Operating Systems Assignment 2 Question 2

Yatharth Taneja | 2019346

Process

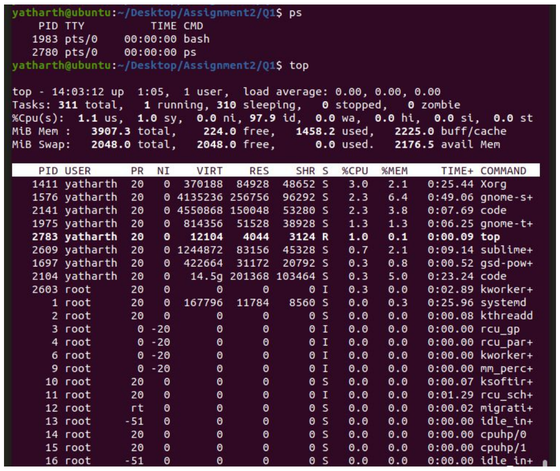
1. **Installing and compiling the kernel v5.9.1**
   - 1.1 wget <link> to download the file from kernel.org
   - 1.2 Installing the downloaded file to /usr/src/ using tar -xvz <file name> -C/usr/src/
   - 1.3 installing necessary files and libraries like gcc, bison flex and use
     sudo apt-get update
     sudo apt-get upgrade
   - 1.4 Make .config file using make menu config
   - 1.5 Compile and install the kernel and modules using
     make -j4
     make modules_install install
   - 1.6 reboot
2. **Writing the syscall**
   - 2.1. Make a directory task_info in /usr/src
   - 2.2. Create a file sh_task_info.c in this directory
   - 2.3. Make a Makefile for the same
     This is to make sure file is compiled and object code is created
   - 2.4. Go to the /usr/src directory and edit it's makefile to add /task_info to the second occurrence of core-y:
     This is to tell the compiler that the source files of our new system call (sys_sh_task_struct()) are in present in the task_info directory.
   - 2.5. include the syscall in system call table present in arch/x86/entry/syscalls/
   - 2.6 include the syscall in system call header file
   - 2.7 compile the kernel again.

Description of code

1. **sh_task_info.c file**
   - 1.1. Here we will have a pid as input therefore we will make a **task_struct** and assign values to it using the **find_task_by_vpid()**
   - 1.2. Then we will get the data and allocate it to a buffer data using **snprintf()** function
   - 1.3. This data can be printed on kernel log using **printk()** function
   - 1.4. To write to a file we will first save the current segment and do a file open using **flip_open()**
   - 1.5. We have the file_path as an argument we will copy it to a buffer using strcpy_from_user and write using **kernel_write()** with this buffer and data, data size as arguments.
   - 1.6. At every important point, error handling is done.
2. **test.c file**
   - 2.1. It calls the defined syscall sh_task_info and assigns a return value to a variable.
   - 2.2. The syscall returns 0 for a successful run and -1 for errors

2.3. If there are any errors, the error is printed using strerror(errno) since errno is automatically assigned.

The inputs User should give

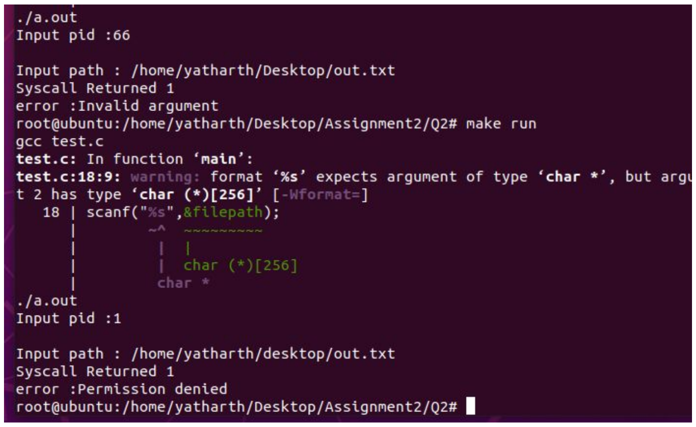


```
yatharth@ubuntu:~/Desktop/Assignment2/Q1$ ps
    PID TTY          TIME CMD
   1983 pts/0    00:00:00 bash
   2780 pts/0    00:00:00 ps
yatharth@ubuntu:~/Desktop/Assignment2/Q1$ top

top - 14:03:12 up  1:05,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 311 total,   1 running, 310 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.1 us,  1.0 sy,  0.0 ni, 97.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3907.3 total,    224.0 free,   1458.2 used,   2225.0 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   2176.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1411 yatharth  20   0  370188  84928  48652 S   3.0   2.1   0:25.44 Xorg
   1576 yatharth  20   0 4135236 256756  96292 S   2.3   6.4   0:49.06 gnome-s+
   2141 yatharth  20   0 4550868 150048  53280 S   2.3   3.8   0:07.69 code
   1975 yatharth  20   0  814356  51528  38928 S   1.3   1.3   0:06.25 gnome-t+
   2783 yatharth  20   0   12104   4044   3124 R   1.0   0.1   0:00.09 top
   2609 yatharth  20   0 1244872  83156  45328 S   0.7   2.1   0:09.14 sublime+
   1697 yatharth  20   0  422664  31172  20792 S   0.3   0.8   0:00.52 gsd-pow+
   2104 yatharth  20   0   14.5g 201368 103464 S   0.3   5.0   0:23.24 code
   2603 root      20   0       0      0      0 I   0.3   0.0   0:02.89 kworker+
      1 root      20   0  167796  11784   8560 S   0.0   0.3   0:25.96 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.08 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.07 ksoftir+
     11 root      20   0       0      0      0 I   0.0   0.0   0:01.29 rcu_sch+
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migrati+
     13 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
```

The user should be able to run the program by using make run.

The user should give the pid and filepath as input in test.c and depending on the task whether it is running or not we will get the output.
We can check the process running either by ps or top command

Output



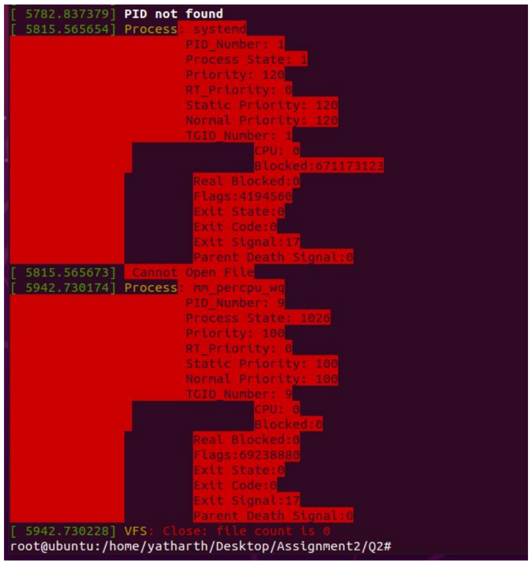
```
./a.out
Input pid :66

Input path : /home/yatharth/Desktop/out.txt
Syscall Returned 1
error :Invalid argument
root@ubuntu:/home/yatharth/Desktop/Assignment2/Q2# make run
gcc test.c
test.c: In function ‘main’:
test.c:18:9: warning: format ‘%s’ expects argument of type ‘char *’, but argu
t 2 has type ‘char (*)[256]’ [-Wformat=]
   18 |  scanf("%s",&filepath);
      |         ~^  ~~~~~~~~~
      |          |  |
      |          |  char (*)[256]
      |          char *
./a.out
Input pid :1

Input path : /home/yatharth/desktop/out.txt
Syscall Returned 1
error :Permission denied
root@ubuntu:/home/yatharth/Desktop/Assignment2/Q2# 
```

If the pid is correct and syscall returns 0 and success message is printed

Else if there is an error sys call returns 1 along withe the error type

Here Invalid Argument is shown since 66 pid is does not exist

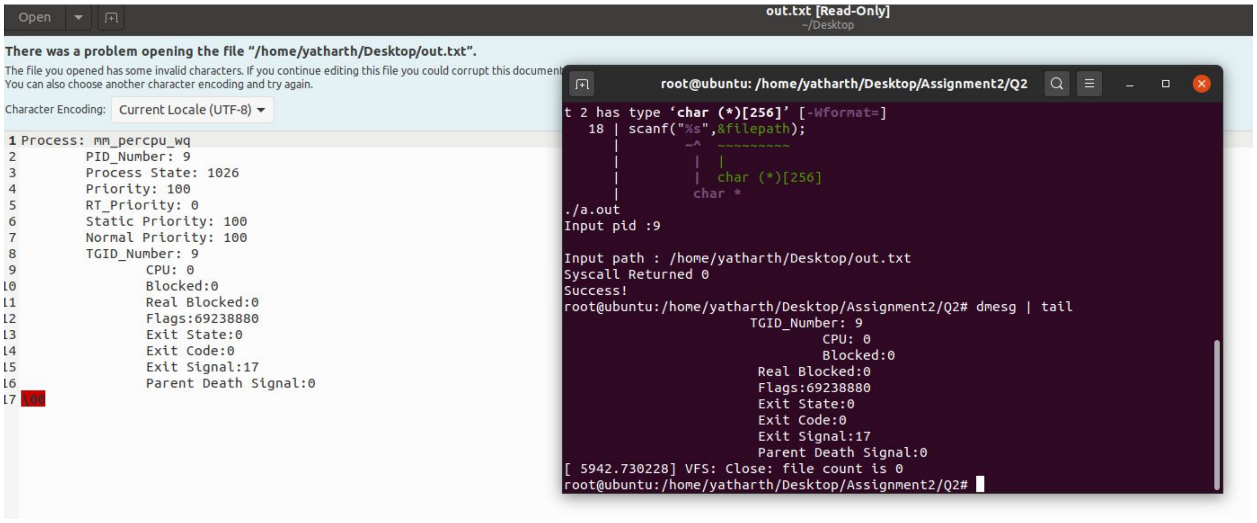
And Permission denied is printed since I sent a wrong address. (desktop instead of Desktop)



```
[ 5782.837379] PID not found
[ 5815.565654] Process: systemd
                PID_Number: 1
                Process State: 1
                Priority: 120
                RT_Priority: 0
                Static Priority: 120
                Normal Priority: 120
                TGID Number: 1
                         CPU: 0
                         Blocked:671173123
                 Real Blocked:0
                 Flags:4194560
                 Exit State:0
                 Exit Code:0
                 Exit Signal:17
                 Parent Death Signal:0
[ 5815.565673]  Cannot Open File
[ 5942.730174] Process: mm_percpu_wq
                PID_Number: 9
                Process State: 1026
                Priority: 100
                RT_Priority: 0
                Static Priority: 100
                Normal Priority: 100
                TGID Number: 9
                         CPU: 0
                         Blocked:0
                 Real Blocked:0
                 Flags:69238880
                 Exit State:0
                 Exit Code:0
                 Exit Signal:17
                 Parent Death Signal:0
[ 5942.730228] VFS: Close: file count is 0
root@ubuntu:/home/yatharth/Desktop/Assignment2/Q2#
```

This is the Output in the kernel log.

1. PID not found is for the Invalid argument
2. Cannot open is for the wrong address sent ,
(task_struct info is printed)
3. Log of the file successfully executed and
closed



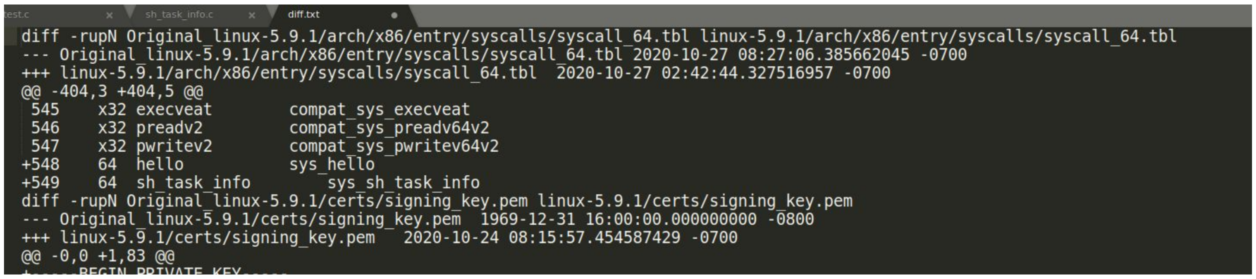
```
There was a problem opening the file "/home/yatharth/Desktop/out.txt".
The file you opened has some invalid characters. If you continue editing this file you could corrupt this document
You can also choose another character encoding and try again.
Character Encoding: Current Locale (UTF-8)

Process: mm_percpu_wq
        PID_Number: 9
        Process State: 1026
        Priority: 100
        RT_Priority: 0
        Static Priority: 100
        Normal Priority: 100
        TGID_Number: 9
                CPU: 0
                Blocked:0
                Real Blocked:0
                Flags:69238880
                Exit State:0
                Exit Code:0
                Exit Signal:17
                Parent Death Signal:0
\00
```

```
root@ubuntu: /home/yatharth/Desktop/Assignment2/Q2
t 2 has type 'char (*)[256]' [-Wformat=]
   18 | scanf("%s",&filepath);
      |         ~^ ~~~~~~~~~
      |          |  |
      |          |  char (*)[256]
      |          char *
./a.out
Input pid :9

Input path : /home/yatharth/Desktop/out.txt
Syscall Returned 0
Success!
root@ubuntu:/home/yatharth/Desktop/Assignment2/Q2# dmesg | tail
                        TGID_Number: 9
                                 CPU: 0
                                 Blocked:0
                         Real Blocked:0
                         Flags:69238880
                         Exit State:0
                         Exit Code:0
                         Exit Signal:17
                         Parent Death Signal:0
[ 5942.730228] VFS: Close: file count is 0
root@ubuntu:/home/yatharth/Desktop/Assignment2/Q2#
```

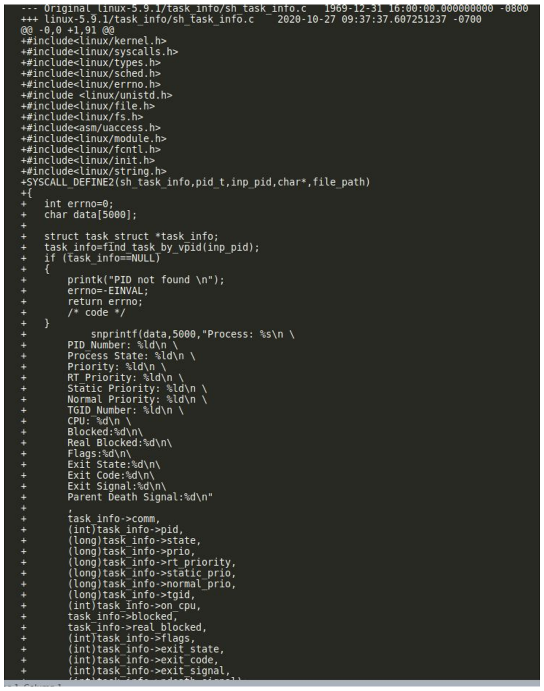
This is the combined output for the input pid 9 , dmesg | tail is used to print the last lines and on the left is the text file generated.

Note: You have to be in sudo -s mode for the file to be created

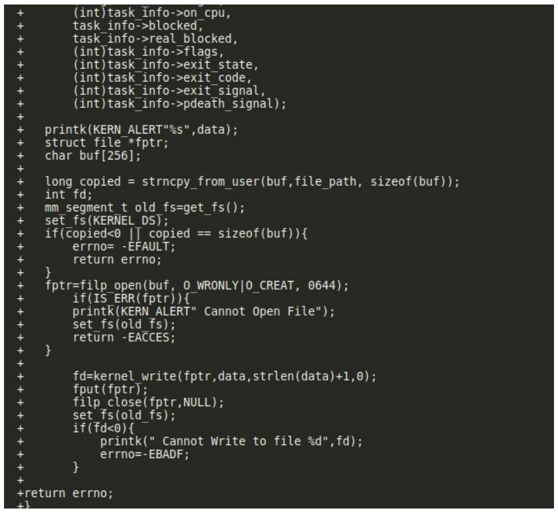
diff.txt These are the important changes shown in the patch file.



```
diff -rupN Original_linux-5.9.1/arch/x86/entry/syscalls/syscall_64.tbl linux-5.9.1/arch/x86/entry/syscalls/syscall_64.tbl
--- Original_linux-5.9.1/arch/x86/entry/syscalls/syscall_64.tbl 2020-10-27 08:27:06.385662045 -0700
+++ linux-5.9.1/arch/x86/entry/syscalls/syscall_64.tbl  2020-10-27 02:42:44.327516957 -0700
@@ -404,3 +404,5 @@
 545    x32 execveat        compat_sys_execveat
 546    x32 preadv2         compat_sys_preadv64v2
 547    x32 pwritev2        compat_sys_pwritev64v2
+548    64  hello           sys_hello
+549    64  sh_task_info        sys_sh_task_info
diff -rupN Original_linux-5.9.1/certs/signing_key.pem linux-5.9.1/certs/signing_key.pem
--- Original_linux-5.9.1/certs/signing_key.pem  1969-12-31 16:00:00.000000000 -0800
+++ linux-5.9.1/certs/signing_key.pem   2020-10-24 08:15:57.454587429 -0700
@@ -0,0 +1,83 @@
```



```
--- Original_linux-5.9.1/task_info/sh_task_info.c   1969-12-31 16:00:00.000000000 -0800
+++ linux-5.9.1/task_info/sh_task_info.c    2020-10-27 09:37:37.607251237 -0700
@@ -0,0 +1,91 @@
+#include<linux/kernel.h>
+#include<linux/syscalls.h>
+#include<linux/types.h>
+#include<linux/sched.h>
+#include<linux/errno.h>
+#include <linux/unistd.h>
+#include<linux/file.h>
+#include<linux/fs.h>
+#include<asm/uaccess.h>
+#include<linux/module.h>
+#include<linux/fcntl.h>
+#include<linux/init.h>
+#include<linux/string.h>
+SYSCALL_DEFINE2(sh_task_info,pid_t,inp_pid,char*,file_path)
+{
+   int errno=0;
+   char data[5000];
+
+   struct task_struct *task_info;
+   task_info=find_task_by_vpid(inp_pid);
+   if (task_info==NULL)
+   {
+       printk("PID not found \n");
+       errno=-EINVAL;
+       return errno;
+       /* code */
+   }
+           snprintf(data,5000,"Process: %s\n \
+       PID_Number: %ld\n \
+       Process State: %ld\n \
+       Priority: %ld\n \
+       RT_Priority: %ld\n \
+       Static Priority: %ld\n \
+       Normal Priority: %ld\n \
+       TGID_Number: %ld\n \
+       CPU: %d\n \
+       Blocked:%d\n\
+       Real Blocked:%d\n\
+       Flags:%d\n\
+       Exit State:%d\n\
+       Exit Code:%d\n\
+       Exit Signal:%d\n\
+       Parent Death Signal:%d\n"
+       ,
+       task_info->comm,
+       (int)task_info->pid,
+       (long)task_info->state,
+       (long)task_info->prio,
+       (long)task_info->rt_priority,
+       (long)task_info->static_prio,
+       (long)task_info->normal_prio,
+       (long)task_info->tgid,
+       (int)task_info->on_cpu,
+       task_info->blocked,
+       task_info->real_blocked,
+       (int)task_info->flags,
+       (int)task_info->exit_state,
+       (int)task_info->exit_code,
+       (int)task_info->exit_signal,
```



```
+       (int)task_info->on_cpu,
+       task_info->blocked,
+       task_info->real_blocked,
+       (int)task_info->flags,
+       (int)task_info->exit_state,
+       (int)task_info->exit_code,
+       (int)task_info->exit_signal,
+       (int)task_info->pdeath_signal);
+
+   printk(KERN_ALERT"%s",data);
+   struct file *fptr;
+   char buf[256];
+
+   long copied = strncpy_from_user(buf,file_path, sizeof(buf));
+   int fd;
+   mm_segment_t old_fs=get_fs();
+   set_fs(KERNEL_DS);
+   if(copied<0 || copied == sizeof(buf)){
+       errno= -EFAULT;
+       return errno;
+   }
+   fptr=filp_open(buf, O_WRONLY|O_CREAT, 0644);
+       if(IS_ERR(fptr)){
+       printk(KERN_ALERT" Cannot Open File");
+       set_fs(old_fs);
+       return -EACCES;
+   }
+
+       fd=kernel_write(fptr,data,strlen(data)+1,0);
+       fput(fptr);
+       filp_close(fptr,NULL);
+       set_fs(old_fs);
+       if(fd<0){
+           printk(" Cannot Write to file %d",fd);
+           errno=-EBADF;
+       }
+
+return errno;
```

Error Handling

Incorrect pid	EINVAL	22	/* Invalid argument */
Unable to copy	EFAULT	14	/* Bad address */
Unable to open file	EACCES	13	/* Permission denied */
Unable to write to file	EBADF	9	/* Bad file number */

These are the four cases where errors are handled and you can see it in the syscall definition in the patch file the message inside /* */ is the message that will be printed on the terminal as explained above in the description of code using strerror(errno).