

Operating Systems Programming Assignment 1

Liz Parker - elpa8934@colorado.edu

arch/x86/kernel/cs3753_add.c

The c code for my custom system call can be found at this path within the kernel space

```
cd /home/kernel/linux-hwe-4.15.0/arch/x86/kernel
```

```
sudo vim cs3753_add.c
```

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

asmlinkage long sys_cs3753_add(int number1, int number2, int *resultLoc)
{
    printk(KERN_ALERT "First number: %d\n", number1);
    printk(KERN_ALERT "Second number: %d\n", number2);
    *resultLoc = number1 + number2;
    printk(KERN_ALERT "Added result: %d\n", *resultLoc);
    return 0;
}
```

arch/x86/kernel/Makefile

The makefile can be found at this path within the kernel space

```
cd /home/kernel/linux-hwe-4.15.0/arch/x86/kernel
```

```
sudo vim Makefile
```

The Makefile is very large, but my changes can be seen in the bottom two lines

```
obj-$(CONFIG_UNWINDER_ORC) += unwind_orc.o
obj-$(CONFIG_UNWINDER_FRAME_POINTER) += unwind_frame.o
obj-$(CONFIG_UNWINDER_GUESS) += unwind_guess.o
obj-y+=helloworld.o
obj-y+=cs3753_add.o
```

arch/x86/entry/syscalls/syscall_64.tbl

The syscall_64.tbl can be found at this path within the kernel space

```
cd /home/kernel/linux-hwe-4.15.0/arch/x86/entry/syscalls
```

```
sudo vim syscall_64.tbl
```

Again, the table is very large, but numbers 333 and 334 are my modifications

330	common	pkey_alloc	sys_pkey_alloc
331	common	pkey_free	sys_pkey_free
332	common	statx	sys_statx
333	64	helloworld	sys_helloworld
334	64	cs3753_add	sys_cs3753_add

include/linux/syscalls.h

The syscalls.h can be found at this path within the kernel space

```
cd /home/kernel/linux-hwe-4.15.0/include/linux
```

```
sudo vim syscalls.h
```

Again, the header file is very large, but the last two lines are my modifications

```
asmlinkage long sys_pkey_free(int pkey);
asmlinkage long sys_statx(int dfd, const char __user *path, unsigned flags,
                          unsigned mask, struct statx __user *buffer);
asmlinkage long sys_helloworld(void);
asmlinkage long sys_cs3753_add(int, int, int *);
#endif
```

/var/log/syslog

The syslog can be found at this path within the user space

```
cd /home/user
```

```
sudo tail /var/log/syslog
```

After running my testAdd.c function, the syslog shows the printf outputs from my syscall

```
Jan 29 17:08:23 cu-cs-vm blueman-mechanism: loading Network
Jan 29 17:08:34 cu-cs-vm kernel: [ 69.957161] First number: -1
Jan 29 17:08:34 cu-cs-vm kernel: [ 69.957167] Second number: 5
Jan 29 17:08:34 cu-cs-vm kernel: [ 69.957169] Added result: 4
Jan 29 17:08:53 cu-cs-vm blueman-mechanism: Exiting
```

Source code for my test program - testAdd.c

My test program can be found in the home directory of the user space

```
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>
int main()
{
    int num1 = -1;
    int num2 = 5;
    int *result;
    long int test = syscall(334, num1, num2, result);
    printf("The result from test function is: %d\n", *result);
    return 0;
}
```

It can be compiled using the command: `gcc testAdd.c`

And run using the command: `./a.out`

```
user@cu-cs-vm:~$ ls
a.out      Desktop  Downloads  Pictures  Templates  testProgram.c
cu-cs-apt.deb Documents Music      Public    testAdd.c  Videos
user@cu-cs-vm:~$ gcc testAdd.c
user@cu-cs-vm:~$ ./a.out
The result from test function is: 1
```

After running, the printk outputs can be seen in syslog or from the dmesg command

```
[ 69.957161] First number: -1
[ 69.957167] Second number: 5
[ 69.957169] Added result: 4
```