Task I (70 points): Add system call monitoring to Linux kernel

Task I.1 (20 points): Determine where at entry.S to add new code

Between line 355 and 356. (Between 'syscall_call:' and 'call *sys_call_table(,%eax,4)')

```
syscall_call:
        pushl %eax
        call isa673_syscallmon
        popl %eax
        call *sys_call_table(,%eax,4)
        movl %eax,EAX(%esp)             # store the return value
```

Task I.2 (20 points): Determine what to add to entry.S

Add these three lines of code in entry.S:

```
pushl %eax
call isa673_syscallmon
popl %eax
```

Task I.3 (30 points): Modify isa673_syscallmon() defined in irq.c

The modified syscallmon function is written below:

```
asmlinkage void isa673_syscallmon(unsigned int syscallNo)
{

       printk(KERN_INFO "ISA673 System Call Monitor: pid %ld (%s):
       syscallNo=%4d\n",sys_getpid(),current->comm,syscallNo);

}
```

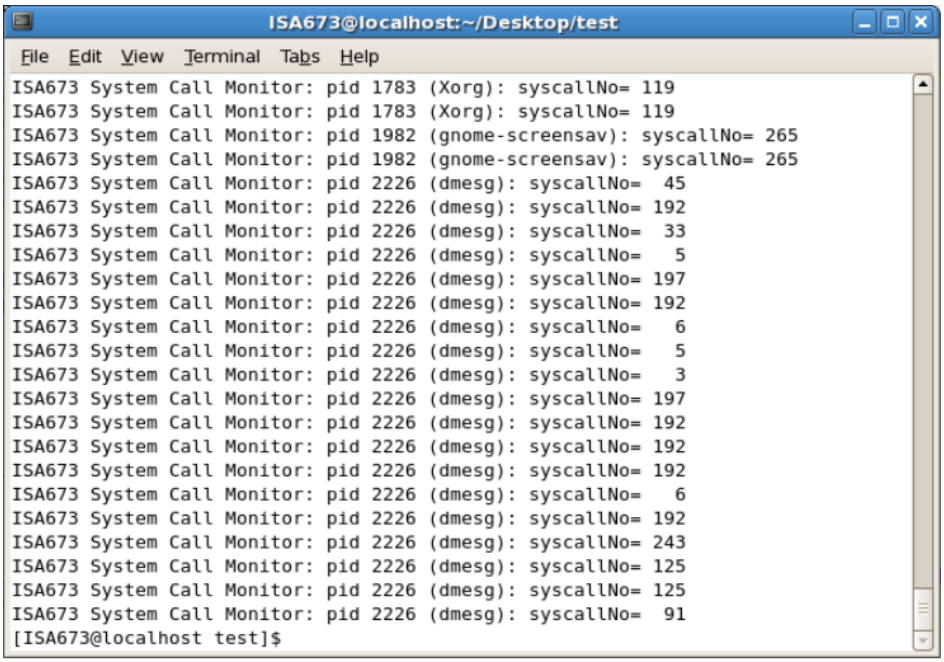
The screenshot of the code:

```
asmlinkage void isa673_syscallmon(unsigned int syscallNo)
{

        printk(KERN_INFO "ISA673 System Call Monitor: pid %ld (%s): syscallNo=%4d\n",sys_getpid
(),current->comm,syscallNo);

}
```

The screenshot after 'dmesg' command:



```
ISA673@localhost:~/Desktop/test
File  Edit  View  Terminal  Tabs  Help
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1982 (gnome-screensav): syscallNo= 265
ISA673 System Call Monitor: pid 1982 (gnome-screensav): syscallNo= 265
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=  45
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=  33
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=   5
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 197
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=   6
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=   5
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=   3
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 197
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=   6
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 192
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 243
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 125
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo= 125
ISA673 System Call Monitor: pid 2226 (dmesg): syscallNo=  91
[ISA673@localhost test]$
```

Task II (30 points): Add dynamic control on the system call monitoring:

First step in our process is to add these two lines at the top of 'irq.c' file, just before the syscallmon function:

```
int isa673_printk=0;
EXPORT_SYMBOL(isa673_printk);
```

Once this is done, add an if condition inside syscallmon function like the code snippet below:

```
//ISA
int isa673_printk=0;
EXPORT_SYMBOL(isa673_printk);
asmlinkage void isa673_syscallmon(unsigned int syscallNo)
{
        if(isa673_printk!=0)
        {
        printk(KERN_INFO "ISA673 System Call Monitor: pid %ld (%s): syscallNo=%4d\n",sys_getpid
 (),current->comm,syscallNo);
        }
}
//ISA
```

Second step of our process is to create a loadable kernel module. Now, create a C file name isa_lkm.c and add following code in the file.

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

extern int isa673_printk;
int init_module(void)
{

        printk(KERN_INFO"Syscallmon is switched on\n");
        isa673_printk=1;
        return 0;
}

void cleanup_module(void)
{
        printk(KERN_INFO"Syscallmon is switched off\n");
        isa673_printk=0;
}
```

Now, create a 'Makefile' using the code below:

```
obj-m += isa_lkm.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
```

Once both files are created (Both Makefile and isa_lkm.c file are attached in the submission zip file) , open the directory where these files are placed in a terminal and write the following command in the terminal:

```
make
```

Once these steps are done, we just need to load our module. Now, I faced a specific problem where insmod and rmmod commands were not working. If you face a similar problem, open your terminal and write the following command:

```
export PATH=$PATH:/sbin
```

Now, once this is resolved we can move onto the loading the module. In order to load your module, write the command:

```
sudo insmod isa_lkm.ko
```

once you do that you'll get following output:

```
Syscallmon is switched on
ISA673 System Call Monitor: pid 1987 (bash): syscallNo= 119
ISA673 System Call Monitor: pid 363 (udevd): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1982 (gnome-screensav): syscallNo= 265
ISA673 System Call Monitor: pid 1982 (gnome-screensav): syscallNo= 265
ISA673 System Call Monitor: pid 1982 (gnome-screensav): syscallNo= 265
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
ISA673 System Call Monitor: pid 1783 (Xorg): syscallNo= 119
```

In order to remove your module, write the following command:

```
sudo rmmod isa_lkm.ko
```

and you'll get following output:

```
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 192
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 192
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo=   6
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 192
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 243
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 125
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo= 125
ISA673 System Call Monitor: pid 27216 (rmmod): syscallNo=  91
Syscallmon is switched off
[ISA673@localhost test]$
```