Project 1 Report  
CSC 4320/6320 - Operating Systems  
Spring 2023  
Name: Blake Caraballo  
Email: bcaraballo1@student.gsu.edu

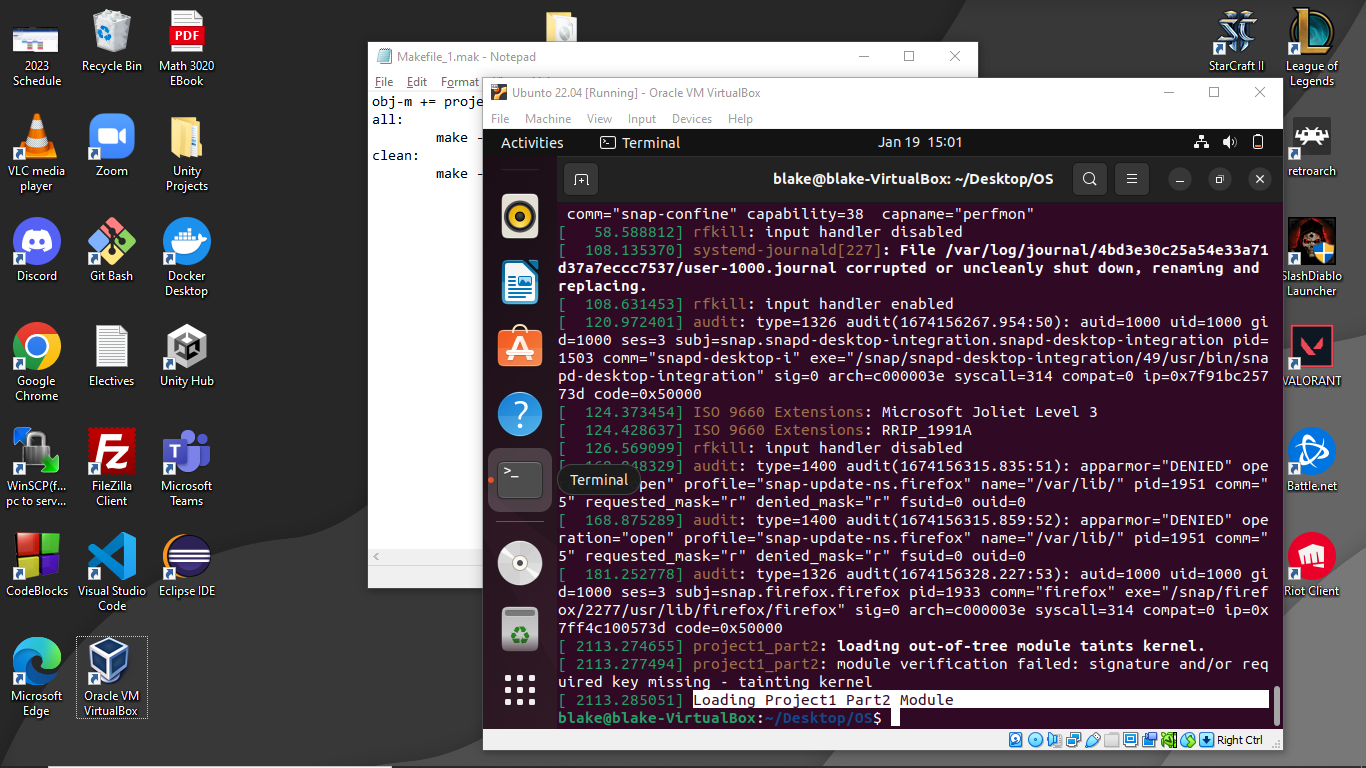
Part 1:  
1) Screenshots of your installed Ubuntu in VirtualBox.

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

Part 2:  
1) Kernel log buffer contents after loading kernel module project1\_part2  


2) Kernel log buffer contents after removing kernel module project1\_part2.  
A screenshot of a computer

Description automatically generated with medium confidence

Part 3:  
1) Kernel log buffer contents after loading kernel module project1\_part3.

Text

Description automatically generated

2) Kernel log buffer contents after removing kernel module project1\_part3.



#include <linux/init.h>

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/list.h>

#include <linux/slab.h>

struct birthday

{

    char \*name;

    int month;

    int day;

    int year;

    struct list\_head list; //next pointer?

};

/\*\*

 \* The following defines and initializes a list\_head object named birthday\_list

 \*/

static LIST\_HEAD(birthday\_list); //head node

int part3\_init(void)

{

    printk(KERN\_INFO "\*\*\*Loading Proejct1 Part3 Module\*\*\*\n");

    /\* Create a linked list containing five struct birthday elements\*/

    /\* NOTE:THE NAME OF FIRST STRUCT BIRTHDAY SHOULD BE YOUR OWN NAME\*/

    struct birthday \*person;

    struct birthday \*person2;

    struct birthday \*person3;

    struct birthday \*person4;

    struct birthday \*person5;

    person = kmalloc(sizeof(\*person), GFP\_KERNEL); //node

    person->day = 7;

    person->month = 3;

    person->year = 1993;

    person->name = "Blake Caraballo";

      INIT\_LIST\_HEAD(&person->list);

    person2 = kmalloc(sizeof(\*person2), GFP\_KERNEL); //node

    person2->day = 2;

    person2->month = 8;

    person2->year = 1989;

    person2->name = "John";

    INIT\_LIST\_HEAD(&person2->list);

    person3 = kmalloc(sizeof(\*person3), GFP\_KERNEL); //node

    person3->day = 1;

    person3->month = 1;

    person3->year = 2000;

    person3->name = "Gary";

    INIT\_LIST\_HEAD(&person3->list);

    person4 = kmalloc(sizeof(\*person4), GFP\_KERNEL); //node

    person4->day = 9;

    person4->month = 1;

    person4->year = 1967;

    person4->name = "Wilma";

    INIT\_LIST\_HEAD(&person4->list);

    person5 = kmalloc(sizeof(\*person5), GFP\_KERNEL); //node

    person5->day = 4;

    person5->month = 11;

    person5->year = 1999;

    person5->name = "Terry";

    INIT\_LIST\_HEAD(&person5->list);

    list\_add\_tail(&person->list, &birthday\_list);

    list\_add\_tail(&person2->list, &birthday\_list);

    list\_add\_tail(&person3->list, &birthday\_list);

    list\_add\_tail(&person4->list, &birthday\_list);

    list\_add\_tail(&person5->list, &birthday\_list);

    /\* Traverse the linked list  \*/

    struct birthday \*ptr;

    list\_for\_each\_entry(ptr, &birthday\_list, list){

        printk(KERN\_INFO "Name = %s Birthday: Month: %d Day: %d Year %d",ptr->name,ptr->month,ptr->day,ptr->year);

    }

    return 0;

}

void part3\_exit(void) {

    printk(KERN\_INFO "\*\*\*Removing Project1 Part3 Module\*\*\*\n");

    /\* Remove the elements from the linked list and return the free memory back to the kernel \*/

    struct birthday \*ptr, \*next;

     list\_for\_each\_entry\_safe(ptr,next,&birthday\_list,list){

        list\_del(&ptr->list);

         printk(KERN\_INFO "Removing: Name = %s Birthday: Month: %d Day: %d Year %d",ptr->name,ptr->month,ptr->day,ptr->year);

        kfree(next);

     }

}

module\_init( part3\_init );

module\_exit( part3\_exit );

MODULE\_LICENSE("GPL");

MODULE\_DESCRIPTION("Proejct1 Part3");

MODULE\_AUTHOR("GSU\_CSC4320\_6320\_TH\_Spring2023");