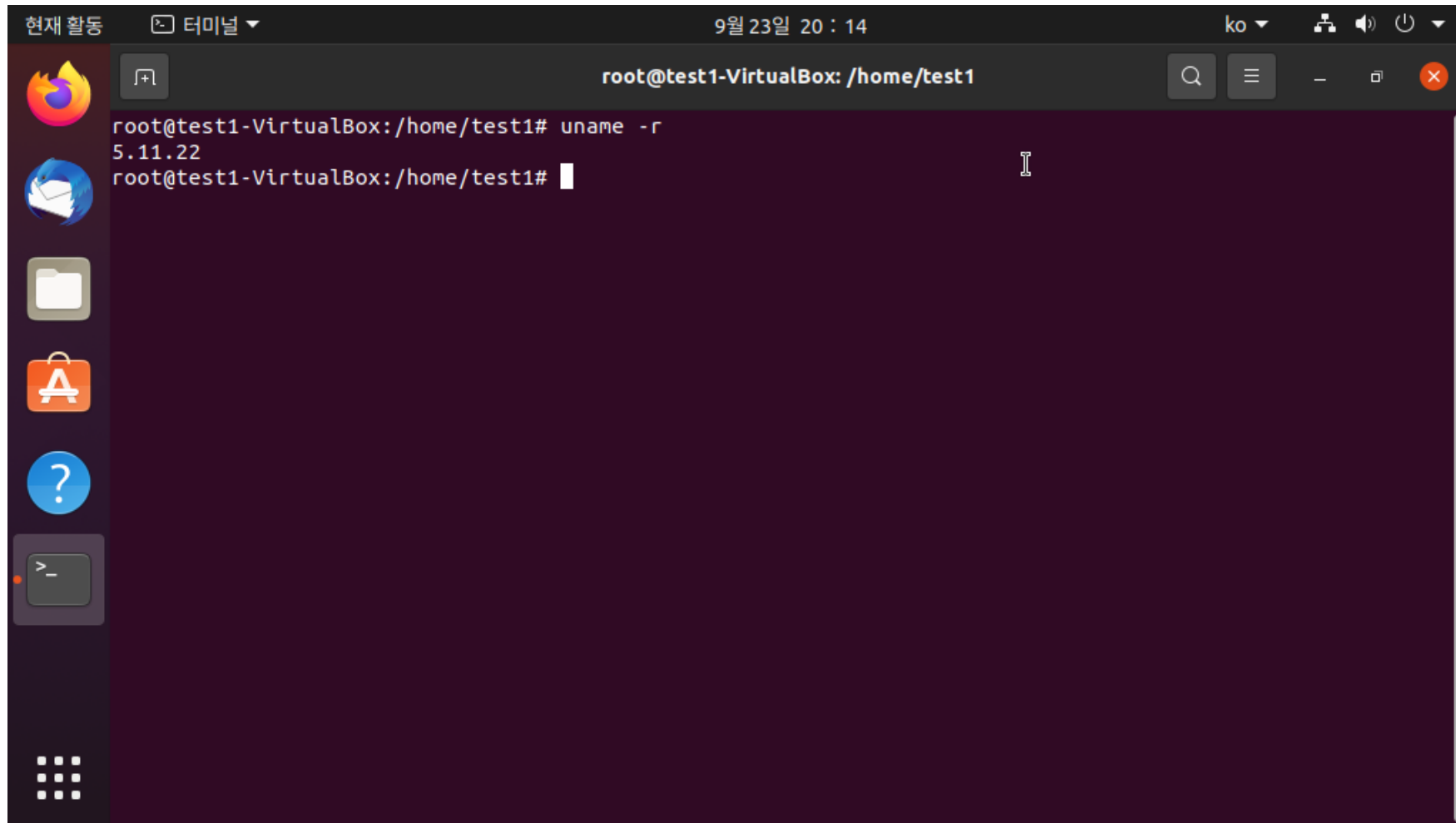


과제 #3 참고자료

(Hello World 시스템 콜 추가)

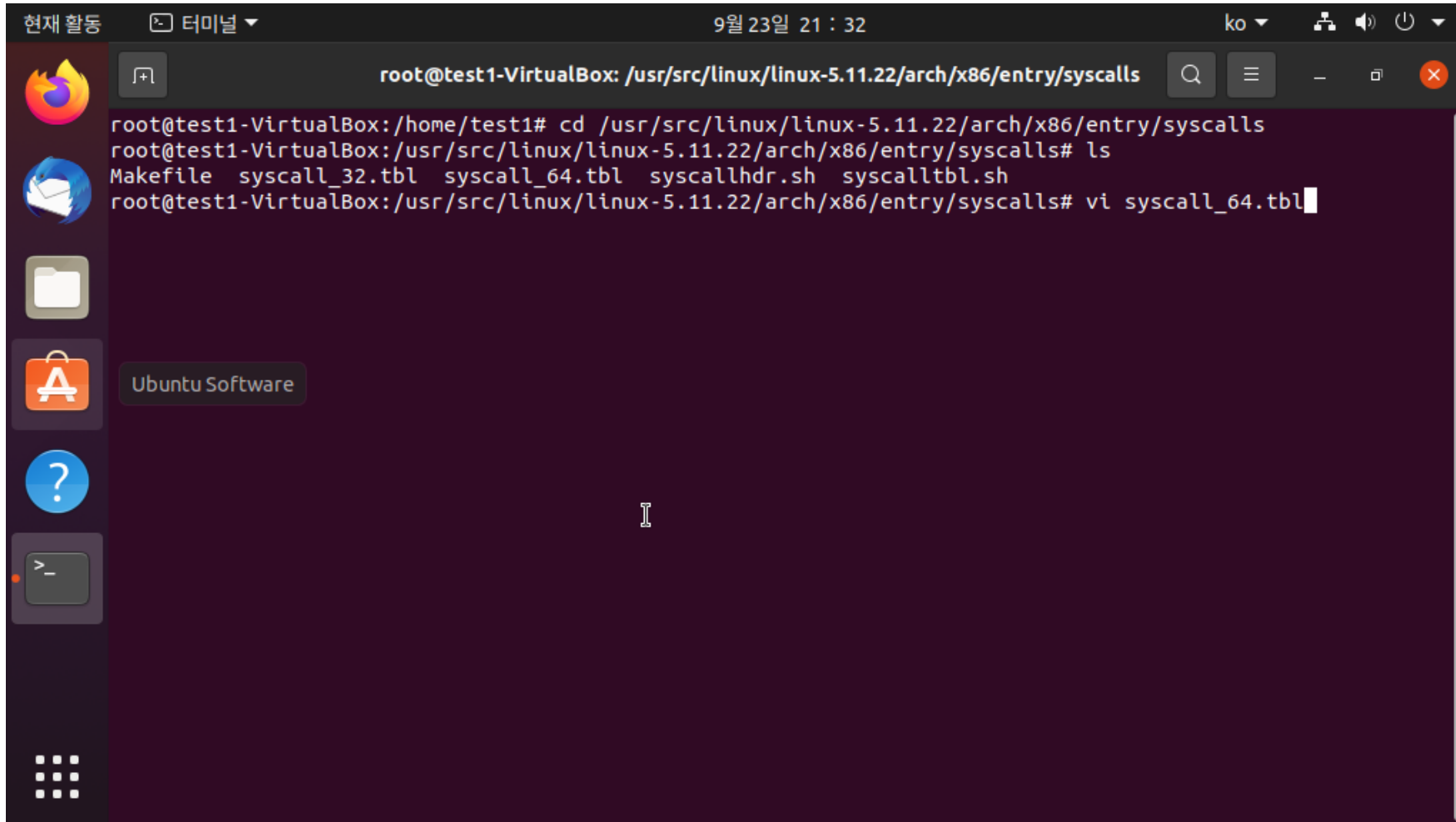


A screenshot of a Linux terminal window. The window title bar shows '현재 활동' (Current Activity), '터미널' (Terminal), the date '9월 23일 20:14', and the language 'ko'. The terminal content shows the command 'uname -r' being executed, resulting in the output '5.11.22'. The prompt 'root@test1-VirtualBox: /home/test1#' is visible. The window has a dark theme and a sidebar with application icons on the left.

```
root@test1-VirtualBox: /home/test1# uname -r
5.11.22
root@test1-VirtualBox: /home/test1#
```

*** 3,4번 과제는 커널 버전 5.11.22 설치된 환경에서 실시함 ***

1. 시스템 콜 추가 - 시스템 콜 테이블 등록



The screenshot shows a terminal window titled "root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls". The terminal output is as follows:

```
root@test1-VirtualBox:/home/test1# cd /usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls# ls
Makefile syscall_32.tbl syscall_64.tbl syscallhdr.sh syscalltbl.sh
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls# vi syscall_64.tbl
```

The terminal window has a dark background and a light-colored text. The left sidebar shows various application icons, including a web browser, a mail client, a file manager, and the Ubuntu Software center. The top bar of the window displays the current date and time as "9월 23일 21 : 32" and the language as "ko".

/usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls 디렉토리로 이동 후
운영체제 종류 (64bit 혹은 32bit)에 맞는 syscall_xx.tbl 파일 편집

* 본 참고자료의 경우 64bit로 진행함

1. 시스템 콜 추가 - 시스템 콜 테이블 등록

```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls
430    common    fsopen          sys_fsopen
431    common    fsconfig       sys_fsconfig
432    common    fsmount       sys_fsmount
433    common    fspick        sys_fspick
434    common    pidfd_open    sys_pidfd_open
435    common    clone3        sys_clone3
436    common    close_range   sys_close_range
437    common    openat2       sys_openat2
438    common    pidfd_getfd   sys_pidfd_getfd
439    common    faccessat2    sys_faccessat2
440    common    process_madvise sys_process_madvise
441    common    epoll_pwait2  sys_epoll_pwait2
442    common    print_hello   sys_print_hello
# On x32 as compared to native x86_64, these syscalls have numbers 512-547.
# Do not add new syscalls to this range. Numbers 548 and above are available
# for non-x32 use.
#
512    x32       rt_sigaction   compat_sys_rt_sigaction
513    x32       rt_sigreturn  compat_sys_x32_rt_sigreturn
514    x32       ioctl         compat_sys_ioctl
515    x32       readv         sys_readv
516    x32       writev        sys_writev
517    x32       recvmmsg      compat_sys_recvmmsg
518    x32       sendmsg       compat_sys_sendmsg
519    x32       recvmsg       compat_sys_recvmsg
520    x32       execve        compat_sys_execve
-- 끼워넣기 --
```

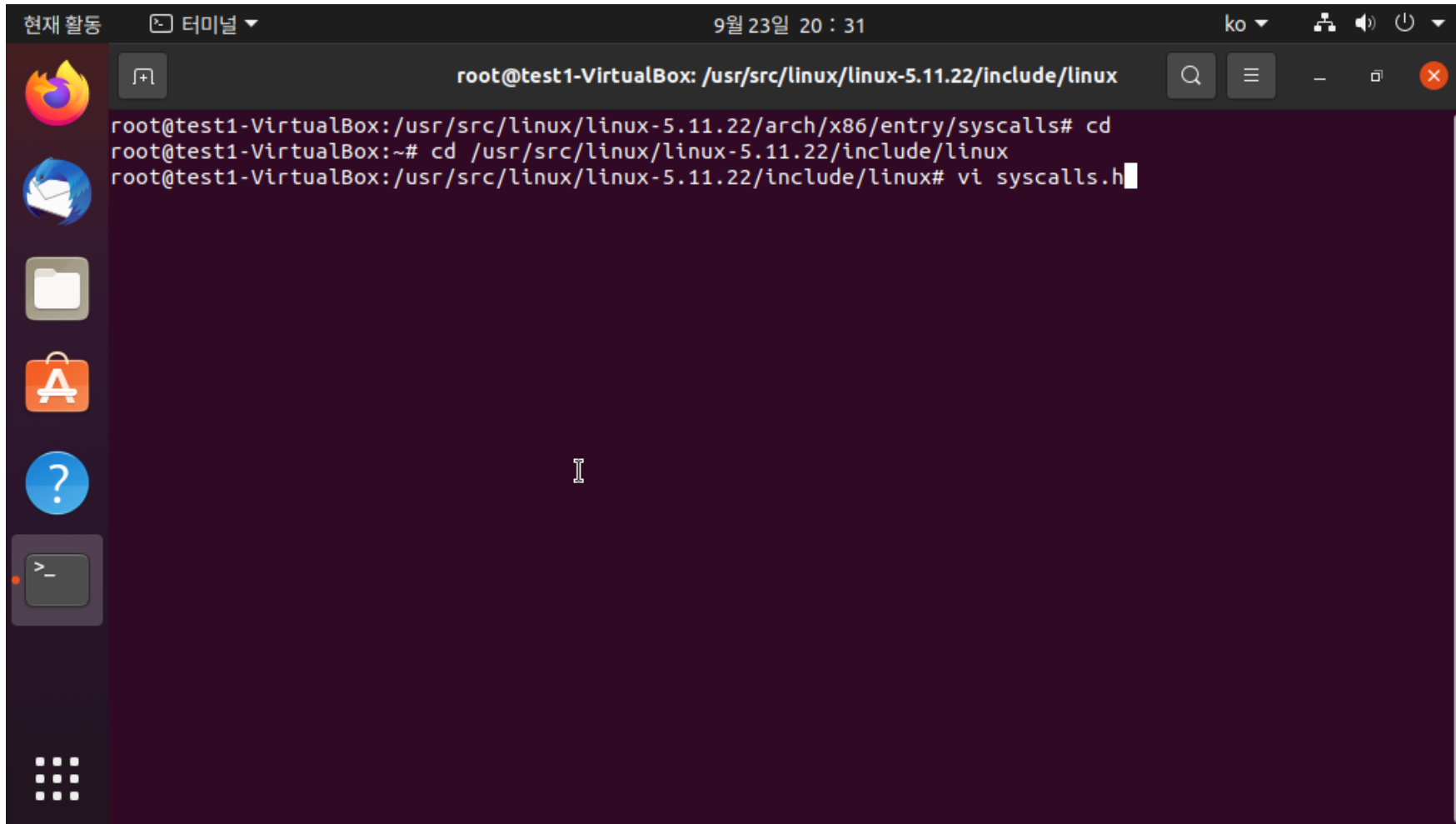
시스템 콜 테이블은 <number> <abi> <name> <entry point> 순으로 구성되어 있음

64bit 시스템 콜 테이블의 마지막에 이어지도록 작성함

도중에 사용 불가능한 구간이 존재하므로 주석을 잘 읽어보고 작성

추가한 시스템 콜의 번호는 기억해야 함

1. 시스템 콜 추가 - 시스템 콜 헤더 파일에 등록



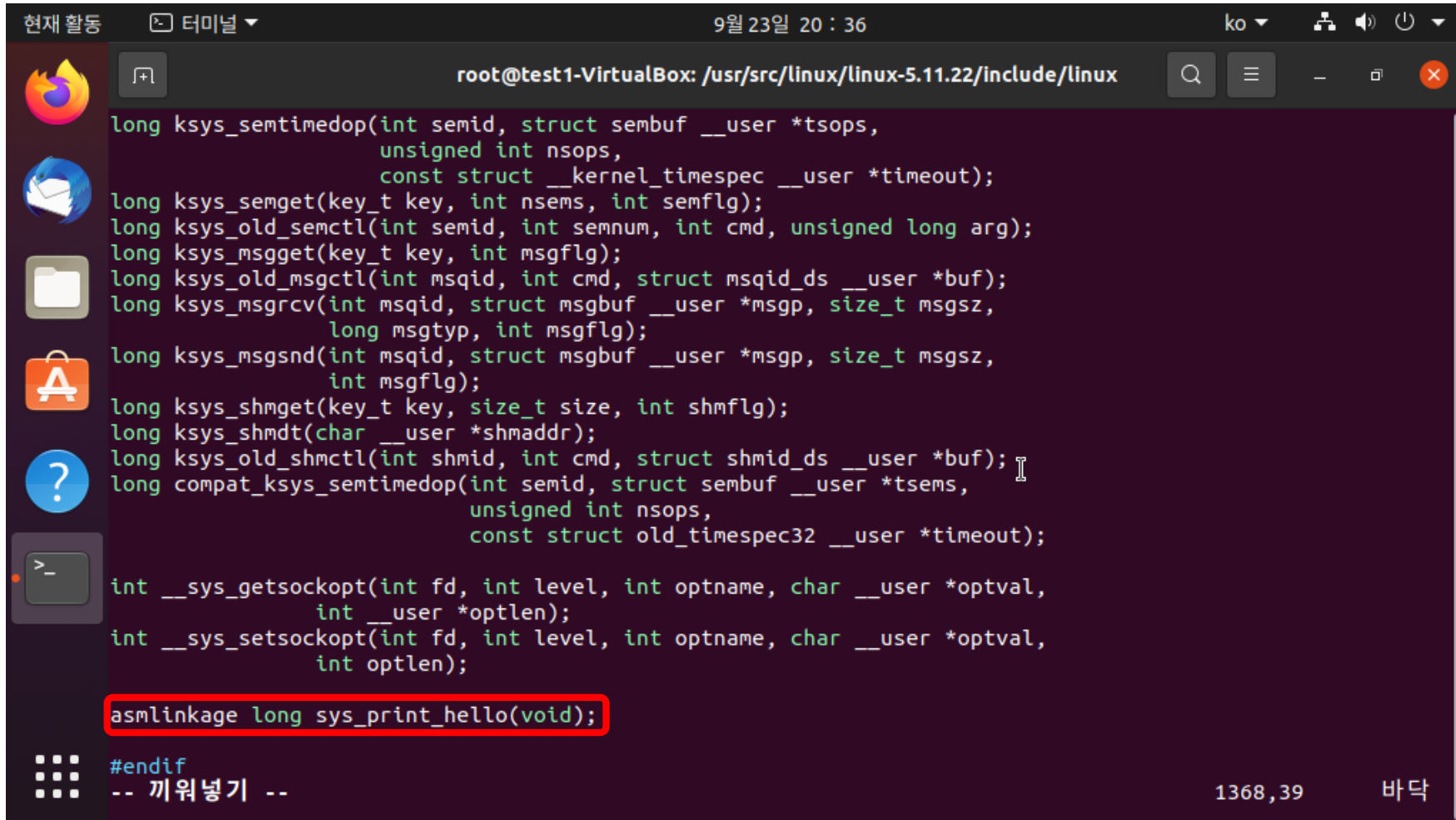
A terminal window titled "현재 활동" (Current Activity) with a subtitle "터미널" (Terminal). The window shows a root user at a test1-VirtualBox machine. The terminal output is as follows:

```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/include/linux
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/arch/x86/entry/syscalls# cd
root@test1-VirtualBox:~# cd /usr/src/linux/linux-5.11.22/include/linux
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/include/linux# vi syscalls.h
```

The terminal window has a dark background with a light-colored text. The left sidebar shows icons for Firefox, a mail client, a file manager, an application store, and a help icon. The top bar shows the date and time "9월 23일 20 : 31" and the language "ko".

/usr/src/linux/linux-5.11.22/include/linux 디렉토리로 이동 후 syscalls.h를 vi 편집기로 열기

1. 시스템 콜 추가 - 시스템 콜 헤더 파일에 등록



```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/include/linux

long ksys_semtimedop(int semid, struct sembuf __user *tsops,
                    unsigned int nsops,
                    const struct __kernel_timespec __user *timeout);
long ksys_semget(key_t key, int nsems, int semflg);
long ksys_old_semctl(int semid, int semnum, int cmd, unsigned long arg);
long ksys_msgget(key_t key, int msgflg);
long ksys_old_msgctl(int msqid, int cmd, struct msqid_ds __user *buf);
long ksys_msgrcv(int msqid, struct msgbuf __user *msgp, size_t msgsz,
                long msgtyp, int msgflg);
long ksys_msgsnd(int msqid, struct msgbuf __user *msgp, size_t msgsz,
                int msgflg);
long ksys_shmget(key_t key, size_t size, int shmflg);
long ksys_shmdt(char __user *shmaddr);
long ksys_old_shmctl(int shmid, int cmd, struct shmid_ds __user *buf);
long compat_ksys_semtimedop(int semid, struct sembuf __user *tsems,
                            unsigned int nsops,
                            const struct old_timespec32 __user *timeout);

int __sys_getsockopt(int fd, int level, int optname, char __user *optval,
                    int __user *optlen);
int __sys_setsockopt(int fd, int level, int optname, char __user *optval,
                    int optlen);

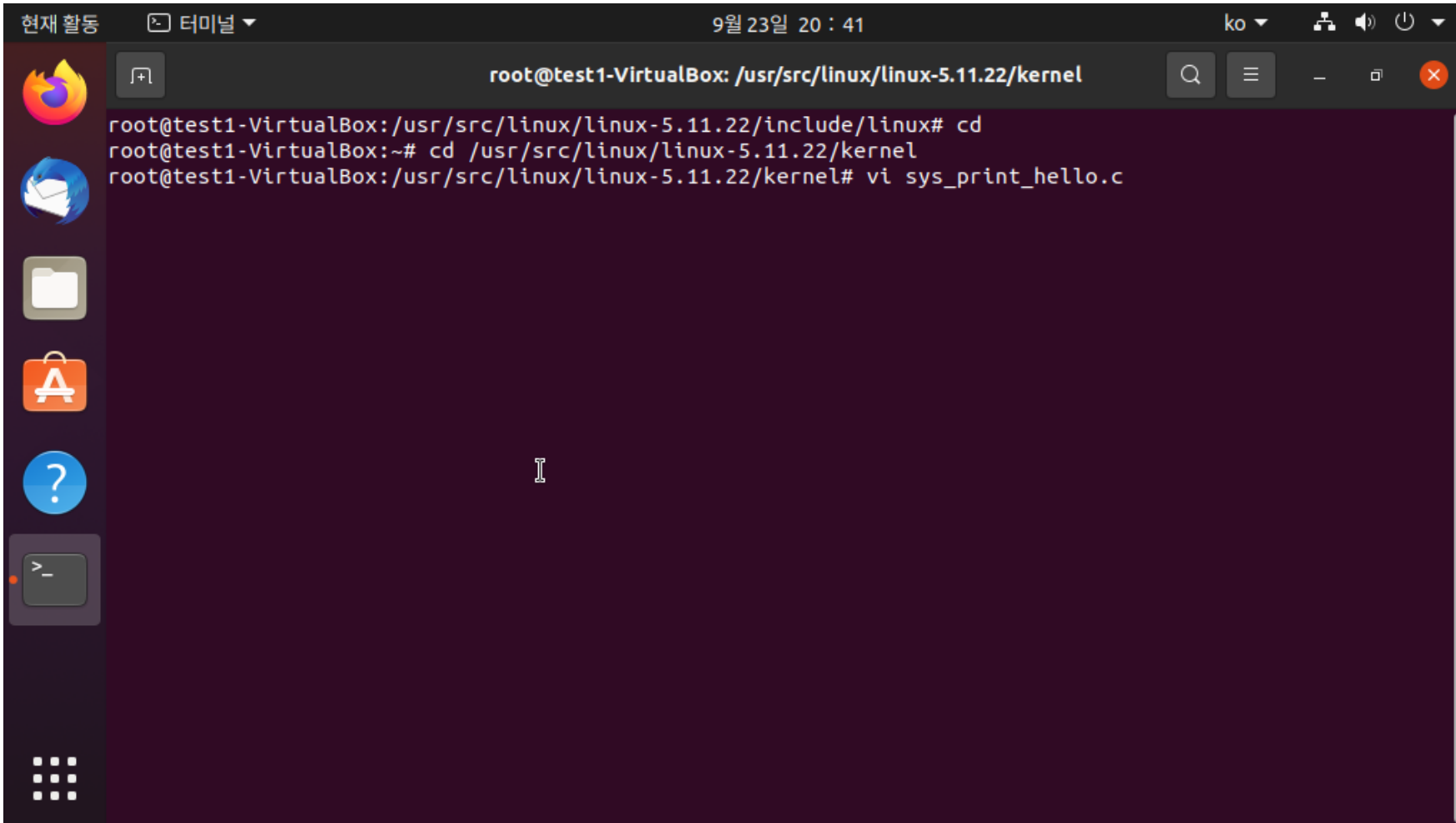
asmlinkage long sys_print_hello(void);

#endif
-- 끼워넣기 --
```

함수의 프로토타입을 정의함

asmlinkage를 앞에 붙힘으로서 어셈블리 코드에서도 C 함수 호출이 가능해짐

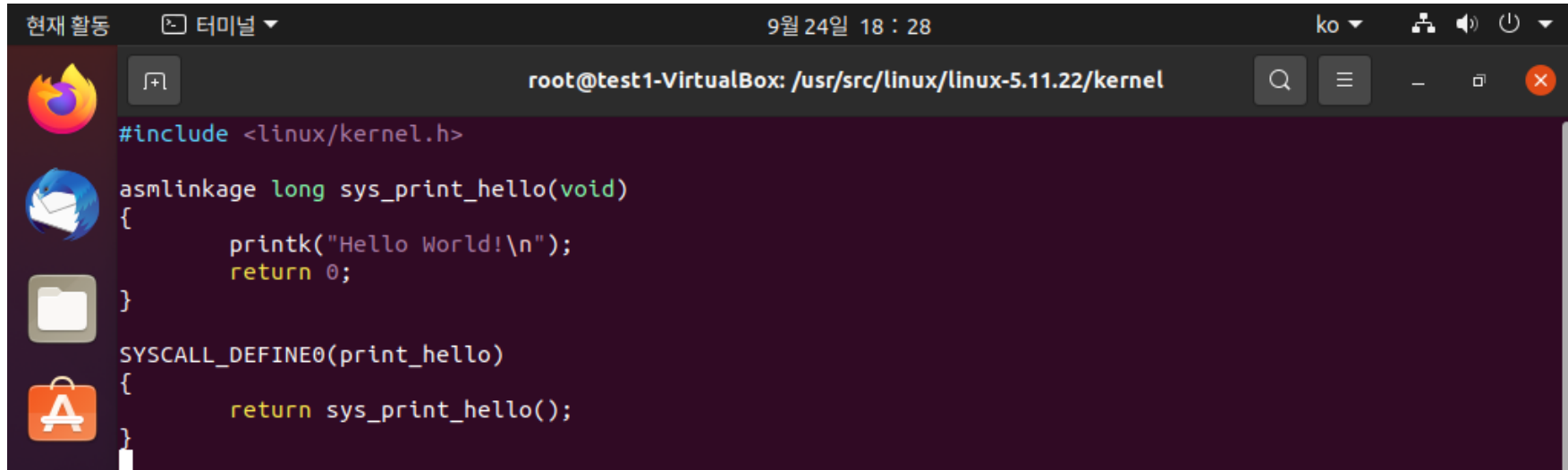
1. 시스템 콜 추가 - 시스템 콜 함수 구현



```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/kernel
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/include/linux# cd
root@test1-VirtualBox:~# cd /usr/src/linux/linux-5.11.22/kernel
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/kernel# vi sys_print_hello.c
```

/usr/src/linux/linux-5.11.22/kernel 디렉토리로 이동 후 추가할 시스템 콜의 구현 파일을 편집

1. 시스템 콜 추가 - 시스템 콜 함수 구현



The screenshot shows a terminal window titled "현재 활동" (Current Activity) with a dropdown menu "터미널" (Terminal). The window displays the command prompt "root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/kernel". The code being edited is as follows:

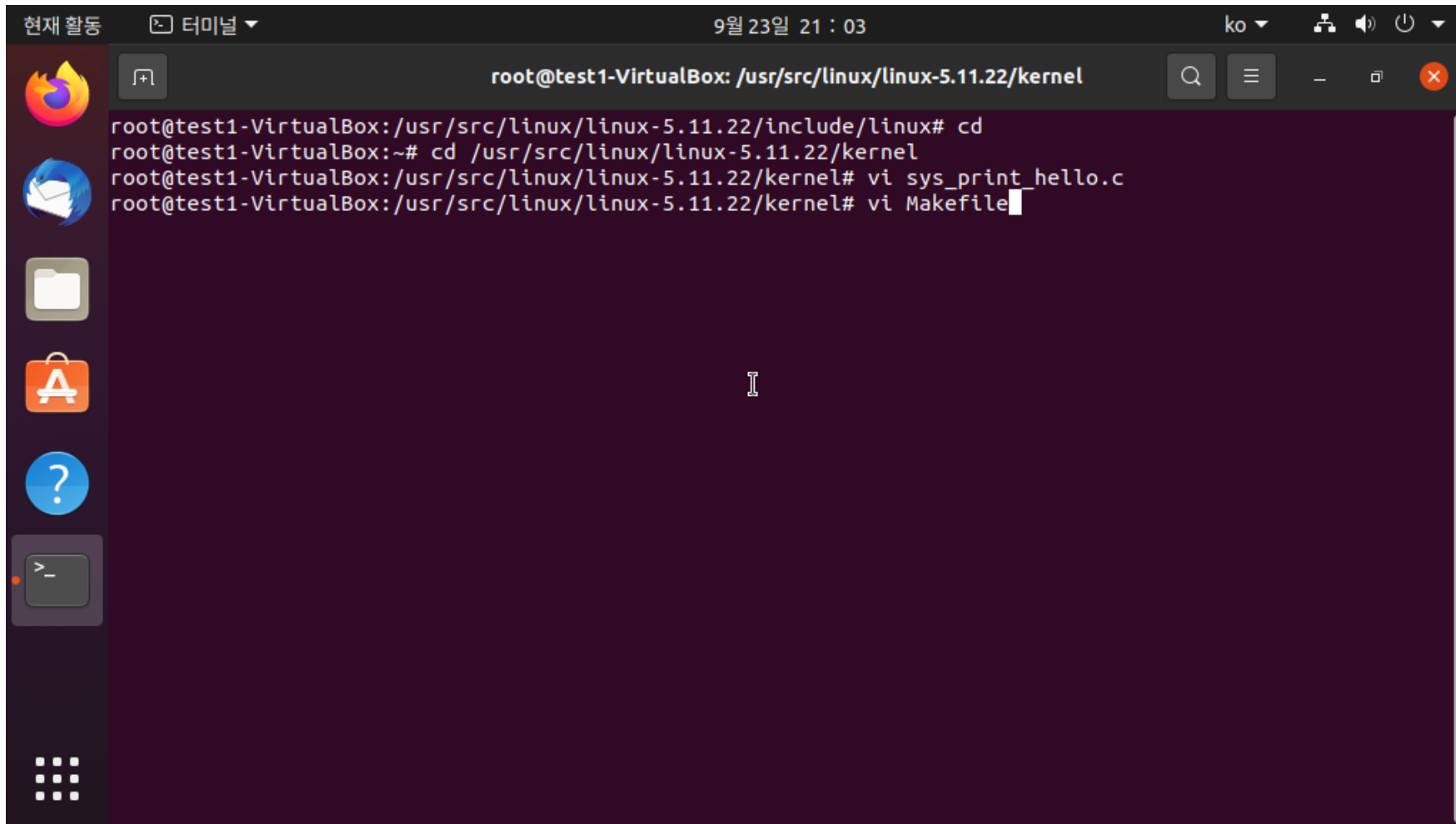
```
#include <linux/kernel.h>

asmlinkage long sys_print_hello(void)
{
    printk("Hello World!\n");
    return 0;
}

SYSCALL_DEFINE0(print_hello)
{
    return sys_print_hello();
}
```

이때 파일명은 시스템 콜 이름과 달라도 되지만 파일 내부의 함수는 `sys_시스템 콜 이름`으로 작성해야 함
`printk` 명령어를 통해 커널 공간에서 메시지 출력이 되도록 작성함

1. 시스템 콜 추가 – Makefile에 등록



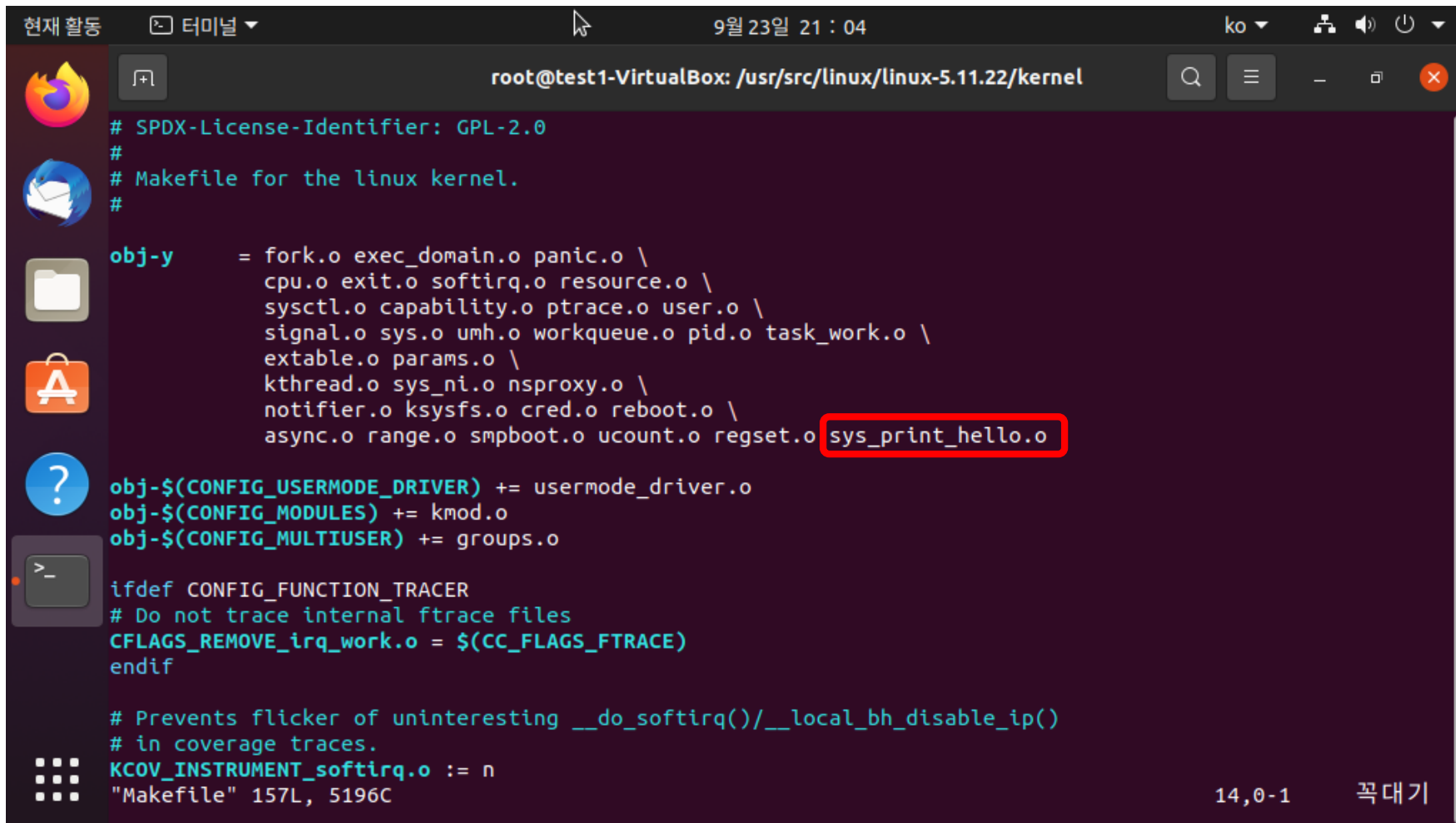
The screenshot shows a terminal window titled "현재 활동" (Current Activity) with a subtitle "터미널" (Terminal). The window displays the following commands and their outputs:

```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/kernel
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/include/linux# cd
root@test1-VirtualBox:~# cd /usr/src/linux/linux-5.11.22/kernel
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/kernel# vi sys_print_hello.c
root@test1-VirtualBox:/usr/src/linux/linux-5.11.22/kernel# vi Makefile
```

The terminal window has a dark background and a light-colored text. The window title bar includes the text "root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/kernel". The window also features a sidebar with various application icons and a search bar.

이전 단계와 동일한 디렉토리 내에서 Makefile 편집

1. 시스템 콜 추가 – Makefile에 등록



```
# SPDX-License-Identifier: GPL-2.0
#
# Makefile for the linux kernel.
#
obj-y      = fork.o exec_domain.o panic.o \
             cpu.o exit.o softirq.o resource.o \
             sysctl.o capability.o ptrace.o user.o \
             signal.o sys.o umh.o workqueue.o pid.o task_work.o \
             extable.o params.o \
             kthread.o sys_ni.o nsproxy.o \
             notifier.o ksysfs.o cred.o reboot.o \
             async.o range.o smpboot.o ucount.o regset.o sys_print_hello.o

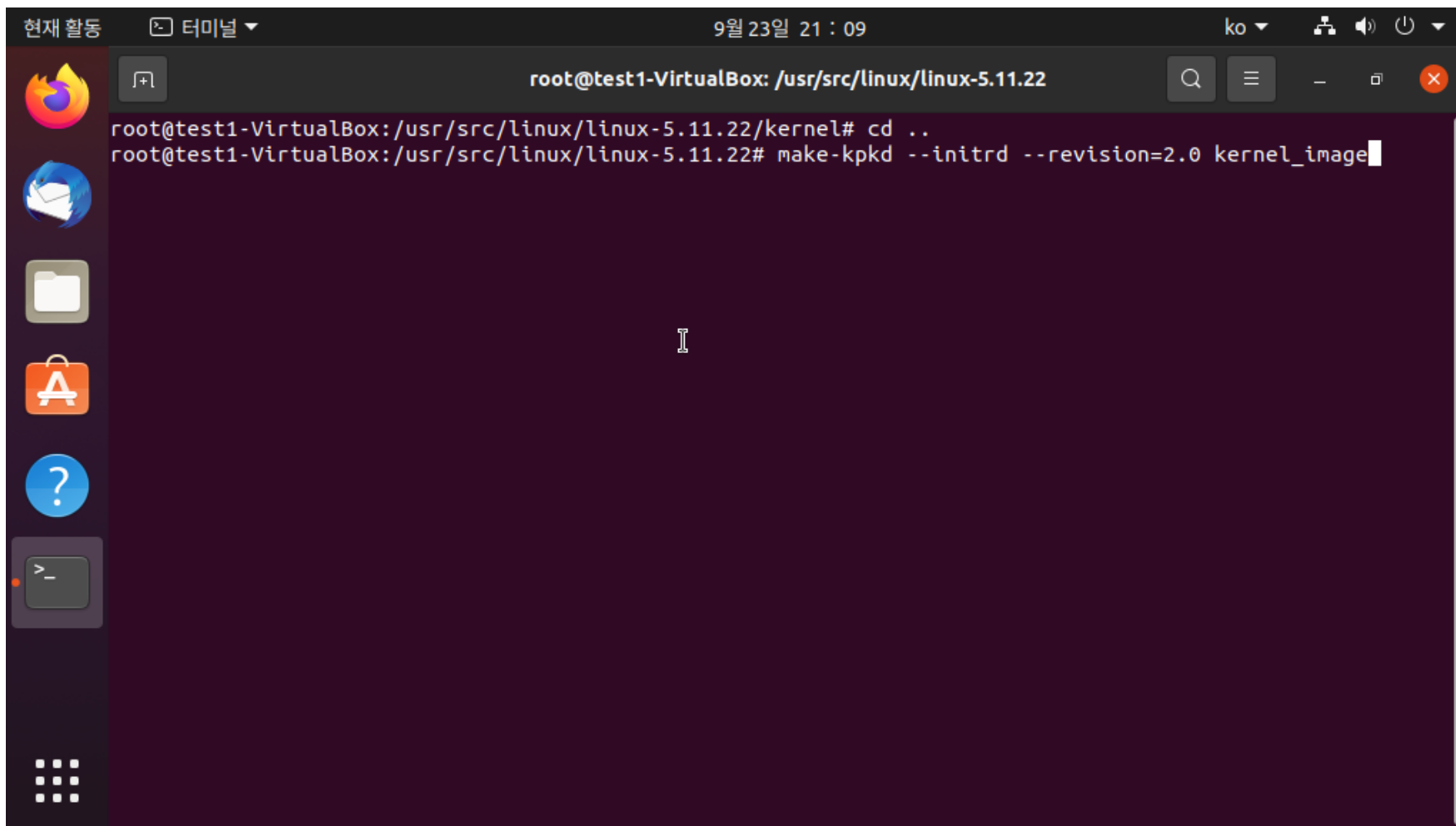
obj-$(CONFIG_USERMODE_DRIVER) += usermode_driver.o
obj-$(CONFIG_MODULES) += kmod.o
obj-$(CONFIG_MULTIVER) += groups.o

ifdef CONFIG_FUNCTION_TRACER
# Do not trace internal ftrace files
CFLAGS_REMOVE_irq_work.o = $(CC_FLAGS_FTRACE)
endif

# Prevents flicker of uninteresting __do_softirq()/__local_bh_disable_ip()
# in coverage traces.
KCOV_INSTRUMENT_softirq.o := n
"Makefile" 157L, 5196C
```

추가한 시스템 콜이 다른 시스템 콜과 함께 컴파일될 수 있도록 편집

1. 시스템 콜 추가 – Makefile에 등록



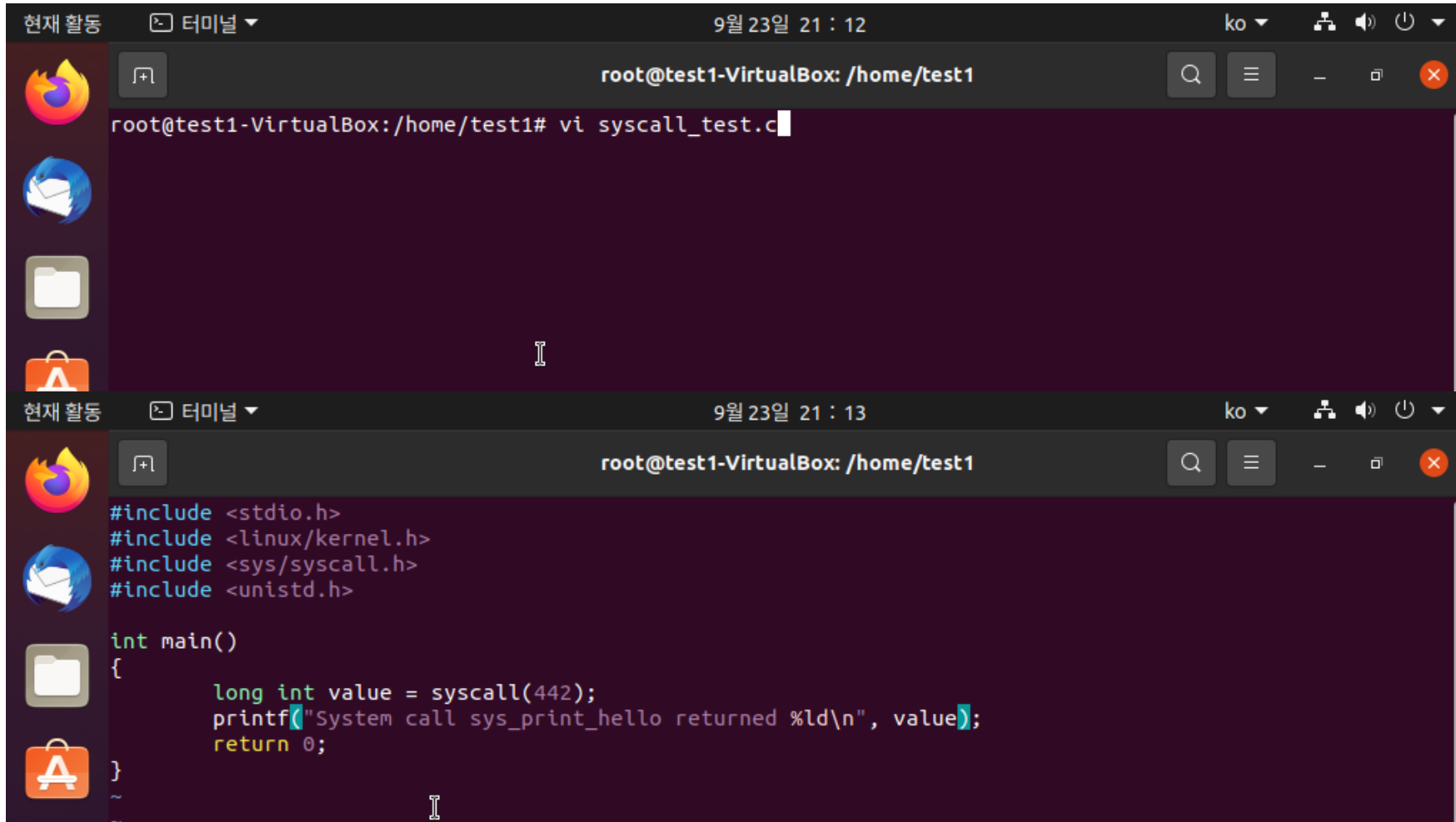
The screenshot shows a terminal window titled 'root@test1-VirtualBox: /usr/src/linux/linux-5.11.22'. The terminal output shows the user navigating to the parent directory and then running the 'make-kpkd' command with options '--initrd' and '--revision=2.0' to build the 'kernel_image'. The terminal has a dark purple background and a light-colored cursor. On the left side of the terminal window, there is a vertical sidebar with several icons: a Firefox logo, a blue bird icon, a folder icon, an App Store icon, a question mark icon, and a terminal icon. At the bottom of the sidebar is a grid of dots. The top of the window has a status bar with '현재 활동', '터미널', the date '9월 23일 21:09', and a language dropdown set to 'ko'.

```
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22/kernel# cd ..
root@test1-VirtualBox: /usr/src/linux/linux-5.11.22# make-kpkd --initrd --revision=2.0 kernel_image
```

커널 소스 디렉토리로 이동하여 새로 컴파일 후 재부팅 실시함

이때 revision의 값은 기존의 1.0과 구분을 할 수 있도록 다른 값을 입력하여 컴파일 실시 (정수 값으로 설정)

2. 추가된 시스템 콜 호출



The image shows two sequential screenshots of a terminal window within a virtual machine environment. The top screenshot shows the terminal prompt `root@test1-VirtualBox: /home/test1` and the command `vi syscall_test.c` being entered. The bottom screenshot shows the same terminal window after the file has been opened in the vi editor. The editor displays the following C code:

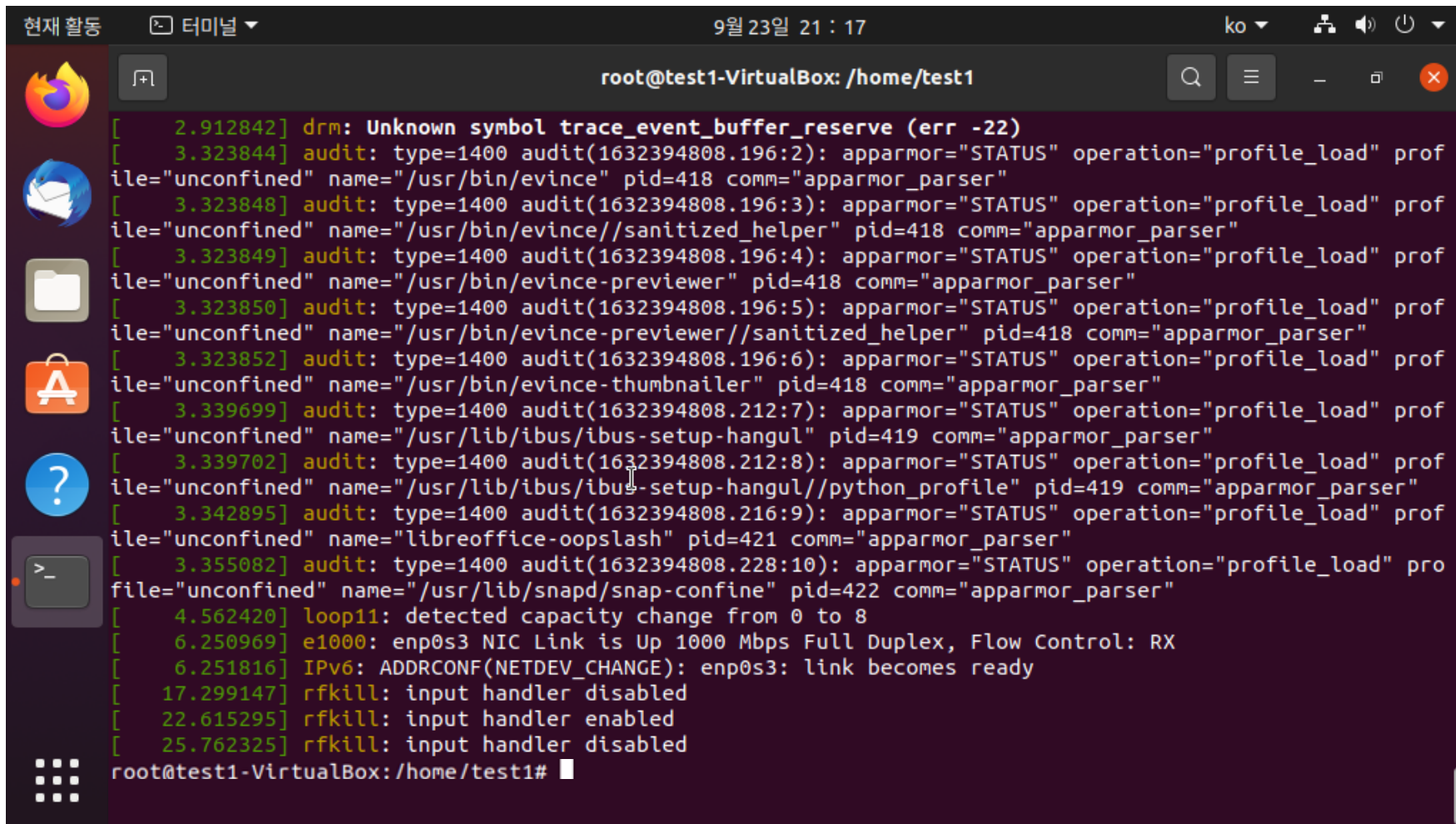
```
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>

int main()
{
    long int value = syscall(442);
    printf("System call sys_print_hello returned %ld\n", value);
    return 0;
}
```

재부팅 후 추가한 시스템 콜을 호출하는 함수를 생성함

앞서 시스템 콜을 테이블에 등록할 때 생성한 번호를 사용하여 호출

2. 추가된 시스템 콜 호출



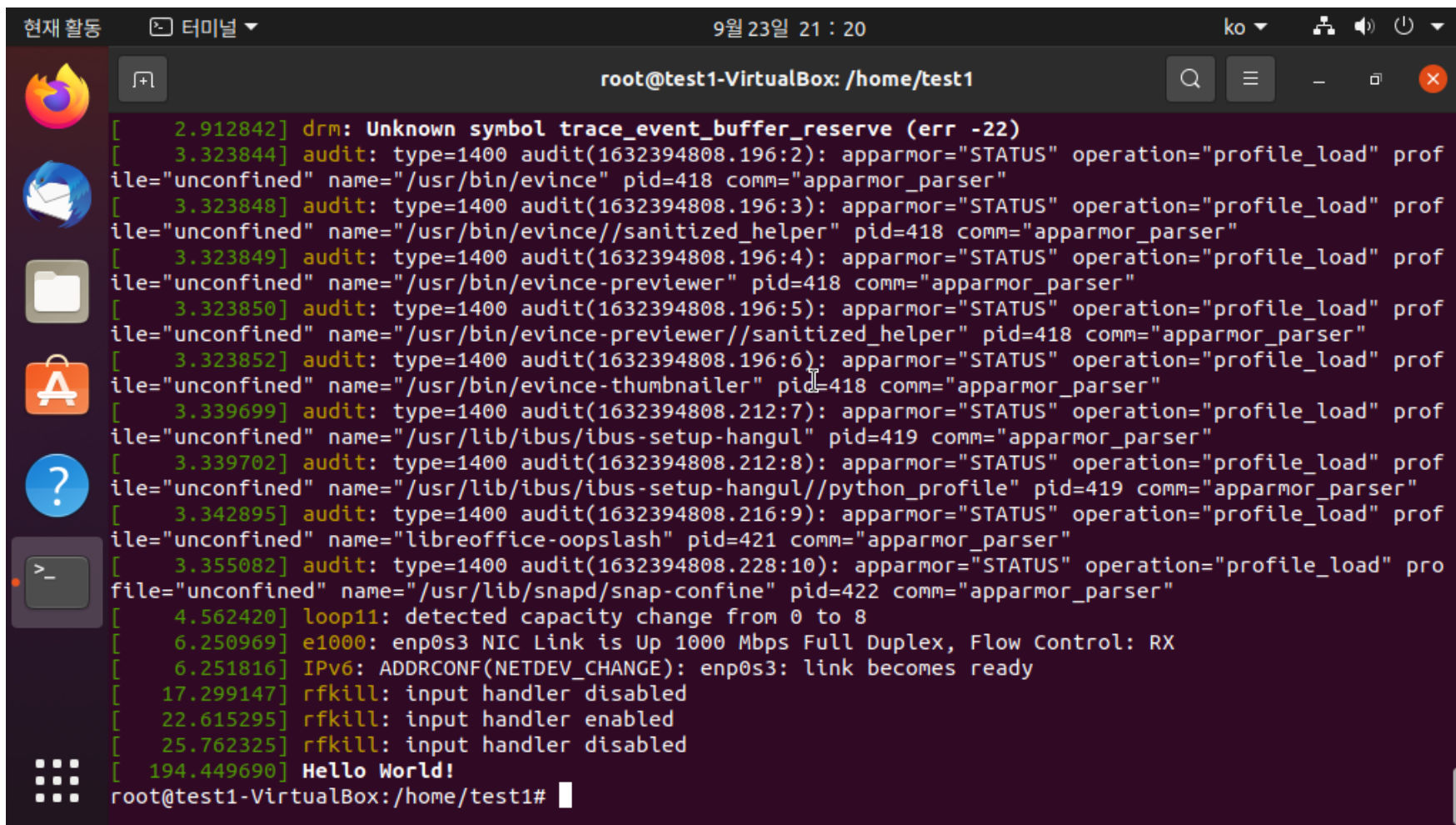
```
현재 활동  터미널  9월 23일 21 : 17  ko  [Search] [Menu] [Window] [Close]

root@test1-VirtualBox: /home/test1

[ 2.912842] drm: Unknown symbol trace_event_buffer_reserve (err -22)
[ 3.323844] audit: type=1400 audit(1632394808.196:2): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince" pid=418 comm="apparmor_parser"
[ 3.323848] audit: type=1400 audit(1632394808.196:3): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince//sanitized_helper" pid=418 comm="apparmor_parser"
[ 3.323849] audit: type=1400 audit(1632394808.196:4): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince-previewer" pid=418 comm="apparmor_parser"
[ 3.323850] audit: type=1400 audit(1632394808.196:5): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince-previewer//sanitized_helper" pid=418 comm="apparmor_parser"
[ 3.323852] audit: type=1400 audit(1632394808.196:6): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/bin/evince-thumbnailer" pid=418 comm="apparmor_parser"
[ 3.339699] audit: type=1400 audit(1632394808.212:7): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/ibus/ibus-setup-hangul" pid=419 comm="apparmor_parser"
[ 3.339702] audit: type=1400 audit(1632394808.212:8): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/ibus/ibus-setup-hangul//python_profile" pid=419 comm="apparmor_parser"
[ 3.342895] audit: type=1400 audit(1632394808.216:9): apparmor="STATUS" operation="profile_load" profile="unconfined" name="libreoffice-oopslash" pid=421 comm="apparmor_parser"
[ 3.355082] audit: type=1400 audit(1632394808.228:10): apparmor="STATUS" operation="profile_load" profile="unconfined" name="/usr/lib/snapd/snap-confine" pid=422 comm="apparmor_parser"
[ 4.562420] loop11: detected capacity change from 0 to 8
[ 6.250969] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 6.251816] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 17.299147] rfkill: input handler disabled
[ 22.615295] rfkill: input handler enabled
[ 25.762325] rfkill: input handler disabled
root@test1-VirtualBox: /home/test1#
```

테스트 프로그램 실행 전 `dmesg` 명령어를 통해 커널의 로그를 출력함

2. 추가된 시스템 콜 호출



```
현재 활동  터미널  9월 23일 21 : 20  ko  [Icons]
root@test1-VirtualBox: /home/test1

[ 2.912842] drm: Unknown symbol trace_event_buffer_reserve (err -22)
[ 3.323844] audit: type=1400 audit(1632394808.196:2): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/bin/evince" pid=418 comm="apparmor_parser"
[ 3.323848] audit: type=1400 audit(1632394808.196:3): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/bin/evince//sanitized_helper" pid=418 comm="apparmor_parser"
[ 3.323849] audit: type=1400 audit(1632394808.196:4): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/bin/evince-previewer" pid=418 comm="apparmor_parser"
[ 3.323850] audit: type=1400 audit(1632394808.196:5): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/bin/evince-previewer//sanitized_helper" pid=418 comm="apparmor_parser"
[ 3.323852] audit: type=1400 audit(1632394808.196:6): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/bin/evince-thumbnailer" pid=418 comm="apparmor_parser"
[ 3.339699] audit: type=1400 audit(1632394808.212:7): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/lib/ibus/ibus-setup-hangul" pid=419 comm="apparmor_parser"
[ 3.339702] audit: type=1400 audit(1632394808.212:8): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="/usr/lib/ibus/ibus-setup-hangul//python_profile" pid=419 comm="apparmor_parser"
[ 3.342895] audit: type=1400 audit(1632394808.216:9): apparmor="STATUS" operation="profile_load" prof
ile="unconfined" name="libreoffice-oopslash" pid=421 comm="apparmor_parser"
[ 3.355082] audit: type=1400 audit(1632394808.228:10): apparmor="STATUS" operation="profile_load" pro
file="unconfined" name="/usr/lib/snapd/snap-confine" pid=422 comm="apparmor_parser"
[ 4.562420] loop11: detected capacity change from 0 to 8
[ 6.250969] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 6.251816] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 17.299147] rfkill: input handler disabled
[ 22.615295] rfkill: input handler enabled
[ 25.762325] rfkill: input handler disabled
[ 194.449690] Hello World!
root@test1-VirtualBox: /home/test1#
```

테스트 프로그램 실행 후 dmesg 명령어를 통해 커널 로그를 출력하면

Hello World! 가 출력된 것을 확인할 수 있음