

OS Assignment 1 Report

110550108 施柏江

I. Kernel Compilation

1. 安裝需要的 library

```
sudo apt install make gcc flex bison libelf-dev libssl-dev
```

2. 清理之前的配置，確保從乾淨的狀態開始

```
make mrproper
```

3. 複製當前 kernel 的配置

```
cp /boot/config-`uname -r`* .config
```

4. 調整配置以更改 kernel 版本的 suffix

```
CONFIG_LOCALVERSION="-os-110550108"
```

```
CONFIG_LOCALVERSION_AUTO=y
```

5. 使用所有可用的 cpu core 來加快編譯速度

```
make -j $(nproc)
```

6. 編譯完後將 modules 安裝到系統

```
sudo make modules_install
```

7. 將編譯好的 kernel 安裝到系統

```
sudo make install
```

8. 建立一個與新 kernel 對應的 initramfs

```
sudo update-initramfs -c -k 6.1.0-os-110550108
```

9. 更新 GRUB，使系統可以選擇新 kernel

```
sudo update-grub
```

10. 重新啟動以使用新 kernel

```
sudo reboot
```

Result:

```
brian@brian-VirtualBox:~/桌面$ uname -a
Linux brian-VirtualBox 6.1.0-os-110550108 #1 SMP PREEMPT_DYNAMIC Mon Oct 7 17:15:33
CST 2024 x86_64 x86_64 x86_64 GNU/Linux
brian@brian-VirtualBox:~/桌面$ cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.1 LTS (Noble Numbat)"
VERSION_CODENAME=noble
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=noble
LOGO=ubuntu-logo
```

II. System call implementation

1. 創建 revstr 資料夾並在其中實作 revstr function，它接受來自 user space 的字串，並將字串反轉後再傳回 user space。

```
mkdir revstr
```

```
cd revstr
```

```
vim revstr.c
```

```
#include <linux/kernel.h>
#include <linux/syscalls.h>
#include <linux/uaccess.h>

SYSCALL_DEFINE2(revstr, char __user *, str, int, len) {
    char r_str[256];
    if(copy_from_user(r_str, str, len)){
        printk("Fail to copy from user!!!\n");
        return -EFAULT;
    }
    printk("The origin string: %s\n", r_str);
    for (int i = 0; i < len/2; ++i) {
        char tmp = r_str[i];
        r_str[i] = r_str[len-i-1];
        r_str[len-i-1] = tmp;
    }
    printk("The reversed string: %s\n", r_str);
    if(copy_to_user(str, r_str, len)){
        printk("Fail to copy to user!!!\n");
        return -EFAULT;
    }
    return 0;
}
```

2. 將 revstr.c 加入到 Makefile，讓 kernel 能夠編譯

```
vim Makefile
```

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

3. 在 kernel 的 Makefile 中增加 revstr

```
cd ..
```

```
vim Makefile
```

```
core-y := kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ revstr/
```

4. 為 revstr 分配 system call 代號

```
vim arch/x86/entry/syscalls/syscall_64.tbl
```

```
451      common  revstr      sys_revstr
```

5. 宣告 sys_revstr

```
vim include/linux/syscalls.h
```

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

6. 重新編譯 kernel 並安裝 modules

```
make -j $(nproc)
```

```
sudo make modules_install
```

```
sudo make install
```

```
sudo reboot
```

7. 使用助教提供的 test code 來測試 sys_revstr

```
cd linux/revstr
```

```
vim test_revstr.c
```

```
gcc -o test_revstr test_revstr.c
```

Result:

```
brian@brian-VirtualBox:~/桌面/linux/revstr$ ./test_revstr
Ori: hello
Rev: olleh
Ori: Operating System
Rev: metsyS gnitarep0
```

```
[ 99.841857] The origin string: hello
[ 99.841907] The reversed string: olleh
[ 99.841915] The origin string: Operating System
[ 99.841916] The reversed string: metsyS gnitarep0
```