

Project 0.5: Communicate With Kernel!

SWE3015

Jeaho Hwang



Announcement

- See you in this time
 - Cannot find a free slot :(
- 9 students in the attendee list
 - Team organization will be announced in next week!



Project Goal

- Communication channels between apps and kernel
 - System call
 - Proc/sys filesystem
- Make your own system call & proc filesystem



Adding a New Syscall

- Add syscall number 600, *sys_os_project(int input)*
 - Print some strings
 - Return ($input == current_pid$)
- No instruction
 - Or Google
 - Hint: following files should be modified.
 - `arch/x86/syscalls/syscall_64.tbl`
 - `include/linux/syscalls.h`



Adding a New Syscall

- The test application

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
```

```
int main()
{
    int pid,res;
    pid = getpid();
    res = syscall(600,pid);
    if(res){
        printf("You passed the test!\n");
    }
}
```

```
hahaman5@ubuntu:~/test_syscall$ gcc -o test test.c
hahaman5@ubuntu:~/test_syscall$ ./test
You passed the test!
hahaman5@ubuntu:~/test_syscall$ █
```



- **A special, software-created file system to export kernel information to the world**
 - Each file under /proc is tied to a kernel function.
 - The kernel function generates file's contents on the fly when the file is read.
 - Ex) /proc/modules always return a list of the currently loaded modules.
- **Users can add proc entries via loadable modules**
 - For extracting useful information of Kernel
 - For kernel debugging information



Adding a New Proc File

- Make a kernel module which adds a proc file
 - By writing to the file, the target pid can be kept
 - E.g. \$ echo 1234 > /proc/os_project
 - By reading the file, the information of pid can be shown
 - E.g.

```
hahaman5@ubuntu:~/test_syscall$ cat /proc/os_project
exec domain: Linux
load weight: 1024 priority: 120 static priority: 120 normal priority: 120
exec_start: 1936096858586 vruntime: 11443507521 sum_exec_runtime: 0
```

- You can show info as you want (at least 4 variables)
- HINT: you can show the date by following

```
len = sprintf(page, "exec domain: %s\n load weight: %d priority: %d static priority: %d\n normal priority: %d\n exec_start: %lu vruntime: %lu sum_exec_runtime: %lu\n",
                ti->exec_domain->name,
                t->se.load.weight,
                t->prio, t->static_prio, t->normal_prio,
                t->se.exec_start,
                t->se.vruntime,
                t->se.sum_exec_runtime
                );
```



Adding a New Proc File

- Key functions and variables
 - init/cleanup_modules()
 - create/cleanup_proc_entry()
 - read/write_info()
 - struct pid *find_get_pid(pid_t nr)



Assignment

- Assignment submission
 - Due date: Mar 29th (Sun) 24:00
 - Mail to hahaman5@gmail.com
 - The title should be “[SWE3015] project0.5 name”
 - The mail must contains following
 - Screenshots of the system call result
 - Screenshots of the proc kernel module
 - A brief report
 - Source code you written
 - Explain how you did it briefly.



Discussion

- We will have some discussion time during class
 - If you have any question, don't hesitate!
- We are preparing web board!
 - Or facebook page could be.