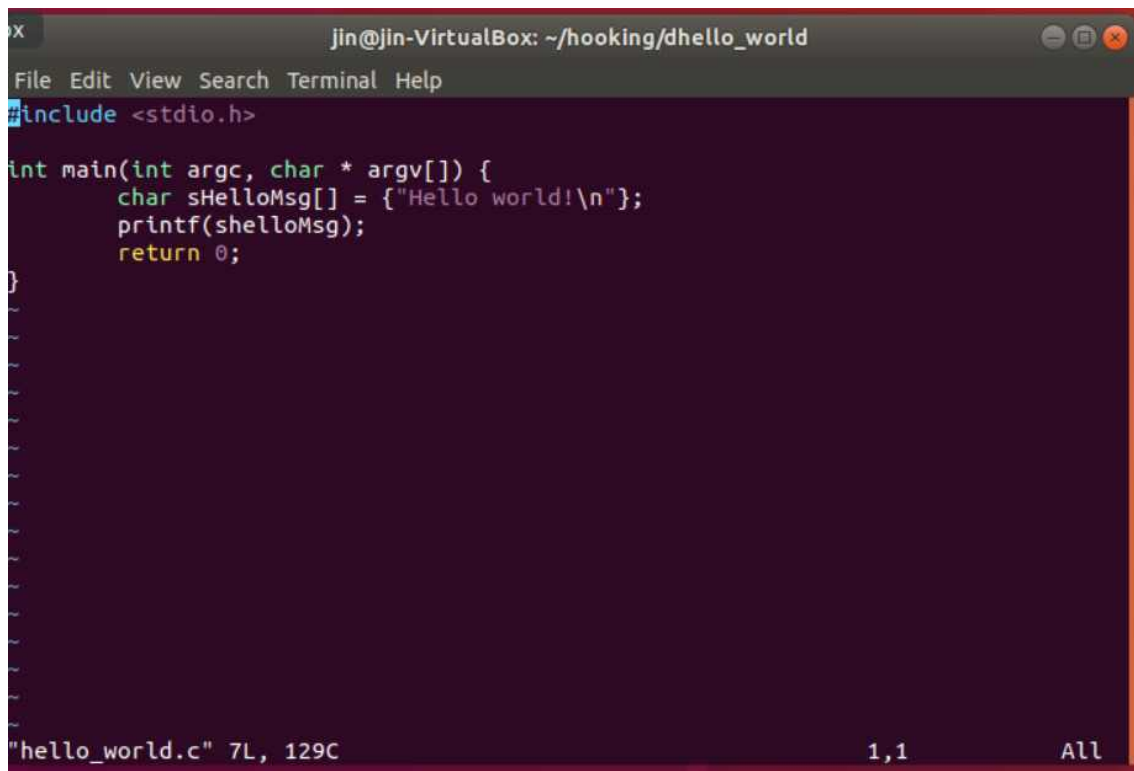


실습예제1)

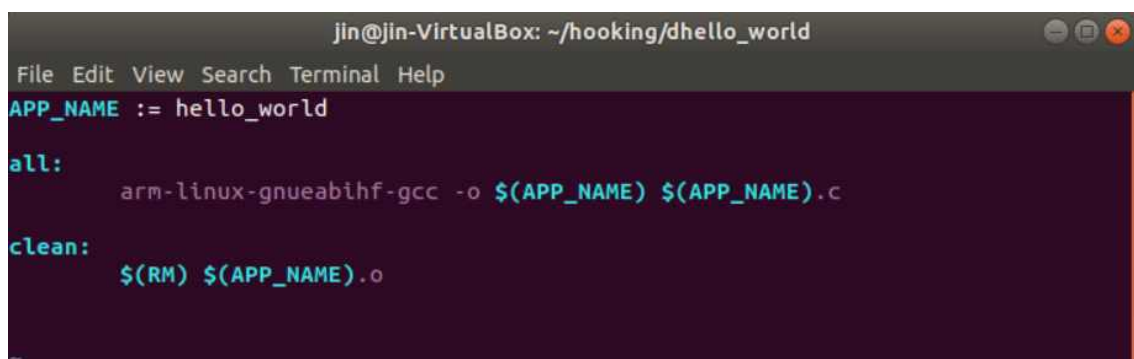


```
jin@jin-VirtualBox: ~/hooking/dhello_world
File Edit View Search Terminal Help
#include <stdio.h>

int main(int argc, char * argv[]) {
    char sHelloMsg[] = {"Hello world!\n"};
    printf(sHelloMsg);
    return 0;
}

"hello_world.c" 7L, 129C 1,1 All
```

hooking 디렉토리를 생성하고 해당 디렉토리 안에 hooking 실행 전 정상적으로 출력이 됨을 보일 dhello_world.c 파일을 생성하였습니다.



```
jin@jin-VirtualBox: ~/hooking/dhello_world
File Edit View Search Terminal Help
APP_NAME := hello_world

all:
    arm-linux-gnueabi-gcc -o $(APP_NAME) $(APP_NAME).c

clean:
    $(RM) $(APP_NAME).o
```

dhello_world의 Makefile을 생성하였습니다.

```
jinn@jin-VirtualBox: ~/hooking/dhooker
File Edit View Search Terminal Help
#include <linux/module.h>
#include <linux/syscalls.h>
#include <linux/string.h>

#define SYSCALL_TABLE_BASE_ADDR (0x8000fbe8)
#define MANAGER_PERMISSION      (0xff)

unsigned int ** g_puSysTableAddr = (unsigned int**)SYSCALL_TABLE_BASE_ADDR;
unsigned int g_uPrevAP = 0x00;
unsigned int g_uNewAP = MANAGER_PERMISSION;
unsigned int (* sys_write_orig)(int fd, char *buf, size_t count);

unsigned int sys_write_hooked(int nFD, char *pBuf, size_t nCnt) {

    if(nFD==1) {
        memset(pBuf, 0, nCnt);
        strcpy(pBuf, "Hacked!!!\n");
        return sys_write_orig(nFD, pBuf, nCnt);
    }

    else
        return sys_write_orig(nFD, pBuf, nCnt);
}

-- INSERT -- 24,2 Bot

jinn@jin-VirtualBox: ~/hooking/dhooker
File Edit View Search Terminal Help

int __init Hook_Init(void) {
    sys_write_orig = (void *)g_puSysTableAddr[__NR_write];

    __asm__ __volatile__("mrc p15, 0, %0, c3, c0" : "=r"(g_uPrevAP));
    __asm__ __volatile__("mcr p15, 0, %0, c3, c0" : "=r"(g_uNewAP));

    g_puSysTableAddr[__NR_write] = (unsigned int *) sys_write_hooked;

    __asm__ __volatile__("mcr p15, 0, %0, c3, c0" : "=r"(g_uPrevAP));

    return 0;
}

void __exit Hook_Exit(void) {

    __asm__ __volatile__("mrc p15, 0, %0, c3, c0" : "=r"(g_uPrevAP));
    __asm__ __volatile__("mcr p15, 0, %0, c3, c0" : "=r"(g_uNewAP));

    g_puSysTableAddr[__NR_write] = (unsigned int *) sys_write_orig;

    __asm__ __volatile__("mcr p15, 0, %0, c3, c0" : "=r"(g_uPrevAP));
}

module_init(Hook_Init);
module_exit(Hook_Exit);
-- INSERT -- 50,24 88%
```

hooking을 실행할 때의 함수 내용은 위 그림과 같습니다. sys_write_hooked 가 실행될 때 system에 write하는 (printf, ls 등등) 함수가 실행되면 pBuf의 char을 바꿉니다.

따라서 어떤 내용을 출력하는 함수는 모두 "Hacked"를 출력하며 해킹 됩니다.

```
pi@raspberrypi:~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo cat /proc/kallsyms | grep sys_call_table  
8000fbe8 T sys_call_table  
pi@raspberrypi:~ $
```

<라즈베리파이의 syscall_table : base address 확인>

```
jin@jin-VirtualBox: ~/hooking  
File Edit View Search Terminal Help  
export APP_NAME = hello_world  
export MOD_NAME = hooker  
  
PWD := $(shell pwd)  
APP_PATH=$(PWD)/d$(APP_NAME)  
MOD_PATH=$(PWD)/d$(MOD_NAME)  
  
all: $(MOD_NAME) $(APP_NAME)  
  
$(MOD_NAME):  
    $(MAKE) -C $(MOD_PATH)  
    mv $(MOD_PATH)/$.ko $(PWD)  
  
$(APP_NAME):  
    $(MAKE) -C $(APP_PATH)  
    mv $(APP_PATH)/$.ko $(PWD)  
  
clean:  
    $(RM) $(PWD)/$(MOD_NAME).ko  
    $(RM) $(PWD)/$(APP_NAME)  
    arm-linux-gnueabi-gcc -C $(MOD_PATH) clean  
    arm-linux-gnueabi-gcc -C $(APP_PATH) clean  
~  
~
```

```
jin@jin-VirtualBox: ~/hooking/dhooker  
File Edit View Search Terminal Help  
obj-m := hooker.o  
  
KDIR=/home/working/linux/  
PWD=$(shell pwd)  
TOOLCHAIN=arm-linux-gnueabi-gcc  
TARGET=arm  
  
all:  
    $(MAKE) -C $(KDIR) M=$(PWD) ARCH=$(TARGET) CROSS_COMPILE=$(TOOLCHAIN) modules  
  
clean:  
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) clean  
~  
~  
~
```

Hooker 파일에 대한 Makefile을 만들고 그 뒤에 hooker 디렉토리를 나와서 전체 프로그램에 대한 Makefile을 새로 만들어 주었습니다.


```

jin@jin-VirtualBox:~/hooking$ make
make -C /home/jin/hooking/dhooker
make[1]: Entering directory '/home/jin/hooking/dhooker'
make -C /home/jin/working/linux/ M=/home/jin/hooking/dhooker ARCH=arm CROSS_COMPILE=arm-li
nux-gnueabihf- modules
make[2]: Entering directory '/home/jin/working/linux'
  CC [M] /home/jin/hooking/dhooker/hooker.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC /home/jin/hooking/dhooker/hooker.mod.o
  LD [M] /home/jin/hooking/dhooker/hooker.ko
make[2]: Leaving directory '/home/jin/working/linux'
make[1]: Leaving directory '/home/jin/hooking/dhooker'
mv /home/jin/hooking/dhooker/hooker.ko /home/jin/hooking
make -C /home/jin/hooking/dhello_world
make[1]: Entering directory '/home/jin/hooking/dhello_world'
arm-linux-gnueabi-gcc -o hello_world hello_world.c
hello_world.c: In function 'main':
hello_world.c:5:9: warning: format not a string literal and no format arguments [-Wformat-
security]
    printf(sHelloMsg);
    ~~~~~
make[1]: Leaving directory '/home/jin/hooking/dhello_world'
mv /home/jin/hooking/dhello_world/hello_world /home/jin/hooking

```

정상적으로 라즈베리파이 환경에서의 cross-compile이 실행되는 것을 확인하였습니다.

```

File Edit Tabs Help
pi@raspberrypi:~$ ls
4.4.50-v7+ Downloads led1.c Music start.cgi Templates
clear.c hello.c led1.cgi Pictures stop.c Videos
clear.cgi hooking led2.c Public stop.cgi WiringPi
Desktop led11.c led22.cgi python_games syscall2_app
Documents led11.cgi led2.c start.c syscall_app
pi@raspberrypi:~$ cd hooking
pi@raspberrypi:~/hooking$ ls
dhello_world dhooker hello_world hooker.ko Makefile
pi@raspberrypi:~/hooking$ ./hello_world
Hello world!
pi@raspberrypi:~/hooking$ sudo insmod hooker.ko
pi@raspberrypi:~/hooking$ ./hello_world
Hacked!!!
pi@raspberrypi:~/hooking$ ls
Hacked!!!
pi@raspberrypi:~/hooking$ vi what
Hacked!!!
Hacked!!!
Hacked!!!
HHHHHHHHHacked!!!
Hacked!!!
HHHackedHacked!!!
Hacked!!!
pi@raspberrypi:~/hooking$ sudo rmmod hooker
pi@raspberrypi:~/hooking$ ./hello_world
Hello world!
pi@raspberrypi:~/hooking$ ls
dhello_world dhooker hello_world hooker.ko Makefile
pi@raspberrypi:~/hooking$

```

이후 라즈베리파이 안으로 파일을 옮긴 후 hooker 실행 이전, 이후를 비교했을 시 후킹이 잘 되는 것을 확인하였습니다.

실습 과제)

```
jin@jin-VirtualBox: ~/woobin_hooking
File Edit View Search Terminal Help
export APP_NAME = hello_woobin
export MOD_NAME = hooker

PWD := $(shell pwd)
APP_PATH=$(PWD)/d$(APP_NAME)
MOD_PATH=$(PWD)/d$(MOD_NAME)

all: $(MOD_NAME) $(APP_NAME)

$(MOD_NAME):
    $(MAKE) -C $(MOD_PATH)
    mv $(MOD_PATH)/$@.ko $(PWD)

$(APP_NAME):
    $(MAKE) -C $(APP_PATH)
    mv $(APP_PATH)/$@ $(PWD)

clean:
    $(RM) $(PWD)/$(MOD_NAME).ko
    $(RM) $(PWD)/$(APP_NAME)
    arm-linux-gnueabihf-gcc -C $(MOD_PATH) clean
    arm-linux-gnueabihf-gcc -C $(APP_PATH) clean
~
~
```

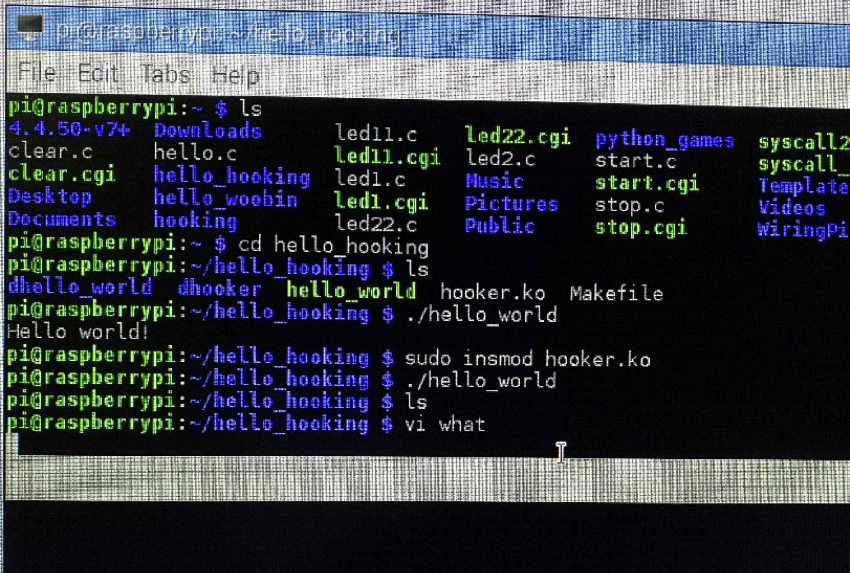
```
jin@jin-VirtualBox: ~/hello_hooking/hello_woobin
File Edit View Search Terminal Help
#include <stdio.h>

int main(int argc, char *argv[]) {
    char sWoobinMsg[] = {"Hello woobin!\n"};
    printf(sWoobinMsg);
    return 0;
}
~
~
```

Hello World 파일은 그대로 두고 이와 비교하기 위한 다른 파일을 생성하였습니다.

```
unsigned int sys_write_hooked(int nFD, char *pBuf, size_t nCnt) {  
    if(nFD==1) {  
        char *pname = current->comm;  
        char *hello = "hello_world";  
        if(pname==hello) {  
            memset(pBuf, 0 ,nCnt);  
            strcpy(pBuf, "Hacked!!!\n");  
            return sys_write_orig(nFD, pBuf, nCnt);  
        }  
    }  
    else  
        return sys_write_orig(nFD, pBuf, nCnt);  
}
```

hooker.c의 대부분 함수 내용은 같으나 sys_write_hooked를 수정하였습니다.
current->comm 명령어를 통해 현재 수행되고 있는 프로세스의 프로세스 name을 알 수 있습니다. 그리고 나서 hello_world와의 비교연산을 통해 프로세스의 name이 "hello_world" 이면 hooking이 일어나도록 수정하였습니다.



```
pi@raspberrypi:~/hello_hooking  
File Edit Tabs Help  
pi@raspberrypi:~$ ls  
4.4.50-v7+ Downloads led11.c led22.cgi python_games syscall2  
clear.c hello.c led11.cgi led2.c start.c syscall_  
clear.cgi hello_hooking led1.c Music start.cgi Template  
Desktop hello_woobin led1.cgi Pictures stop.c Videos  
Documents hooking led22.c Public stop.cgi WiringPi  
pi@raspberrypi:~$ cd hello_hooking  
pi@raspberrypi:~/hello_hooking$ ls  
dhello_world dhooker hello_world hooker.ko Makefile  
pi@raspberrypi:~/hello_hooking$ ./hello_world  
Hello world!  
pi@raspberrypi:~/hello_hooking$ sudo insmod hooker.ko  
pi@raspberrypi:~/hello_hooking$ ./hello_world  
pi@raspberrypi:~/hello_hooking$ ls  
pi@raspberrypi:~/hello_hooking$ vi what
```

이후 hello_hooking 폴더에서 hello_world 실행 결과 hooker.ko 실행 후엔 Hello world! 가 출력되지 않는 것을 확인할 수 있었습니다.