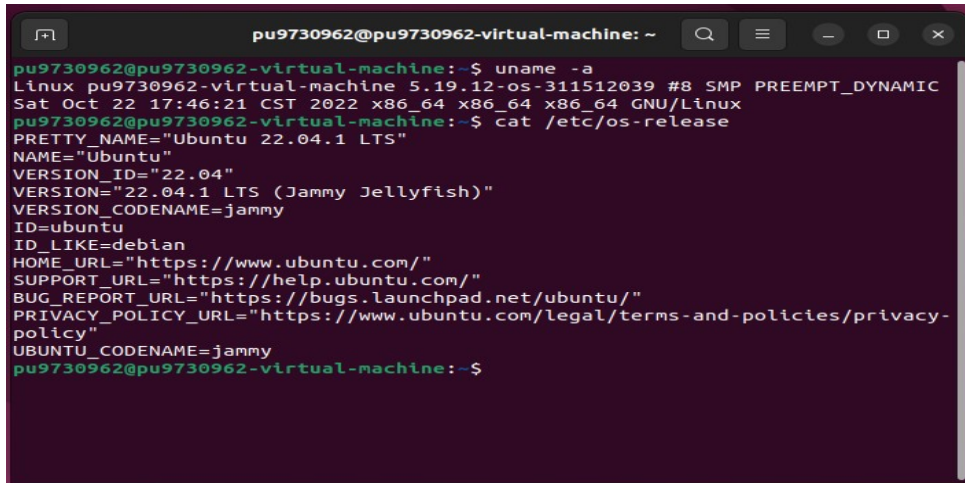


Assignment 1: Compiling Linux Kernel and Adding Custom System Call

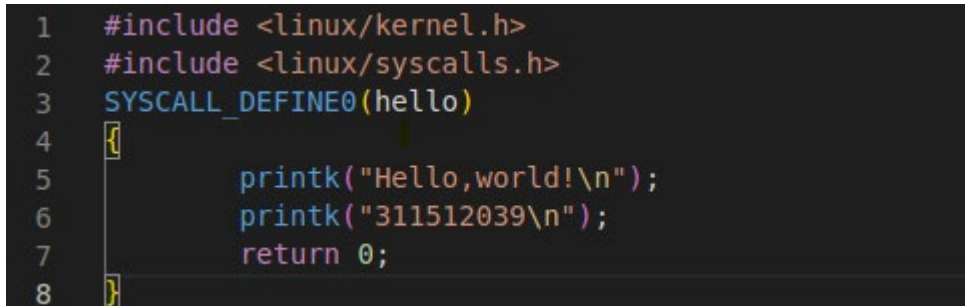
- Part 1: Kernel compilation



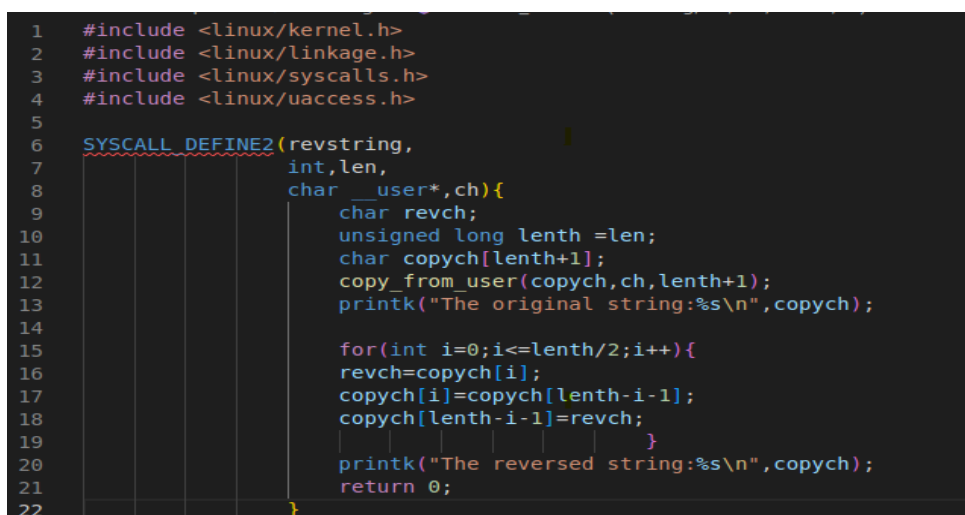
```
pu9730962@pu9730962-virtual-machine: ~  
pu9730962@pu9730962-virtual-machine:~$ uname -a  
Linux pu9730962-virtual-machine 5.19.12-os-311512039 #8 SMP PREEMPT_DYNAMIC  
Sat Oct 22 17:46:21 CST 2022 x86_64 x86_64 x86_64 GNU/Linux  
pu9730962@pu9730962-virtual-machine:~$ cat /etc/os-release  
PRETTY_NAME="Ubuntu 22.04.1 LTS"  
NAME="Ubuntu"  
VERSION_ID="22.04"  
VERSION="22.04.1 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  
pu9730962@pu9730962-virtual-machine:~$
```

- Part 2 : Adding system call

- 在linux-5.19.12下建立新資料夾(workspace)並寫一個自訂義的system call，分別是sys_hello.c及sys_revstr.c，如下圖



```
1 #include <linux/kernel.h>  
2 #include <linux/syscalls.h>  
3 SYSCALL_DEFINE0(hello)  
4 {  
5     printk("Hello,world!\n");  
6     printk("311512039\n");  
7     return 0;  
8 }
```



```
1 #include <linux/kernel.h>  
2 #include <linux/linkage.h>  
3 #include <linux/syscalls.h>  
4 #include <linux/uaccess.h>  
5  
6 SYSCALL_DEFINE2(revstring,  
7 int, len,  
8 char __user*, ch){  
9     char revch;  
10    unsigned long lenth = len;  
11    char copych[lenth+1];  
12    copy_from_user(copych, ch, lenth+1);  
13    printk("The original string:%s\n", copych);  
14  
15    for(int i=0; i<=lenth/2; i++){  
16        revch=copych[i];  
17        copych[i]=copych[lenth-i-1];  
18        copych[lenth-i-1]=revch;  
19    }  
20    printk("The reversed string:%s\n", copych);  
21    return 0;  
22 }
```

- 透過 `__user` 來告訴kernel此變數是由user傳入
- 字串無法直接從user傳入kernel，必須透過指標傳入後並複製 (`copy_from_user`)

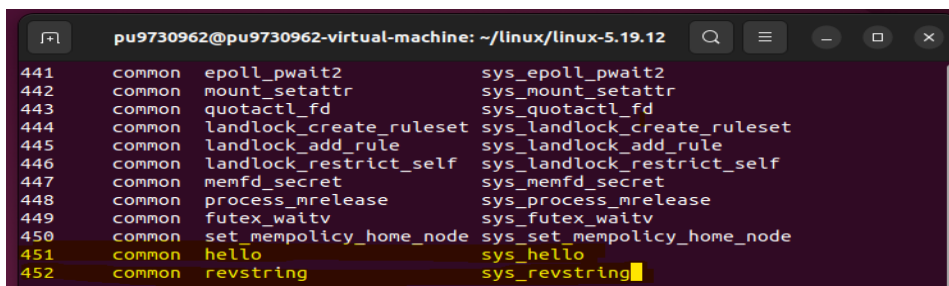
2. 在同資料夾(workspace)下建立一個Makefile檔，內容如下

```
1  obj-y += hello_world.o
2  obj-y += revstring.o
```

3. 修改kernel的Makefile，使compile時可以找到自訂義的system call 目錄

```
ifeq ($(KBUILD_EXTMOD),)
core-y      += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/
core-$(CONFIG_BLOCK) += block/ workspace/
```

4. 修改system call table使OS尋找對應的trap service routine
路徑: vim arch/86/entry/syscalls/syscall_64.tbl



```
441 common epoll_pwait2 sys_epoll_pwait2
442 common mount_setattr sys_mount_setattr
443 common quotactl_fd sys_quotactl_fd
444 common landlock_create_ruleset sys_landlock_create_ruleset
445 common landlock_add_rule sys_landlock_add_rule
446 common landlock_restrict_self sys_landlock_restrict_self
447 common memfd_secret sys_memfd_secret
448 common process_mrelease sys_process_mrelease
449 common futex_waitv sys_futex_waitv
450 common set_mempolicy_home_node sys_set_mempolicy_home_node
451 common hello sys_hello
452 common revstring sys_revstring
```

5. 在system call header file增加自訂義的system call prototype
路徑:vim include/linux/syscalls.h

```
1031 unsigned mask, struct statx __user *buffer);
1032 asmlinkage long sys_hello(void);
1033 asmlinkage long sys_revstring(void);
```

6. Compile the kernel, 依序執行下面指令:

- ◆ sudo make -j(核心數)
- ◆ sudo make modules_install
- ◆ sudo make install
- ◆ sudo reboot

7. 結果

```
pu9730962@pu9730962-virtual-machine: ~/linux/hello_...  
pu9730962@pu9730962-virtual-machine:~/linux/hello_workspace$ sudo dmesg  
[ 488.260460] Hello,world!  
[ 488.260463] 311512039  
pu9730962@pu9730962-virtual-machine:~/linux/hello_workspace$
```

```
pu9730962@pu9730962-virtual-machine: ~/linux/revstring_w...  
pu9730962@pu9730962-virtual-machine:~/linux/revstring_workspace$ sudo dmesg  
[ 234.407952] The original string:hello  
[ 234.407955] The reversed string:olleh  
[ 234.407956] The original string:5Y573M C411  
[ 234.407957] The reversed string:114C M375Y5  
pu9730962@pu9730962-virtual-machine:~/linux/revstring_workspace$ S
```