Operating System

22F-3168

Mariam Fatima

Lab 7

Task 1:

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

#include <linux/kernel.h>

#include <linux/list.h>

#include <linux/slab.h>

#include <linux/init.h>



struct color {

   int red;

   int green;

   int blue;

   struct list_head list;

};



static LIST_HEAD(color_list);



static int __init color_list_init(void) {

   struct color *violet, *green, *yellow, *blue;

   struct color *ptr;
```

```
violet = kmalloc(sizeof(*violet), GFP_KERNEL);
violet->red = 135;
violet->green = 45;
violet->blue = 225;
INIT_LIST_HEAD(&violet->list);
list_add_tail(&violet->list, &color_list);

green = kmalloc(sizeof(*green), GFP_KERNEL);
green->red = 0;
green->green = 255;
green->blue = 0;
INIT_LIST_HEAD(&green->list);
list_add_tail(&green->list, &color_list);

yellow = kmalloc(sizeof(*yellow), GFP_KERNEL);
yellow->red = 255;
yellow->green = 255;
yellow->blue = 0;
INIT_LIST_HEAD(&yellow->list);
list_add_tail(&yellow->list, &color_list);

blue = kmalloc(sizeof(*blue), GFP_KERNEL);
blue->red = 0;
blue->green = 0;
blue->blue = 255;
INIT_LIST_HEAD(&blue->list);
list_add_tail(&blue->list, &color_list);
```

```
    printk(KERN_INFO "Color list created and initialized.\n");



    list_for_each_entry(ptr, &color_list, list) {
        printk(KERN_INFO "Color: Red = %d, Green = %d, Blue = %d\n", ptr->red, ptr->green, ptr->blue);
    }

    return 0;
}

static void __exit color_list_exit(void) {
    struct color *ptr, *next;

    list_for_each_entry_safe(ptr, next, &color_list, list) {
        list_del(&ptr->list);
        kfree(ptr);
    }

    printk(KERN_INFO "Color list removed and memory has been freed.\n");
}

module_init(color_list_init);
module_exit(color_list_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("uname");
MODULE_DESCRIPTION("Linux kernel module for color list");
MODULE_VERSION("0.1");
```

By using dmesg command I got this output.

```
[ 6083.474934] Color list created and initialized.
[ 6083.474934] Color: Red = 135, Green = 45, Blue = 225
[ 6083.474935] Color: Red = 0, Green = 255, Blue = 0
[ 6083.474935] Color: Red = 255, Green = 255, Blue = 0
[ 6083.474935] Color: Red = 0, Green = 0, Blue = 255
```

Task 2(Zombie Task):

```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>


int main()

{

   pid_t pid;

   pid = fork();


   if (pid < 0) {

      perror("Fork failed");

      exit(1);

   }

   if (pid == 0) {

      printf("Child process (PID: %d) is running...\n", getpid());

      sleep(3);

      printf("Child process (PID: %d) is exiting...\n", getpid());

      exit(0);

   }

   else {

      printf("Parent process with PID: %d  created child of PID: %d \n", getpid(), pid);

      printf("Parent process with PID: %d  is going to sleep for 15 seconds...\n", getpid());
```

```
        sleep(15);
        printf("Parent process with PID: %d is waking up...\n", getpid());


        printf("Parent process with PID: %d is exiting...\n", getpid());
    }
    return 0;
}
```