints kernel.
[ 31.117482] Oooohhhhhh! /tick 0/
[ 31.119942] Oooohhhhhh! /tick 1/
[ 31.121166] Oooohhhhhh! /tick 2/
/lab05 # rmmod hello
[ 38.585842] Exit module
[ 38.586633] Time taken = 1620608 ns
[ 38.587488] Time taken = 1207424 ns
[ 38.588344] Time taken = 2443424 ns

```