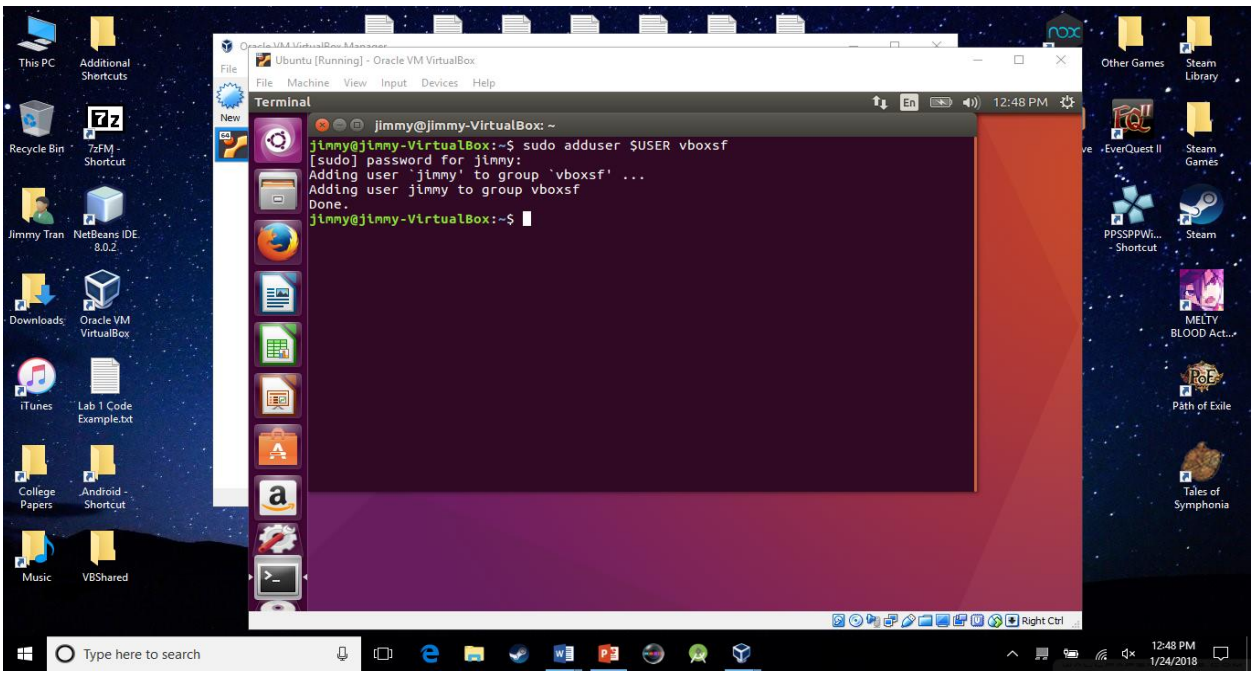
Project 1 Report
CSC 4320 Operating Systems
Spring 2018
Name: Jimmy Tran
Email: jtran25@student.gsu.edu

Part 1: Screenshot below



Part 2: Screenshot below for module loading and removing.

```
[   23.193050] audit: type=1400 audit(1516817041.096:4): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/sbin/dhclient"
 pid=520 comm="apparmor_parser"
[   23.193055] audit: type=1400 audit(1516817041.096:5): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/Networ
kManager/nm-dhcp-client.action" pid=520 comm="apparmor_parser"
[   23.193056] audit: type=1400 audit(1516817041.096:6): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/Networ
kManager/nm-dhcp-helper" pid=520 comm="apparmor_parser"
[   23.193058] audit: type=1400 audit(1516817041.096:7): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/connma
n/scripts/dhclient-script" pid=520 comm="apparmor_parser"
[   23.371435] audit: type=1400 audit(1516817041.272:8): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince
" pid=540 comm="apparmor_parser"
[   23.371439] audit: type=1400 audit(1516817041.272:9): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince
//sanitized_helper" pid=540 comm="apparmor_parser"
[   23.371441] audit: type=1400 audit(1516817041.272:10): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evinc
e-previewer" pid=540 comm="apparmor_parser"
[   23.371443] audit: type=1400 audit(1516817041.272:11): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evinc
e-previewer//sanitized_helper" pid=540 comm="apparmor_parser"
[   26.032929] Adding 522236k swap on /dev/sda5.  Priority:-1 extents:1 across:522236k FS
[   36.360538] vboxsf: Successfully loaded version 5.2.4 (interface 0x00010004)
[   37.454821] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[   37.459809] IPv6: ADDRCONF(NETDEV_UP): enp0s3: link is not ready
[   37.464419] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[   37.465054] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[   39.615654] VBoxService 5.2.4 r119785 (verbosity: 0) linux.amd64 (Dec 18 2017 16:48:59) release log
               00:00:00.000467 main     Log opened 2018-01-24T18:04:17.521810000Z
[   39.615978] 00:00:00.000727 main     OS Product: Linux
[   39.617040] 00:00:00.001726 main     OS Release: 4.10.0-28-generic
[   39.617147] 00:00:00.002032 main     OS Version: #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017
[   39.617313] 00:00:00.002136 main     Executable: /opt/VBoxGuestAdditions-5.2.4/sbin/VBoxService
               00:00:00.002138 main     Process ID: 852
               00:00:00.002140 main     Package type: LINUX_64BITS_GENERIC
[   39.619931] 00:00:00.004796 main     5.2.4 r119785 started. Verbose level = 0
[   39.691697] 00:00:00.076532 automount vbsvcAutoMountWorker: Shared folder 'VBShared' was mounted to '/media/sf_VBShared'
[   99.645955] ISO 9660 Extensions: Microsoft Joliet Level 3
[   99.699027] ISO 9660 Extensions: RRIP_1991A
[  970.171655] Loading Module
[ 1085.416687] Removing Module
jimmy@jimmy-VirtualBox:~/Documents$
```

Part 3:

1 and 2) Screenshot for module load and removal combined . I know the years don't make sense. Sue me for picking random numbers. It works in any case. The above loads and removes are there because I forgot the labels for the variables. Ignore those.

```
jimmy@jimmy-VirtualBox: ~/Documents/project1part2
[  898.218012] Loading: John 3 4 1001
[  898.218014] Loading: Jack 5 6 1002
[  898.218016] Loading: Jill 7 8 1003
[  898.218018] Loading: Joe 9 10 1004
[  921.131852] Removing Module
[  921.131857] Removing: Jimmy Tran 1 2 1000
[  921.131859] Removing: John 3 4 1001
[  921.131860] Removing: Jack 5 6 1002
[  921.131861] Removing: Jill 7 8 1003
[  921.131862] Removing: Joe 9 10 1004
[  935.531610] sched: RT throttling activated
[ 1136.317769] Loading Module
[ 1136.317773] Name: Jimmy Tran Birthday: Month 1 Day 2 Year 1000
[ 1136.317775] Name: John Birthday: Month 3 Day 4 Year 1001
[ 1136.317776] Name: Jack Birthday: Month 5 Day 6 Year 1002
[ 1136.317777] Name: Jill Birthday: Month 7 Day 8 Year 1003
[ 1136.317778] Name: Joe Birthday: Month 9 Day 10 Year 1004
[ 1141.769027] Removing Module
[ 1141.769031] Name: Jimmy Tran Birthday: Month 1 Day 2 Year 1000
[ 1141.769033] Name: John Birthday: Month 3 Day 4 Year 1001
[ 1141.769034] Name: Jack Birthday: Month 5 Day 6 Year 1002
[ 1141.769035] Name: Jill Birthday: Month 7 Day 8 Year 1003
[ 1141.769036] Name: Joe Birthday: Month 9 Day 10 Year 1004
jimmy@jimmy-VirtualBox:~/Documents/project1part2$
```

3) Source Code:

```
#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;
};

/**
 * The following defines and initializes a list_head object named birthday_list
 */
static LIST_HEAD(birthday_list);

int simple_init(void)
{

	printk(KERN_INFO "Loading 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 *person1;
	struct birthday *person2;
```

```
struct birthday *person3;
struct birthday *person4;
struct birthday *person5;

//first element
person1=kmalloc(sizeof(*person1),GFP_KERNEL);
person1->name="Jimmy Tran";
person1->month=1;
person1->day=2;
person1->year=1000;
INIT_LIST_HEAD(&person1->list);

//second element
person2=kmalloc(sizeof(*person2),GFP_KERNEL);
person2->name="John";
person2->month=3;
person2->day=4;
person2->year=1001;
INIT_LIST_HEAD(&person2->list);

//third element
person3=kmalloc(sizeof(*person3),GFP_KERNEL);
person3->name="Jack";
```

```
person3->month=5;
person3->day=6;
person3->year=1002;
INIT_LIST_HEAD(&person3->list);

//fourth element
person4=kmalloc(sizeof(*person4),GFP_KERNEL);
person4->name="Jill";
person4->month=7;
person4->day=8;
person4->year=1003;
INIT_LIST_HEAD(&person4->list);

//fifth element
person5=kmalloc(sizeof(*person5),GFP_KERNEL);
person5->name="Joe";
person5->month=9;
person5->day=10;
person5->year=1004;
INIT_LIST_HEAD(&person5->list);

//build linked list
list_add_tail(&person1->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){
		/* on each iteration ptr points */
		/* to the next birthday struct */
		printk(KERN_INFO "Name: %s Birthday: Month %d Day %d Year %d\n",ptr-
>name,ptr->month,ptr->day,ptr->year);
	}

	return 0;
}

void simple_exit(void) {

	/* Remove the elements from the linked list and return the free memory back to the
kernel */
	printk(KERN_INFO "Removing Module\n");
```

```
	struct birthday *ptr,*next;
	list_for_each_entry_safe(ptr, next, &birthday_list,list){
		/* on each iteration ptr points */
		/* to the next birthday struct */
		printk(KERN_INFO "Name: %s Birthday: Month %d Day %d Year %d\n",ptr-
>name,ptr->month,ptr->day,ptr->year);
		list_del(&ptr->list);
		kfree(ptr);
	}

}

module_init( simple_init );
module_exit( simple_exit );

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Kernel Data Structures");
MODULE_AUTHOR("SGG");
```