Operating System Hw1

學號 1073007S
姓名 葉致廷

Explanation of the code:

After including the needed headers file, we first **create a structure called node_student**, which contains the id and the exact birthday of a student, and also, **a struct list_head node_link**

- **struct list_head is like a pointer which links with another node**.

Then, we create **a constructor function** which **returns a node_student type object pointer with all the needed data**. An Important point here is that we use **kmalloc** to allocate space in memory since the module is loaded into the kernel space.

After that, **LIST_HEAD(student_list)** initializes a linked-list named student_list.

In module entry point, **int hw1_init,**
we **create 5 node_students object** with the constructor function, and once an instance is created, we use **list_add_tail(struct list_head *new, struct list_head *head)** to **add the node into the tail of the list.**
Soon after the list finishes adding 5 nodes, we use list_for_each_entry(pos, head, member) to iterate over our list. During the iteration, we print out the information of the current node.

- pos: the type* to use as a loop cursor, which is stu here, and stu is initally NULL.
- head: the head for our list, which is student_list here.
- member: the name of the list_head within the struct, which is node_link here.

In module exist point, **void hw1_exist,**
we use **list_for_each_entry_safe(pos, n, head, member)** to **iterate over the student list safe against removal of list entry**. During the iteration, we use list_del(struct list_head *entry) to delete the node and also use kfree(pos) to release resources locate at current position.

- pos: the type* to use as a loop cursor, which is stu here, and stu is initally NULL.
- n: another type * to use as temporary storage, which stores the next pos.
- head: the head for our list, which is student_list here.
- member: the name of the list_head within the struct, which is node_link here.

define in <linux/init.h>
module_init(hw1_init): driver initialization entry point.
module_exit(hw1_exit): driver exit point.

define in <linux/slab.h>
kmalloc:

```
kmalloc - allocate memory
 * @size: how many bytes of memory are required.
 * @flags: the type of memory to allocate. * %GFP_KERNEL - Allocate normal kernel ram.
```

kfree: deallocate memory

Screenshot:

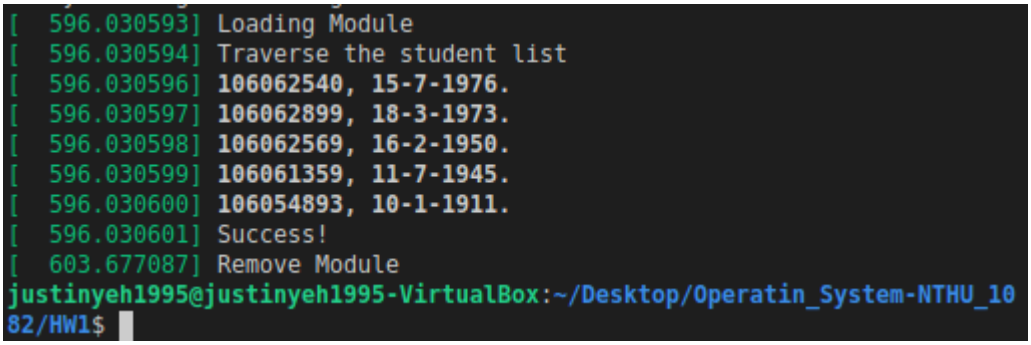
Code

hw1.c

C: > Users > justinyeh1995 > Desktop > Operating_System_NTHU_108-2 > OS_HW1 > hw1_1073007S > C hw1.c

```c
#include <linux/init.h>
#include <linux/list.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/slab.h>


/*Hw1 Data Structure*/
typedef struct node_student {
    int id;
    int year;
    int month;
    int day;
    struct list_head node_link;
}node_student;

struct node_student* create_student(int id, int y, int m, int d) {
    struct node_student *student ;
    student = kmalloc(sizeof(struct node_student), GFP_KERNEL);
    student->id = id;
    student->year = y;
    student->month = m;
    student->day = d;
    return student;
}
//Head of the list
LIST_HEAD(student_list);
// struct list_head student_list;
// INIT_LIST_HEAD(&student_list);

/* Module entry point */
int hw1_init(void) {
    printk(KERN_INFO "Loading Module\n");
    /*Construct students list */
    node_student* student1 = create_student(106062540,1976,7,15);
    list_add_tail(&student1->node_link, &student_list);
    node_student* student2 = create_student(106062899,1973,3,18);
    list_add_tail(&student2->node_link, &student_list);
    node_student* student3 = create_student(106062569,1950,2,16);
    list_add_tail(&student3->node_link, &student_list);
    node_student* student4 = create_student(106061359,1945,7,11);
    list_add_tail(&student4->node_link, &student_list);
    node_student* student5 = create_student(106054893,1911,1,10);
    list_add_tail(&student5->node_link, &student_list);
    /*Traverse the list*/
    printk(KERN_INFO "Traverse the student list\n");
    node_student* stu;
    list_for_each_entry(stu, &student_list, node_link) {
        printk("%d, %d-%d-%d.\n",stu->id,stu->day,stu->month,stu->year);
    };
    printk(KERN_INFO "Success!\n");
    return 0;
}

/* Module exist point */
void hw1_exit(void) {
    node_student *stu, *tmp;
    list_for_each_entry_safe(stu, tmp, &student_list, node_link) {
        list_del(&stu->node_link);
        kfree(stu);
    }
    printk(KERN_INFO "Remove Module\n");
}

/* Register module entry/exit pts */

module_init(hw1_init);
module_exit(hw1_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("HW1 Module");
MODULE_AUTHOR("justinyeh1995");
```

Output


[596.030593] Loading Module
[596.030594] Traverse the student list
[596.030596] 106062540, 15-7-1976.
[596.030597] 106062899, 18-3-1973.
[596.030598] 106062569, 16-2-1950.
[596.030599] 106061359, 11-7-1945.
[596.030600] 106054893, 10-1-1911.
[596.030601] Success!
[603.677087] Remove Module
justinyeh1995@justinyeh1995-VirtualBox:~/Desktop/Operatin_System-NTHU_10
82/HW1$