

NYCU Operation System Homework 1

IOC PhD 徐浩哲 411551005

Part I: Kernel compilation.

```
vboxuser@OCS: ~/Desktop
vboxuser@OCS:~/Desktop$ uname -a
Linux OCS 5.19.12-os-411551005 #3 SMP PREEMPT_DYNAMIC Sat Oct 7 23:11:52 CST 20
23 x86_64 x86_64 x86_64 GNU/Linux
vboxuser@OCS:~/Desktop$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.3 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
vboxuser@OCS:~/Desktop$
```

Part II: System call.

- For both System call

1. Creat “myFunction” folder.
2. Creat three files named “hello.c”, “revstr.c”, and “Makefile”
3. Add following line to “myFunction/Makefile”:

```
1 obj-y := hello.o revstr.o
```

4. Add “/ myFunction” in “linux-5.19.12/Makefile” in line 1103:

```
1103 core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ myFunction/
```

5. Add the new system call to the system call table:

```
cd arch/x86/entry/syscalls/
add following line in syscall_64.tbl
```

```
548 common hello sys_hello
549 common revstr sys_revstr
```

6. Add new system call to the system call header file:

```
cd include/linux/
add following line in syscalls.h
```

```
1388 asmlinkage long sys_hello(void);
1389 asmlinkage long sys_revstr(void);
```

7. Compile the kernel:

```
>sudo make -j14
>sudo make modules_install install
>sudo
```

- sys_hello

```
#include <linux/kernel.h>
#include <linux/syscalls.h>
SYSCALL_DEFINE0(hello)
{
    printk("Hello world.\n");
    printk("441551005\n");
    return 0;
}
```

```
[ 6177.856335] Hello world.
[ 6177.856337] 441551005
```

● sys_revstr

```
#include <linux/kernel.h>
#include <linux/syscalls.h>
#include <linux/uaccess.h> // for copy_from_user() and copy_to_user()

SYSCALL_DEFINE2(revstr, int, length, const char __user *, usr_str)
{
    char *kern_str;
    int i, j;
    kern_str = kmalloc(length + 1, GFP_KERNEL);
    if (!kern_str) {
        return -ENOMEM;
    }
    // Copy string from user space to kernel space
    if (copy_from_user(kern_str, usr_str, length)) {
        kfree(kern_str);
        return -EFAULT;
    }
    // Null-terminate the string
    kern_str[length] = '\0';
    printk(KERN_INFO "The origin string: %s\n", kern_str);
    // Reverse the string
    for (i = 0, j = length - 1; i < j; i++, j--) {
        char temp = kern_str[i];
        kern_str[i] = kern_str[j];
        kern_str[j] = temp;
    }
}
```

```
printk(KERN_INFO "The reversed string: %s\n", kern_str);  
// Copy reversed string back to user space  
if (copy_to_user((void __user *)usr_str, kern_str, length)) {  
    kfree(kern_str);  
    return 0;  
}  
kfree(kern_str);  
return 0;  
}
```

```
[ 73.558261] The origin string: hello  
[ 73.558264] The reversed string: olleh  
[ 73.558269] The origin string: 5Y573M C411  
[ 73.558270] The reversed string: 114C M375Y5
```