Operating System Programming Projects

Team members and responsibility：

109590003 王毓謙 寫報告

109590004 呂育瑋 寫 chap3 UNIX Shell and History Feature 程式

109590041 范遠皓 偵錯程式

109590043 柯瑞霖 寫 chap2 Linux Kernel Module 程式

Compilation or configuration instructions if it needs special environment to compile or run：

Linux Command：

```
sudo apt-get install linux-headers-5.4.0-144-generic
sudo apt install make
sudo apt install gcc
```

Chap.2 Project1 Linux Kernel Modules

Part I:

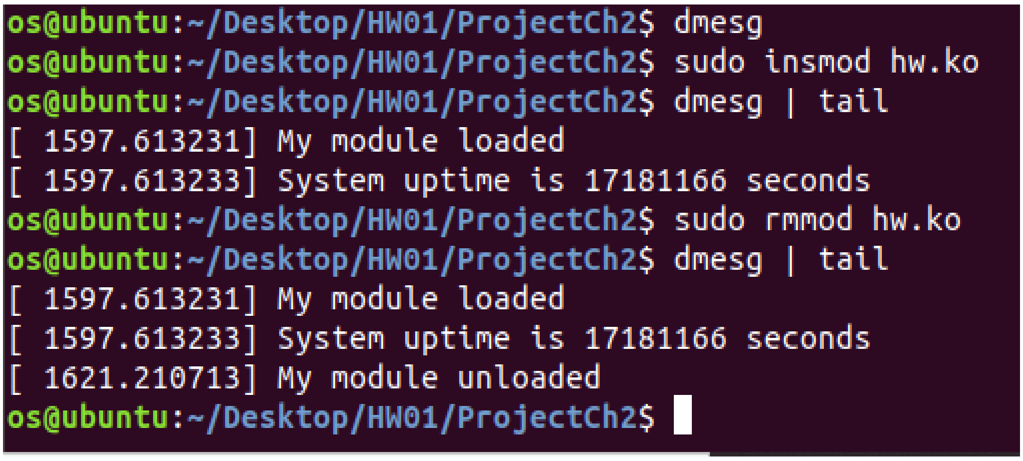
hw.c:

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/jiffies.h>

MODULE_LICENSE("GPL");

static int __init my_module_init(void)
{
    printk(KERN_INFO "My module loaded\n");
    printk(KERN_INFO "System uptime is %lu seconds\n", jiffies / HZ);
    return 0;
}

static void __exit my_module_exit(void)
{
    printk(KERN_INFO "My module unloaded\n");
}

module_init(my_module_init);
module_exit(my_module_exit);
```

Makefile:

```
obj-m += hw.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

command:



```
os@ubuntu:~/Desktop/HW01/ProjectCh2$ dmesg
os@ubuntu:~/Desktop/HW01/ProjectCh2$ sudo insmod hw.ko
os@ubuntu:~/Desktop/HW01/ProjectCh2$ dmesg | tail
[ 1597.613231] My module loaded
[ 1597.613233] System uptime is 17181166 seconds
os@ubuntu:~/Desktop/HW01/ProjectCh2$ sudo rmmod hw.ko
os@ubuntu:~/Desktop/HW01/ProjectCh2$ dmesg | tail
[ 1597.613231] My module loaded
[ 1597.613233] System uptime is 17181166 seconds
[ 1621.210713] My module unloaded
os@ubuntu:~/Desktop/HW01/ProjectCh2$ 
```

Part II:

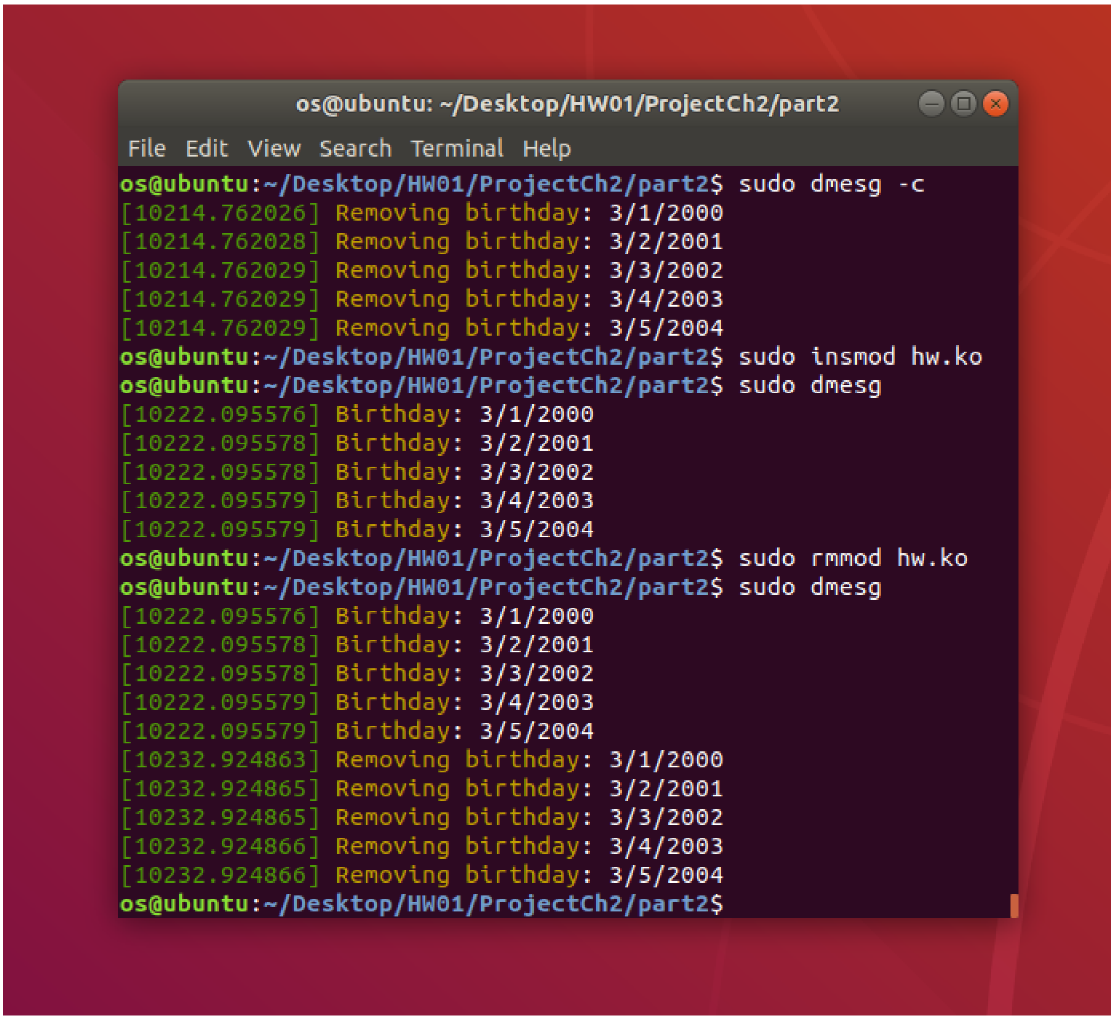
hw.c:

```
#include <linux/list.h> // 包含內核鏈表的相關函數和符號
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/jiffies.h>
#include <linux/slab.h>

struct birthday
{
    int day;
    int month;
    int year;
    struct list_head list;
};

static LIST_HEAD(birthday_list); // 定義鏈表頭

int __init my_module_init(void)
{
    int i;
    struct birthday *person;

    // 創建五個 struct_birthday 元素
    for (i = 0; i < 5; i++)
    {
        person = kmalloc(sizeof(*person), GFP_KERNEL);
        person->day = i + 1;
        person->month = 3;
        person->year = 2000 + i;
        INIT_LIST_HEAD(&person->list);                // 初始化元素的鏈表頭
        list_add_tail(&person->list, &birthday_list); // 將元素添加到鏈表中
    }
    // 遍歷鏈表，輸出每個元素的內容到內核日誌緩衝區
    struct birthday *ptr;
    list_for_each_entry(ptr, &birthday_list, list)
    {
        printk(KERN_INFO "Birthday: %d/%d/%d\n", ptr->month, ptr->day, ptr->year);
    }
    return 0;
}

void __exit my_module_exit(void)
{
    struct birthday *ptr, *next;
    list_for_each_entry_safe(ptr, next, &birthday_list, list)
    {
        printk(KERN_INFO "Removing birthday: %d/%d/%d\n", ptr->month, ptr->day, ptr->year);
        list_del(&ptr->list); // 從鏈表中刪除元素
        kfree(ptr);          // 釋放元素的內存空間
    }
}

module_init(my_module_init);
module_exit(my_module_exit);
```

Makefile:

```
obj-m += hw.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

command:



os@ubuntu: ~/Desktop/HW01/ProjectCh2/part2
File Edit View Search Terminal Help

```
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$ sudo dmesg -c
[10214.762026] Removing birthday: 3/1/2000
[10214.762028] Removing birthday: 3/2/2001
[10214.762029] Removing birthday: 3/3/2002
[10214.762029] Removing birthday: 3/4/2003
[10214.762029] Removing birthday: 3/5/2004
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$ sudo insmod hw.ko
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$ sudo dmesg
[10222.095576] Birthday: 3/1/2000
[10222.095578] Birthday: 3/2/2001
[10222.095578] Birthday: 3/3/2002
[10222.095579] Birthday: 3/4/2003
[10222.095579] Birthday: 3/5/2004
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$ sudo rmmod hw.ko
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$ sudo dmesg
[10222.095576] Birthday: 3/1/2000
[10222.095578] Birthday: 3/2/2001
[10222.095578] Birthday: 3/3/2002
[10222.095579] Birthday: 3/4/2003
[10222.095579] Birthday: 3/5/2004
[10232.924863] Removing birthday: 3/1/2000
[10232.924865] Removing birthday: 3/2/2001
[10232.924865] Removing birthday: 3/3/2002
[10232.924866] Removing birthday: 3/4/2003
[10232.924866] Removing birthday: 3/5/2004
os@ubuntu:~/Desktop/HW01/ProjectCh2/part2$
```

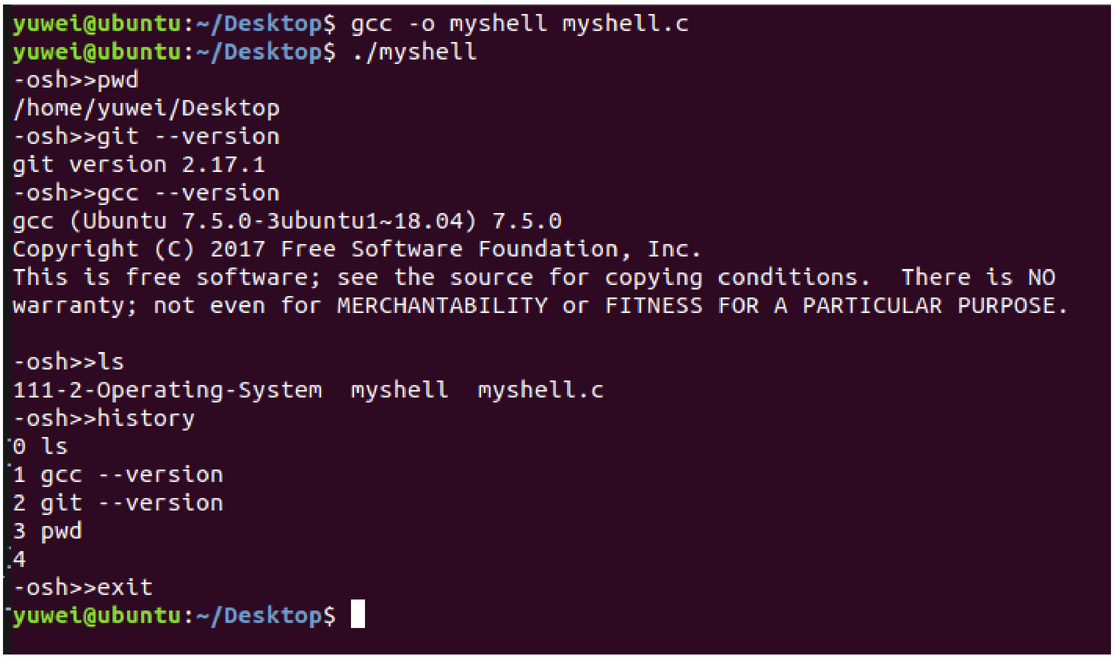
Chap.3 Project1 1: UNIX Shell and History Feature

myshell.c:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>

#define MAX_LINE 80
#define MAX_HISTORY 5

int main(void) {
    int keep_going = 1;
    char *args[MAX_LINE/2 + 1];
    char history_stack[MAX_HISTORY][MAX_LINE + 1] = {"",};
    while (keep_going) {
        printf("-osh>>");
        fflush(stdout);

        char input_char;
        char user_input[MAX_LINE + 1] = {0};
        int char_count = 0;

        while ((input_char = getchar()) != '\n' && input_char != EOF)
        {
            if(char_count < MAX_LINE)
                user_input[char_count] = input_char;
            else if(char_count == MAX_LINE)
                printf("[錯誤] 命令過長(最多 %d 字元)\n", MAX_LINE);
            char_count++;
        }
        if(char_count >= MAX_LINE || char_count == 0)
            continue; // input check
        user_input[MAX_LINE] = 0; // end of char[] '\0'
        if (strcmp(user_input, "exit") == 0){
            return 0;
        }

        if (strcmp(user_input, "history") == 0){
            for (int stack_i = 0; stack_i < MAX_HISTORY; stack_i++) {
                printf("%d %s\n", stack_i, history_stack[stack_i]);
            }
            continue;
        } else {
            for (int stack_i = MAX_HISTORY - 1; stack_i > 0; stack_i--)
{
                strcpy(history_stack[stack_i], history_stack[stack_i -
1]);
            }
            strcpy(history_stack[0], user_input);
        }

        pid_t pid = fork();
        if (pid < 0){
            perror("fork()");
            return 1;
        } else if (pid == 0) {
            char_count = 0;
            char *argv[MAX_LINE/2 - 1];
            char *p;
            p = strtok(user_input, " ");
            argv[char_count++] = p;
            while( p != NULL ) {
                p = strtok(NULL, " ");
                argv[char_count++] = p;
            }
            execvp(argv[0], argv);
            exit(0);
        }
        else wait(NULL);
    }
```

```
    return 0;
}
```

command:



```
yuwei@ubuntu:~/Desktop$ gcc -o myshell myshell.c
yuwei@ubuntu:~/Desktop$ ./myshell
-osh>>pwd
/home/yuwei/Desktop
-osh>>git --version
git version 2.17.1
-osh>>gcc --version
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

-osh>>ls
111-2-Operating-System  myshell  myshell.c
-osh>>history
0 ls
1 gcc --version
2 git --version
3 pwd
4
-osh>>exit
yuwei@ubuntu:~/Desktop$ 
```