#### Operating system experiment report

1. **Name:** Luoyu Mei **Number:** 71117408 **Date:** 02/04/2019
2. **Working target:** Adding a system-call into a open source kernel and change it into Ubuntu system. After that, write a program to call the kernel.

**Working environment:** as basic operating system with virtual machine and running on it. The kernel I use is .

1. **Steps:**
2. Write a system-call function into the file, which is in the folder



1. Add a statement into the system-call registry , which is in the folder. In the kernel I use, the registry number is 548 since there already have 547 entry. Because I use a 64bit machine, so the registry is:
2. Make a adjunction of system-call statement into , which is in the folder



1. After that, I go into the root folder and begin **sudo make -j64**, I used 64 threads to make the compile faster.
2. After the compiling and new kernel installing, I got a new system which have a system-call interface I had just written. Then I wrote a system-call test program to test it.

****

1. **Flow chart:**

Write System call function into the file,

Add a statement into the system-call registry

Make a adjunction of system-call statement into

Go into the root folder and begin **sudo make -j64**

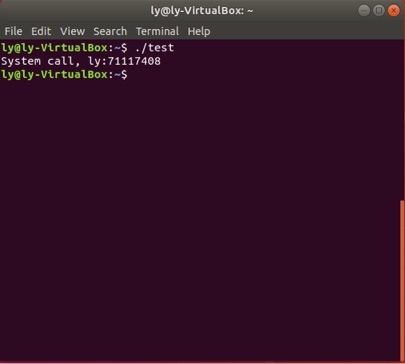
Wrote a system-call test program to test the kernel.

1. **Main data structure:**

As a result of using Linux kernel, I only need to use long as a representation of the return value of my function. I make my function code return school number 7117408.

1. **Experiment result:**

The kernel successfully returned my school number: 71117408.



1. **Filling of experiment:**

To achieve in adding system call function, I need to write a function into the system main file, and make sure I remember the number of it. Then I add the number into system call registry. Then I also add a statement into system call header file. After doing all of this, I began running the kernel to replace my old one. Use multi thread in code running make it faster, however it took more main memory to remember threads.