

Some items are just check-points, other items have written answers. Refer to the "A Simple System Call" assignment.

1. Who is the owner of these two files? syslog
2. What is the group assignment of these two files? adm
3. Try to access these files.
 - a. Can ubuntu (as a normal user) use the cat command to display the contents of these files? Why?
 - b. Log into spock, repeat 3 again. Can you cd into /var/log or cat /var/log/syslog or kern.log?

a. yes, I am a member of the adm group which has read and write permissions

b. yes, I can cd into /var/log. No, I can't cat syslog or kern.log
4. What is the include file needed for the **read** and **write** system call? unistd.h.

5. Assuming that you have relocated to the /usr/include directory. What **find/grep** command combo would locate the unistd_32.h file inside the /usr/include directory? Use your find/grep command to find the full path to the unistd_32.h file, what is the path?

find . -print | grep unistd*

/usr/include/x86_64-linux-gnu/asm/unistd_32.h

6. Once you locate unistd_32.h and unistd_64.h, using an editor (vi) to look through the file to identify the number associated with the following system calls in 32 and 64 bit linux.

	32-bit	64-bit
read	<u>3</u>	<u>0</u>
write	<u>4</u>	<u>1</u>
open	<u>5</u>	<u>2</u>
close	<u>6</u>	<u>3</u>
mkdir	<u>39</u>	<u>83</u>
fork	<u>2</u>	<u>57</u>
exit	<u>1</u>	<u>60</u>
dup	<u>41</u>	<u>32</u>
pipe	<u>42</u>	<u>22</u>

7. Check the man pages on write (remember, man write is not the system call write, use man 2 write to get the correct man page). Using the 64-bit version, **what** are the arguments (parameters) for the write system call and **what** registers should the arguments be placed?

fd, buf, count

rdi, rsi, rdx

8/9. scp your code, hello.s and io.s, to your laptop and get a **nice clean printout**. **Attach** to this document. I'll do a **visual check** on the two programs on your machine, during the class, on the due date of assignment.

___ 10. From the base directory, search through the files in the **arch** subdirectory for the file named syscall_64.tbl. That should get the directory (location) of the syscall_64.tbl. What find command will help you find the file? What is the directory?

find . -print | grep syscall_64.tbl (fill in the blank with the find command used)

cd arch/x86/entry/syscalls/syscall_64.tbl (fill in the blank with the directory found above)

11. What is the first number that follows the last 400 number in the file? 451

✓ 12. From the base director – remember the base directory is as noted at the top of this page,
cd include/linux
edit syscalls.h

✓ 13. From the base directory
cd kernel
edit the Makefile

✓ 14. From the base directory,
cd kernel
create the file named helloworld.c

✓ 15. Create the system call test program as a normal user.

16. Run the code. What should happen? Look back at code for the system call, Any output to screen? Did it run?

The user level program calls the system call 451 which uses printk to output "hello world" to the kernel log.
No output on screen, yes it ran

17. Where should I look to find out if the system call did it's job?

in the kernel log using dmesg

During class I will ask you to show me your system call works. I should find three programs in Ubuntu's home directory. hello.s, io.s, and hello_SCT.c