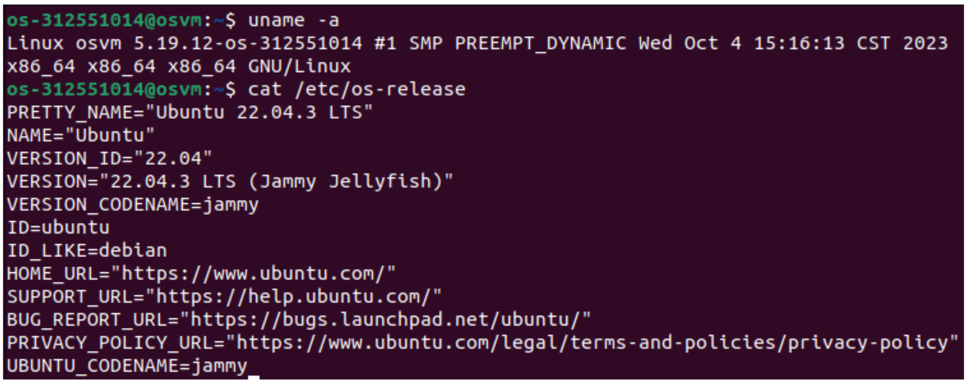
Operating System Assignment 1 Report

Student: **Chia-Hsuan Lin** (312551014)
Date: Oct. 6, 2023

Part I. Compiling Linux Kernel

Paste the screenshot of the results of executing ***uname -a*** and ***cat /etc/os-release*** commands



```
os-312551014@osvm:~$ uname -a
Linux osvm 5.19.12-os-312551014 #1 SMP PREEMPT_DYNAMIC Wed Oct 4 15:16:13 CST 2023
x86_64 x86_64 x86_64 GNU/Linux
os-312551014@osvm:~$ 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
```

Part II. Adding Custom System Calls

Describe how you implemented the two system calls in detail:

1. Create a folder ***my_syscalls*** inside the kernel source directory
2. Define the system calls
   a. Add ***my_syscalls/hello.c***

```c
#include <linux/kernel.h>
#include <linux/syscalls.h>
SYSCALL_DEFINE0(hello) {
    printk("Hello world\n");
    printk("312551014\n");
    return 0;
}
```

   b. Add ***my_syscalls/revstr.c***

```c
#include <linux/kernel.h>
#include <linux/syscalls.h>
#include <linux/uaccess.h>
#define MAX_LEN 256
SYSCALL_DEFINE2(revstr, int, len, const char __user *, str) {
    char rev_str[MAX_LEN] = {0};
    copy_from_user(rev_str, str, sizeof(rev_str));
    printk("The original string: %s\n", rev_str);
    for (int i = 0; i < len / 2; i++) {
        char temp = rev_str[i];
        rev_str[i] = rev_str[len - i - 1];
        rev_str[len - i - 1] = temp;
    }
    printk("The reversed string: %s\n", rev_str);
    return 0;
}
```

c. Add ***my_syscalls/Makefile***

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

3. Modify the kernel source code
    a. Modify ***Makefile***
        i. Find the keyword ***core-y***
        ii. Add the folder ***my_syscalls*** to the line

```
kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ my_syscalls/
```

b. Modify the header file ***include/linux/syscalls.h***

Add the declaration of the system calls above ***#endif*** at the bottom of the file

```
asmlinkage long sys_hello(void);
asmlinkage long sys_revstr(int len, const char __user *str);
```

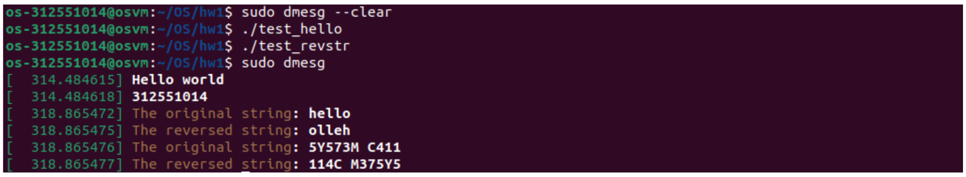
c. Modify the system call table in ***arch/x86/entry/syscalls/syscall_64.tbl***

Add the following lines above the x32 system call section

```
451 common hello sys_hello
452 common revstr sys_revstr
```

4. Compile the kernel source code
5. Test with the provided user code

Paste the screenshot of the messages the system call printed:



```
os-312551014@osvm:~/OS/hw1$ sudo dmesg --clear
os-312551014@osvm:~/OS/hw1$ ./test_hello
os-312551014@osvm:~/OS/hw1$ ./test_revstr
os-312551014@osvm:~/OS/hw1$ sudo dmesg
[  314.484615] Hello world
[  314.484618] 312551014
[  318.865472] The original string: hello
[  318.865475] The reversed string: olleh
[  318.865476] The original string: 5Y573M C411
[  318.865477] The reversed string: 114C M375Y5
```