Write up

Writing a System Call

These are the steps I followed while making a system call.
1.The first step is to download the kernel source from the below command.
wget https://www.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz

2.Then extract the kernel source code in specify directory.

```
sudo tar -xvf linux-5.9.1.tar.xz -C/usr/src/
```

3. Now, change the directory to where the files are extracted:

```
cd /usr/src/linux-5.9.1/
```

4. Add the new system call to the system call table:
creating a system call requires editing a table that is included by a truly huge amount of code. The file containing the system call table for x86_64 is located in arch/x86/entry/syscalls/syscall_64.tbl. The PID for the new system call has to be added. Mine is added at 548.

5.The last step is to write the function for the system call.
You can implement system calls anywhere, but miscellaneous syscalls tend to go in the kernel/sys.c file.
So, go to the end of the sys.c file, and define a function sh_task_info that takes two arguments first is the PID, and the second one is the file name by using SYSCALL_DEFINEN, a macroscopic family that makes it easy to define a system call with N arguments. As we cannot directly use the filename pointer because the process could try to trick us into printing out data from kernel memory by giving us a pointer that maps to kernel space or the process could try to read another process's memory by giving a pointer that maps into another

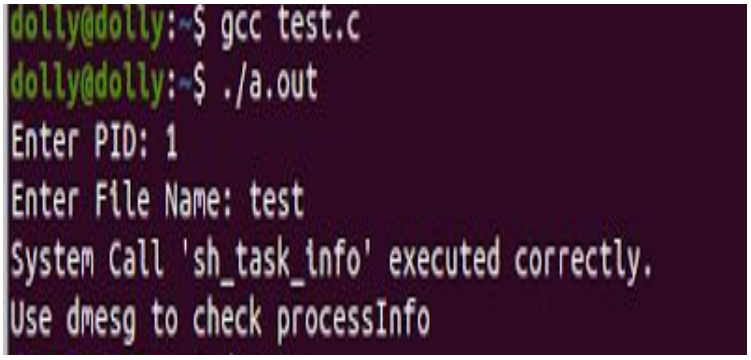
process's address space. So to resolve this, I used the strncpy_from_user() function, which behaves like normal strncpy but checks the user-space memory address first. If the string was too long or if there was a problem copying, it returns EFAULT. Then a task_struct pointer task and struct file pointer file are created. For each task, it's attributes are printed on the console using printk() and simultaneously written to the file. file will be open by using function flip_open() and write by kernel_write().

6. finally, Compile and boot the kernel:
Before starting to compile, I installed a few packages.
sudo apt-get install gcc
sudo apt-get install libncurses5-dev
sudo apt-get install bison
sudo apt-get install flex
sudo apt-get install libssl-dev
sudo apt-get install libelf-dev
sudo apt-get update
sudo apt-get upgrade
Now, to configure the kernel, I use the following command in linux-5.9.1/ directory.
sudo make menuconfig
Once the above command is used to configure the Linux kernel, there will be a pop-up window in which I deselected the network wireless setting and save the configuration, So that it will decrease the compilation time.
Now to compile the kernel, I used the following command:
sudo make -j8 && sudo make modules_install install && shutdown -r now.

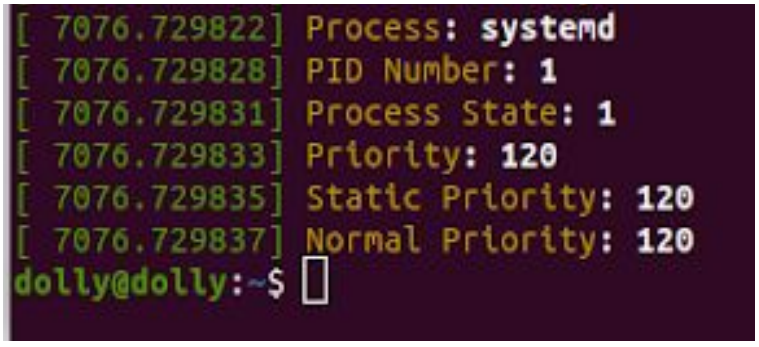
Testing it with sample code

1.The file test.c is the sample test code, compares the process IDs by using syscall. If the input process ID and the updated PID

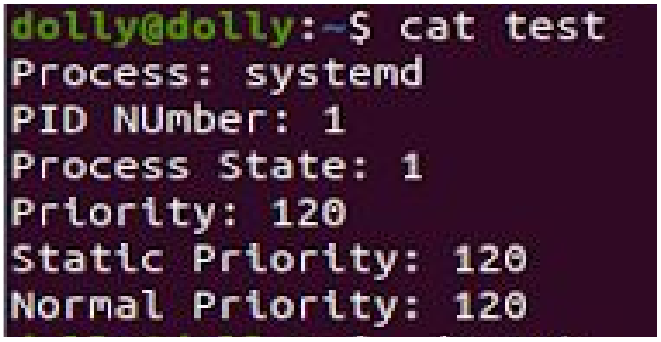
548 match, there is no error in the given PID and the filename. the sh_task_info() will execute correctly.
2.Then, the provided makefile should run it.



3.If it is executed successfully; we can check the log via 'dmesg'.
4.The process, the process ID, process state, priority, static priority, and normal priority are printed.



5.Alternatively, we can check the log with 'cat <filename>'.



Errors Handled

1.If the user enters an invalid PID such as a char or float, the function returns the errno 22 EINVAL invalid argument.

```
Enter PID: pid
Enter File Name: System call sh_task_info did not execute as expected
Error : Invalid argument
Error No.: 22
```

2.If the user enters PID <=0 or greater than 4194304
, the function returns the errno 22 EINVAL invalid argument.
3.If the inputted PID does not exist, then the function prints a message that “PID doesn’t exist” and which can easily be seen by using dmesg command.

```
Enter PID: 0
Enter File Name: test
System call sh_task_info did not execute as expected
Error : Invalid argument
Error No.: 22
dolly@dolly:~$ ./a.out
Enter PID: 588899
Enter File Name: test
System call sh_task_info did not execute as expected
Error : Bad address
Error No.: 14
dolly@dolly:~$ ./a.out
Enter PID: 6778906
Enter File Name: test
System call sh_task_info did not execute as expected
Error : Invalid argument
Error No.: 22
```

4.If the flip_open() for creating a file if it doesn't exist, with write access returns NULL; the function returns the error EISDIR.