
운영체제

과제 4. 커널 컴파일 및 시스템 콜

보고서

소속	소프트웨어학과
학번	2015041003
이름	구 경 민
교수님	이 건 명 교수님
제출 날짜	2018.4.13

1. 시스템 콜 함수 정의

sys_kyeongmincall()

cd /usr/src/linux-4.1.51/kernel 로 들어간다.

sudo vi mincall.c

```

kyeongmin@kyeongmin-VirtualBox: /usr/src/linux-4.1.51/kernel
18
19 asmlinkage long kkmSyscall(void){
20
21     struct net_device *device;
22     struct in_device *in_dev;
23     struct in_ifaddr *if_info;
24     __u8 *addr;
25     struct rtc_time tm;
26     struct timex txc;
27     device = dev_get_by_name(&init_net,"enp0s3");
28     in_dev = (struct in_device *)device->ip_ptr;
29     addr = (char *)&if_info->ifa_local;
30     do_gettimeofday(&(txc.time));
31     rtc_time_to_tm(txc.time.tv_sec, &tm);
32     printk(KERN_INFO "Today is : %dyear, %dmonth, %dday, %dhour %dmin %dsec\n",
33             tm.tm_year + 1900, tm.tm_mon+1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec);
34     printk(KERN_WARNING "Get My IP : %u.%u.%u.%u\n", (__u32)addr[0],(__u32)addr[1],
35             (__u32)addr[2],(__u32)addr[3]);
36     printk(KERN_INFO "My name ku kyeong min\n");
37     return 0;
-- 끼워넣기 --                                     36,1      94%

```

sudo vi Makefile

시스템콜 mincall.o를 추가 시켰다.

```

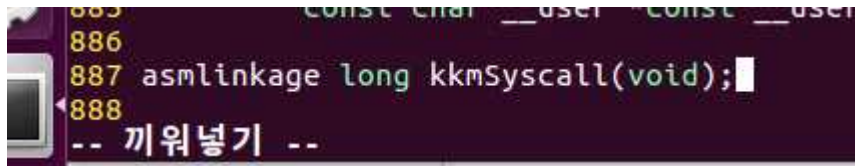
kyeongmin@kyeongmin-VirtualBox: /usr/src/linux-4.1.51/kernel
1 #
2 # Makefile for the linux kernel.
3 #
4
5 obj-y      = fork.o exec_domain.o panic.o \
6             cpu.o exit.o softirq.o resource.o \
7             sysctl.o sysctl_binary.o capability.o ptrace.o user.o \
8             signal.o sys.o kmod.o workqueue.o pid.o task_work.o \
9             extable.o params.o \
10            kthread.o sys_ni.o nsproxy.o \
11            notifier.o ksysfs.o cred.o reboot.o \
12            async.o range.o smpboot.o mincall.o
13

```

cd /usr/src/linux-4.1.51/include/linux

로 이동한뒤 시스템 호출 함수의 헤더파일인 **syscall.h**파일을 수정한다.

sudo vi syscalls.h



syscalls.h파일 마지막 #endif 바로 위에 시스템콜 함수를 추가한다.

2. 커널 컴파일

<중간에 오류 발생함>

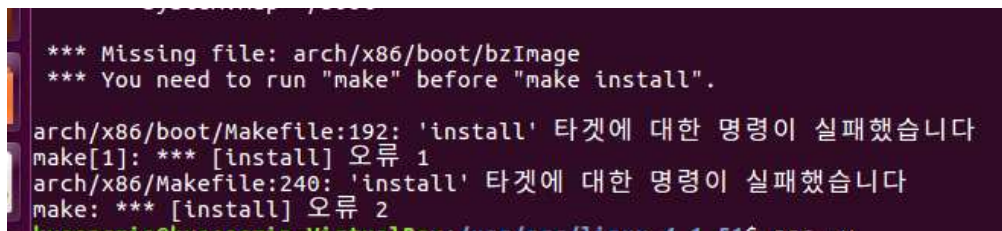
make modules_install install 과정에서 **_modinst_** 오류 발생

1) **cd /usr/src/linux**

2) **make modules** //커널 환경설정에서 모듈로 설정한 기능들을 컴파일한다.

3) **make modules_install** //컴파일 된 모듈을 /lib/modules 아래 설치한다.

위의 방법을 했으나



arch/x86/boot/bzImage 파일이 없다고 뜬다.

sudo make bzImage

명령을 입력해서 압축된 커널 이미지를 만든다.

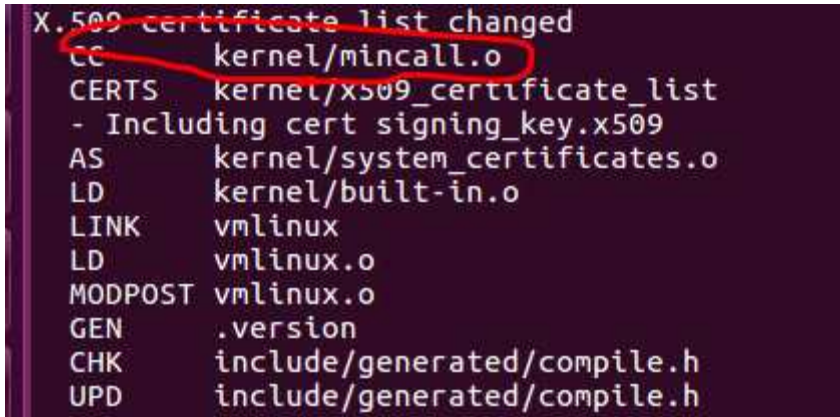
컴파일을 하기 위해 이미지를 만들어야 한다.

성공했다면 /usr/src/linux-4.1.51/arch/x86/i386/boot 디렉토리 아래에 bzImage를 찾을 수 있다.

<오류 수정 후 컴파일>

1) cd /usr/src/linux

1-1) linux폴더에 makefile여부를 확인한 후 **\$sudo make**



```
X.509_certificate_list changed
CC      kernel/mincall.o
CERTS   kernel/x509_certificate_list
- Including cert signing_key.x509
AS      kernel/system_certificates.o
LD      kernel/built-in.o
LINK    vmlinux
LD      vmlinux.o
MODPOST vmlinux.o
GEN     .version
CHK     include/generated/compile.h
UPD     include/generated/compile.h
```

컴파일이 진행 되고 있다.

2)sudo make modules

3)sudo make modules_install

4)sudo make install

재부팅 하기전에grub을 띄워준다

깔아놓은 컴파일을 선택하기 위한 설정이다

구글 참고

<https://www.youtube.com/watch?v=2MK7Z19rMn8>

grub띄우는 과정에서 어려움이 있었다. 컴파일 후 reboot만 해줬더니 컴파일 버전이 바뀌지 않았다.

위의 링크를 통해 grub 띄우기 설정에 들어가서 재부팅시 grub창이 나오게 바꿨다.

여러 가지 커널 버전들이 많이 깔린 상태라 용량이 부족했다.

기존버전인 4.13.0-38

새로 다운받은 4.15.1

두 개 빼고 삭제 했다.

\$dpkg --get-architecture linux-image

커널 이미지의 리스트가 뜬다.

sudo apt-get purge linux-image-4.13.0-36* //4.13.0.36버전의 커널을 삭제한다.

삭제 후

sudo apt-get autoremove 로 남은 불필요한 파일도 자동삭제한다.
update-grub (혹은 update-grub2)

<grub 설정시 패닉오류: out of memory>가 발생했다.

원인: /boot/initrd.img 파일이 너무 커서 발생하는 오류

<해결법>

cd /lib/modules/4.1.51
sudo find . -name *.ko -exec strip --strip-unneeded {} +
sudo update-initramfs -c -k 4.1.51

이미지 압축 후 reboot 한다.

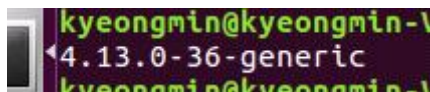
재부팅화면에서 grub설정이 뜨면
우분투 설정용-> 4.1.51 버전으로 들어간다.

재부팅 완료.

<비교>

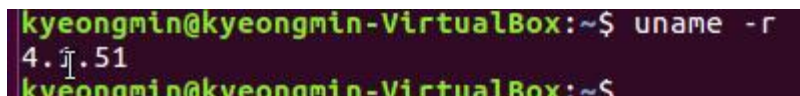
재부팅 전 커널의 원래 버전을 확인한다

uname -r



```
kyeongmin@kyeongmin-VirtualBox:~$ uname -r
4.13.0-36-generic
```

부팅 후



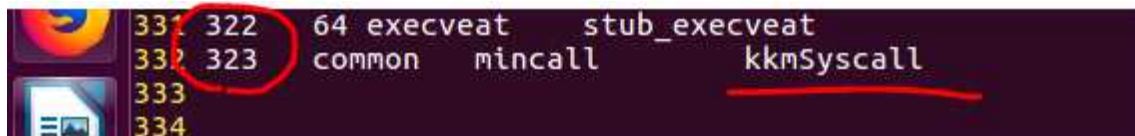
```
kyeongmin@kyeongmin-VirtualBox:~$ uname -r
4.15.1
```

3. 시스템콜 호출하기

cd /usr/src/linux-4.1.51/arch/x86/syscalls

vi syscall_64.tbl을 열어서

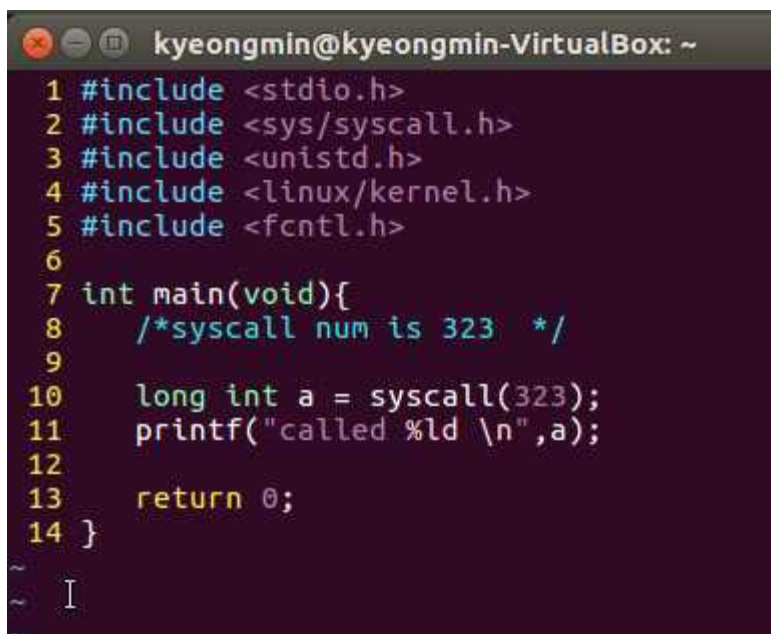
323 확인한다.



323	322	64	execveat	stub_execveat	
323	323	common	mincall		kkmSyscall
323					
323					

시스템 콜을 하기위한 소스코드를 짤다

vi syscall_test.c



```
kyeongmin@kyeongmin-VirtualBox: ~
1 #include <stdio.h>
2 #include <sys/syscall.h>
3 #include <unistd.h>
4 #include <linux/kernel.h>
5 #include <fcntl.h>
6
7 int main(void){
8     /*syscall num is 323 */
9
10    long int a = syscall(323);
11    printf("called %ld \n",a);
12
13    return 0;
14 }
~
~ I
~
```

gcc를 이용하여 컴파일 및 빌드한다.

```
kyeongmin@kyeongmin-VirtualBox:~$ vi syscall_test.c
kyeongmin@kyeongmin-VirtualBox:~$ gcc syscall_test.c
kyeongmin@kyeongmin-VirtualBox:~$ ls a.out
a.out
kyeongmin@kyeongmin-VirtualBox:~$ ./a.out
called -1
kyeongmin@kyeongmin-VirtualBox:~$
```

4. 시스템콜 호출하기

컴파일 후 dmesg 입력하면

```
[ 694.169329] RX
[ 694.169329] Today is : 2018year, 4month, 12day, 18hour 16min 19sec
[ 694.169333] Get My IP : 10.0.2.15
[ 694.169335] My name ku kyeong min
kyeongmin@kyeongmin-VirtualBox:~$
```