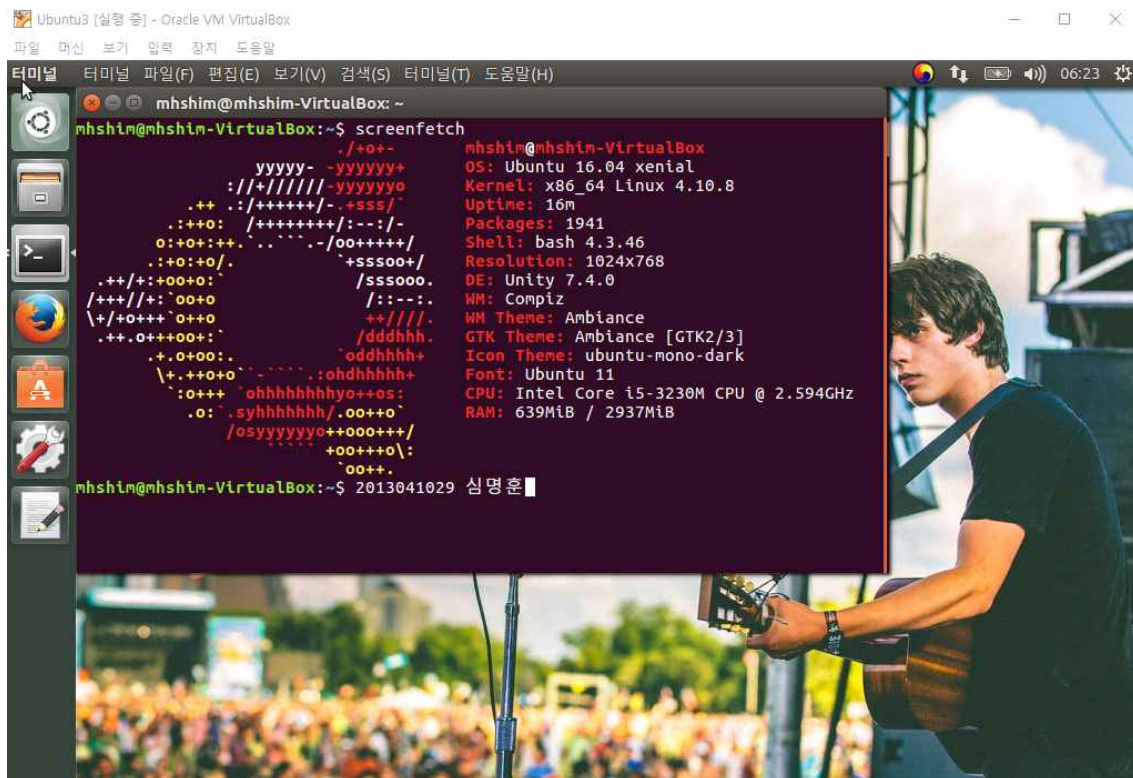


# 운영체제 리눅스 시스템 '콜'

2013041029 심명훈



```
mhshim@mhshim-VirtualBox: ~  
mhshim@mhshim-VirtualBox:~$ screenfetch  
      ./+0+/-  
      yyyyy- -yyyyyy+  
      ://+///// -yyyyyyo  
      .++ .:/++++++/- .+sss/  
      .:++o: /+++++++/:-:-/-  
      o:+o+:+. .-.-.-.-/oo+++++/  
      .:o+:o+/.      +sssoo+/  
      .++/+:+oo+o:      /sssooo.  
      /+++//+:`oo+o      /:-:-:.  
      \+/+o+++`o+o+      ++///.  
      .++.o+++oo+:`      /dddhhh.  
      .+.o+oo:      .oddhhhh+  
      \+.+o+o`-.-.-.-.-:ohdhhhh+  
      .o+++`ohhhhhhhhyo++os:  
      .o: .syhhhhhhh/.oo++o  
      /osyyyyyyo++ooo+++/  
      .+++++o\:  
      oo++.  
mhshim@mhshim-VirtualBox:~$ 2013041029 심명훈
```

mhshim@mhshim-VirtualBox  
OS: Ubuntu 16.04 xenial  
Kernel: x86\_64 Linux 4.10.8  
Uptime: 16m  
Packages: 1941  
Shell: bash 4.3.46  
Resolution: 1024x768  
DE: Unity 7.4.0  
WM: Compiz  
WM Theme: Ambiance  
GTK Theme: Ambiance [GTK2/3]  
Icon Theme: ubuntu-mono-dark  
Font: Ubuntu 11  
CPU: Intel Core i5-3230M CPU @ 2.594GHz  
RAM: 639MiB / 2937MiB

시스템콜 추가하는 과정

1. 시스템 콜 번호 등록
2. 시스템 콜 함수 등록
3. 시스템 콜 처리 함수 구현
4. 커널 컴파일, 설치 및 리부팅
5. 새로운 시스템 콜을 사용하는 프로그램 작성 및 테스트

VIM 설치

```
$ sudo apt-get install vim
```

VIM 설정

```
$ vi .vimrc
```

환경 설정 값

```
set number // 행 번호 표시하기
```

```

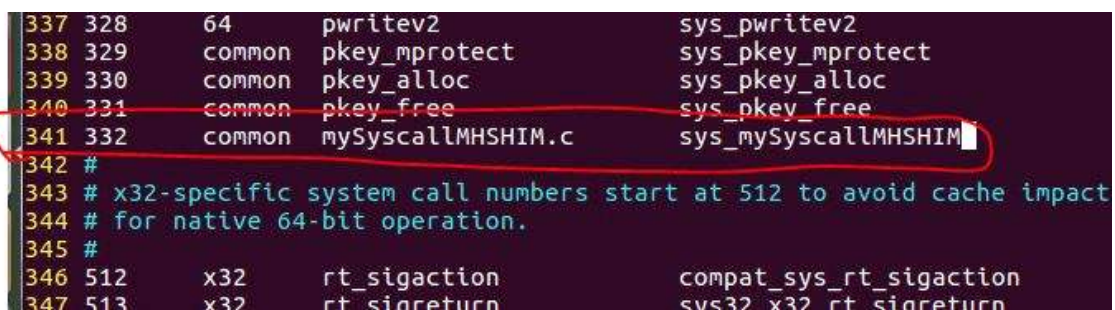
set smartindent // 스마트 들여쓰기
set autoindent // 자동 들여쓰기
set cindent // C언어 들여쓰기
set title // 현재 편집중인 파일 이름 표시하기
syntax on // 구문 강조 켜기

```

:!wq //저장 후 종료

출처: <http://blog.min0628.com/104> [Min]

---



337	328	64	pwritev2	sys_pwritev2
338	329	common	pkey_mprotect	sys_pkey_mprotect
339	330	common	pkey_alloc	sys_pkey_alloc
340	331	common	pkey_free	sys_pkey_free
341	332	common	mySyscallMHSHIM.c	sys_mySyscallMHSHIM
342	#			
343	# x32-specific system call numbers start at 512 to avoid cache impact			
344	# for native 64-bit operation.			
345	#			
346	512	x32	rt_sigaction	compat_sys_rt_sigaction
347	513	x32	rt_sigreturn	sys32_x32_rt_sigreturn

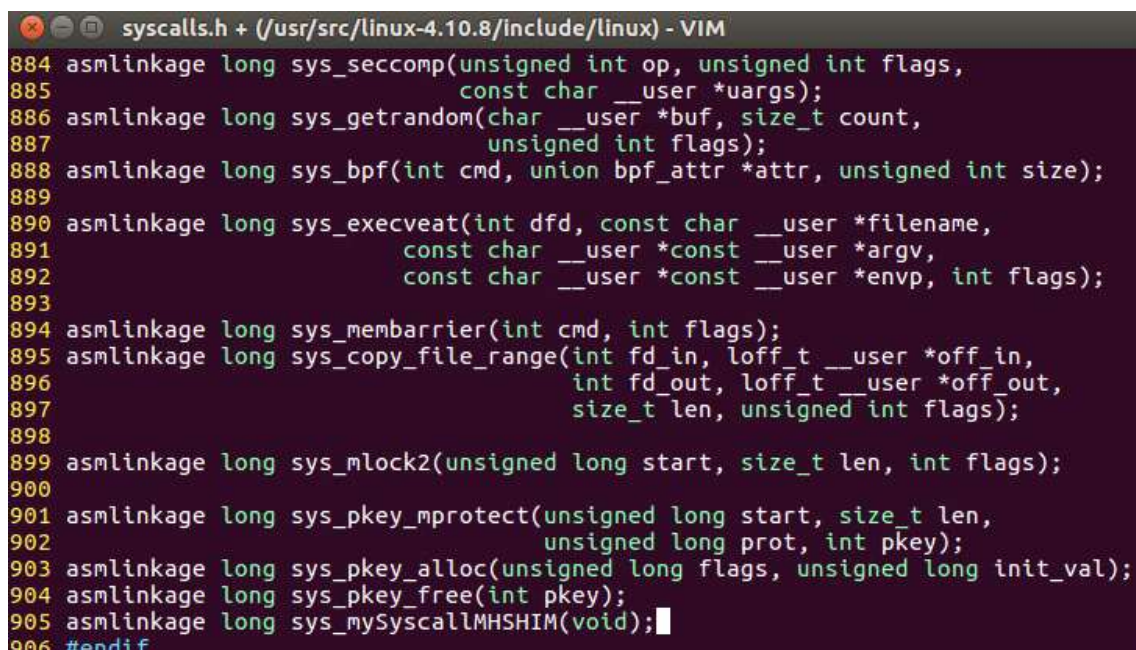
시스템 테이블에 호출 번호 등록

```

$ cd /usr/src/linux-4.5/arch/x86/entry/syscalls
$ vi syscall_64.tbl

```

출처: <http://blog.min0628.com/184#recentTrackback> [Min]



```

884 asmlinkage long sys_seccomp(unsigned int op, unsigned int flags,
885                             const char __user *uargs);
886 asmlinkage long sys_getrandom(char __user *buf, size_t count,
887                               unsigned int flags);
888 asmlinkage long sys_bpf(int cmd, union bpf_attr *attr, unsigned int size);
889
890 asmlinkage long sys_execveat(int dfd, const char __user *filename,
891                             const char __user *const __user *argv,
892                             const char __user *const __user *envp, int flags);
893
894 asmlinkage long sys_membarrier(int cmd, int flags);
895 asmlinkage long sys_copy_file_range(int fd_in, loff_t __user *off_in,
896                                     int fd_out, loff_t __user *off_out,
897                                     size_t len, unsigned int flags);
898
899 asmlinkage long sys_mlock2(unsigned long start, size_t len, int flags);
900
901 asmlinkage long sys_pkey_mprotect(unsigned long start, size_t len,
902                                   unsigned long prot, int pkey);
903 asmlinkage long sys_pkey_alloc(unsigned long flags, unsigned long init_val);
904 asmlinkage long sys_pkey_free(int pkey);
905 asmlinkage long sys_mySyscallMHSHIM(void);
906 #endif

```

## 시스템 호출 함수 등록

```
mySyscallMHSHIM.c (/usr/src/linux-4.10.8/kernel) - VIM
1 #include<linux/linkage.h>
2 #include<linux/kernel.h>
3 #include<linux/time.h>
4 #include<linux/ktime.h>
5 #include<linux/timekeeping.h>
6
7 asmlinkage long sys_mySyscallMHSHIM(void){
8
9     unsigned long current_time;
10    struct tm tm_time;
11
12    current_time = get_seconds();
13    time_to_tm(current_time, 32400, &tm_time);
14
15    printk(KERN_EMERG "My Name is Myung Hun\n");
16    printk(KERN_EMERG "IP address : 203.255.71.198\n");
17    printk(KERN_EMERG "%ld년 %ld월 %ld일 %d시 %d분 %d초\n",
18    tm_time.tm_year + 1900, tm_time.tm_mon+1, tm_time.tm_mday, tm_time.t
19    m_hour, tm_time.tm_min, tm_time.tm_sec);
20    printk(KERN_EMERG "I'm doing OS homework hahaha\n");
```

## 시스템 콜 작성하기

### 커널 컴파일 및 빌드

\$apt-get install gcc

\$apt-get install libncurses5-dev

\$apt-get update

\$apt-get upgrade

커널로 이동

\$cd /usr/src/linux-4.10.8

\$make defconfig

### 커널 설치

1. \$make

2. \$make modules\_install install

3. 재부팅 전 \$uname -r 로 현재 커널 버전 확인

```
root@mhshim-VirtualBox:/usr/src/linux-4.10.8# uname -r
4.8.0-36-generic
```

4. \$shutdown -r now 로 재부팅

5. \$uname -r 로 변경된 커널 버전 확인

```
mhshim@mhshim-VirtualBox:~$ uname -r
4.10.8
```

**GOOOOOD!!!**

/\*시스템 콜 수정 후 커널 경로로 이동한다.

```
$cd /usr/src/linux-4.10.8
```

```
$make defconfig
```

```
$make modules_install install //커널 설치
```

```
$uname -r //커널 버전 확인
```

```
$shutdown -r now //재부팅
```

```
$uname -r //재부팅 후 커널 버전 확인*/
```

시스템 콜 호출하기

```
vi test.c
```

```
#include<stdio.h>
#include<sys/syscall.h>

int main(void){
    long int a = syscall(332);
    printf("Syscall : %ld", a);
    return 0;
}
```

호출 프로그램 컴파일

```
$gcc test.c
```

```
$/a.out
```

```
$dmesg
```

("display message" 또는 "driver message"를 의미)는 대부분의 리눅스와 유닉스 기반 운영 체제에 있는 명령어로 **커널의 메시지 버퍼를 출력**한다.

root 권한으로 들어가서 수정해야 한다!!



배너 메시지를 변경한다.

## 로그인 공지사항

로그인시에 보여줄 메시지를 설정할 수 있다.

### 설정파일

설정파일	설명	상태
/etc/issue	콘솔(로컬) 접속시도시 보여줄 메시지파일	로그인 전
/etc/issue.net	원격지에서 접속시도시 보여줄 메시지파일	로그인 전
/etc/motd	로컬접속, 원격접속 모두에 해당하며 로그인 성공 후 보여줄 메시지 파일	로그인 후

이를 이용하여 접속시도시에는 "불법적인 접근을 허용하지 않습니다" 라는 메시지들을 출력할 수 있고, 로그인 성공 후에는 각 사용자별 현재 디렉토리 사용량들이나 공지사항을 출력할 수 있다.

/etc/issue.net 이나 /etc/issue 의 내용을 작성해도 ssh 를 사용해 로그인 한다면 작동하지 않는데 그럴 경우 /etc/ssh/sshd\_config 를 수정해야 한다.

BASH

```
vi /etc/ssh/sshd_config  
  
# no default banner path  
Banner /etc/issue.net
```

ssh를 재시작 한다.

BASH

```
/etc/init.d/sshd restart
```



Putty 를 이용해 SSH 로 접속시에는 옵션에서 창 > 변환 에서 수신한 데이터를 이 문자셋으로 가정(R): 의 드롭다운 메뉴에서 UTF-8 을 선택해서 접속해야 한글 문자셋이 제대로 출력된다.

BASH

```
vi /etc/issue  
  
+-----+  
| 불법적인 접근 금지 |  
+-----+
```

BASH

```
vi /etc/issue.net  
  
+-----+  
| 불법적인 접근 금지 |  
+-----+
```

BASH

```
vi /etc/motd  
  
+-----+  
| 환영합니다. |  
+-----+
```

# 느낀점

일주일 간 날마다 틈이 나면 노트북을 붙들고 리눅스 커널에 들어가 과제를 해내려고 씨름했다. 1, 2학년 때 참여했던 수업의 내용을 거의 기억하지 못한 채 과제를 수행해서 답답한 심정이 컸다. 구글과 네이버 등을 뒤지며 커널 경로를 따라 파일들을 수정해나갔다. 이러한 과정에서 조상님 탓인지 필연적으로 커널을 새로 깔거나 우분투 자체를 새로 깔아버리는 일이 생겼다. 나라를 잃은 심정이었지만 이내 마음을 가라앉히고 재도전하였다. 주말을 맞이하며 밤을 샌 뒤에 드디어 시스템 콜을 할 수 있게 되었다.

실패에서 교훈을 얻었다. 만약 한 번에 커널 컴파일에 성공하고 시스템콜을 해냈다면 내가 해낸 일들을 금새 잊어버리게 된다. 일주일 내내 괴롭고도 즐거웠다. 뜻대로 되지 않아 괴로웠고 그 와중에 배워가는 것이 있어 즐겁기도 했다. 정신이 몽롱하고 침대가 생각난다.

올해는 이전과는 다르게 과제가 어렵거나 많아도 긍정적인 마음으로 대하게 된다. 알고자 하는 의지가 있기 때문이다.

교수님 말씀이 맞다. 지금 아니면 언제 리눅스 커널 컴파일을 해보겠나... 정신없지만 알찬 일주일이었다.