

# 운영체제론 실습 2주차

정보보호연구실 @ 한양대학교

# 실습 조교 소개

- 연구실 : 제 3공학관 519-2호 정보보호연구실
- 전화번호 : 031-400-4037
- 실습조교정보
  - 김예은: gikosei@hanyang.ac.kr
  - 유동민: ehdals6@hanyang.ac.kr

# 커널 컴파일

1. 패키지 설치
2. Kernel 압축본 다운로드
3. Kernel 컴파일
4. 모듈 프로그래밍

# 패키지 설치

리눅스용 소프트웨어들은 대부분 인터넷에서 다운로드 하게 되어 있다.  
대개 패키지 파일 형태거나 직접 설치할 수 있도록 소스 코드 형태로 제공한다.

## • 패키지란?

- 커널 및 라이브러리 버전의 배포판 환경에 맞추어 빌드한 실행파일을 압축한 것

## • 패키지 관리 시스템

- 저수준 툴은 패키지 파일을 설치하고 삭제하는 작업을 관리하고 고수준 툴은 메타데이터 검색 및 의존성 문제 해결과 같은 작업을 수행함

배포판	저수준 도구	고수준 도구
데비안 형식	Dpkg	Apt-get, aptitude
페도라, 레드햇, CentOS	rpm	yum

# 패키지 설치

## 1) 설치되어 있는 패키지를 모두 새 버전으로 업그레이드

- `$ sudo apt-get update`

## 2) 커널을 컴파일에 필요한 패키지 설치

- `$ sudo apt-get install build-essential libncurses5 libncurses5-dev bin86 kernel-package libssl-dev libelf-dev flex bison -y`

```
Modified configuration file
kernel-img.conf: A new version (/etc/kernel-img.conf.ucf-new) of
configuration file /etc/kernel-img.conf is available, but the version
installed currently has been locally modified.

What do you want to do about modified configuration file
kernel-img.conf?

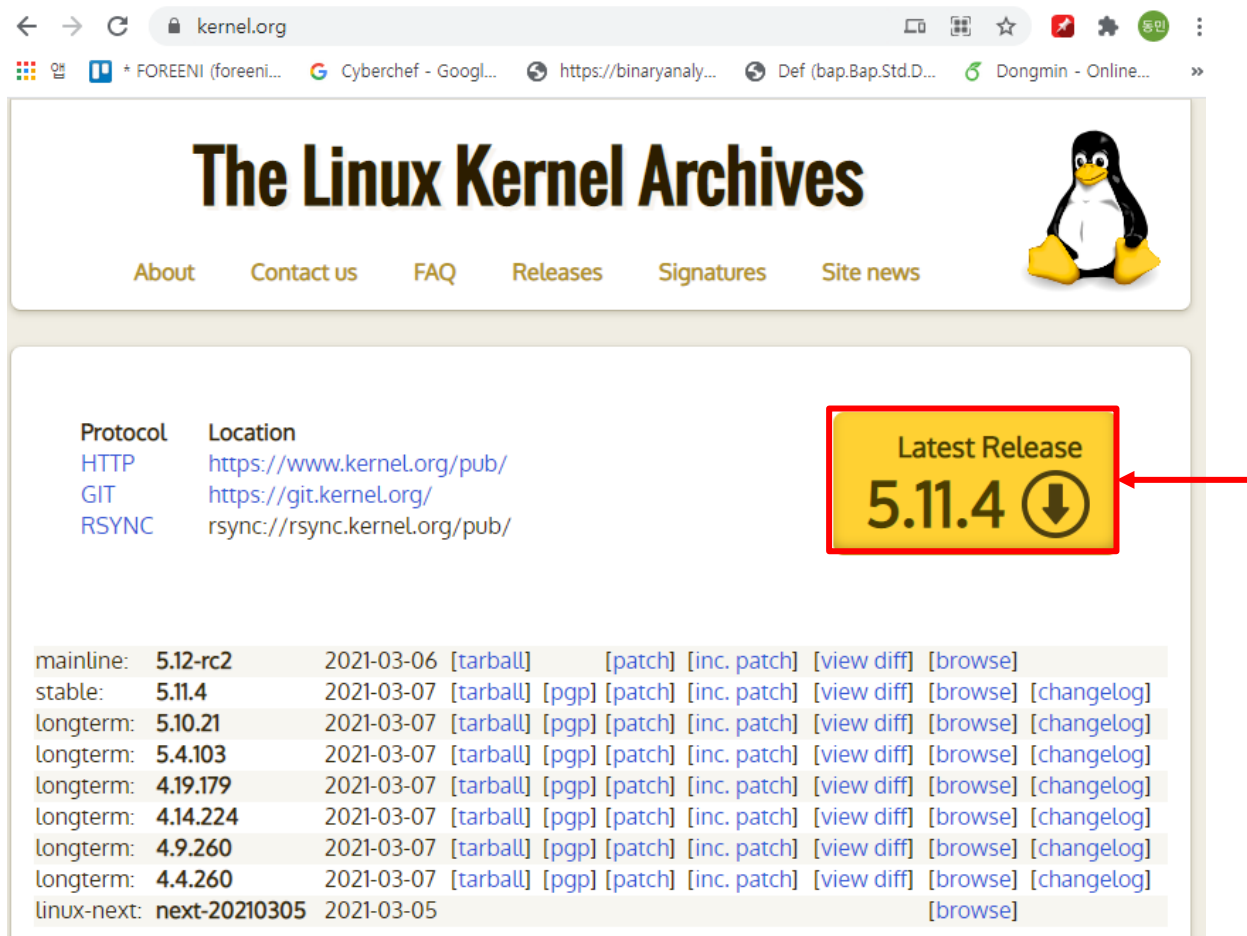
install the package maintainer's version
keep the local version currently installed
show the differences between the versions
show a side-by-side difference between the versions
start a new shell to examine the situation

<Ok>
```

해당 화면이 나올 경우  
디폴트로 되어 있는  
옵션으로 설정


# Kernel 압축본 다운로드

- <https://www.kernel.org/>



The screenshot shows the 'The Linux Kernel Archives' website. The header includes the title and a navigation menu with links: About, Contact us, FAQ, Releases, Signatures, and Site news. A Tux penguin logo is on the right. Below the header, there is a table with download protocols and locations, and a 'Latest Release' box highlighting version 5.11.4 with a download icon. A red arrow points to this box. Below these, a table lists various kernel versions and their corresponding download links.

	Protocol	Location
	HTTP	<a href="https://www.kernel.org/pub/">https://www.kernel.org/pub/</a>
	GIT	<a href="https://git.kernel.org/">https://git.kernel.org/</a>
	RSYNC	<a href="rsync://rsync.kernel.org/pub/">rsync://rsync.kernel.org/pub/</a>

**Latest Release**  
**5.11.4** 

	Version	Date	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
mainline:	5.12-rc2	2021-03-06	<a href="#">[tarball]</a>		<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	
stable:	5.11.4	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	5.10.21	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	5.4.103	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	4.19.179	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	4.14.224	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	4.9.260	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
longterm:	4.4.260	2021-03-07	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	<a href="#">[changelog]</a>
linux-next:	next-20210305	2021-03-05						<a href="#">[browse]</a>	

# 원하는 버전의 Kernel 압축본 다운로드

- \$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.11.4.tar.xz

원하는 버전

```
os@os-VirtualBox:~/Desktop/2week$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.11.4.tar.xz
--2021-03-10 09:08:52-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.11.4.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.77.176, 2a04:4e42:12::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.77.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117631532 (112M) [application/x-xz]
Saving to: 'linux-5.11.4.tar.xz'

linux-5.11.4.tar.xz 100%[=====>] 112.18M  12.8MB/s   in 8.5s

2021-03-10 09:09:00 (13.1 MB/s) - 'linux-5.11.4.tar.xz' saved [117631532/117631532]

os@os-VirtualBox:~/Desktop/2week$ ls
linux-5.11.4.tar.xz
```

# Kernel 컴파일(1)

## 1) 커널 압축 해제

- \$ tar xvf [압축파일명]

```
os@os-VirtualBox:~/Desktop/2week$ tar xvf linux-5.11.4.tar.xz
linux-5.11.4/
linux-5.11.4/.clang-format
linux-5.11.4/.cocciconfig
linux-5.11.4/.get_maintainer.ignore
linux-5.11.4/.gitattributes
```

## 2) 해당 디렉토리로 이동

- \$ cd [linux-5.x.x]

```
os@os-VirtualBox:~/Desktop/2week$ ls
linux-5.11.4  linux-5.11.4.tar.xz
os@os-VirtualBox:~/Desktop/2week$ cd linux-5.11.4/
os@os-VirtualBox:~/Desktop/2week/linux-5.11.4$ pwd
/home/os/Desktop/2week/linux-5.11.4
```



## Kernel 컴파일(2)

### 3) 현재 커널의 버전을 확인 그 버전의 config파일을 복사

- \$ uname -r

```
os@os-VirtualBox:~/Desktop/os/linux-5.11.4$ uname -r  
Linux 5.8.0-44-generic x86_64
```

↑  
현재 Linux버전

### 4) 현재 커널 버전의 config 파일을 복사

- \$ sudo cp -v /boot/config-\$(uname -r) .config

```
os@os-VirtualBox:~/Desktop/2week/linux-5.11.4$ sudo cp -v /boot/config-$(uname -r) .config  
'/boot/config-5.8.0-44-generic' -> '.config'
```

# Kernel 컴파일(3)

## 5) Menuconfig를 통해 빌드할 커널의 설정을 한다.

- \$ sudo make menuconfig

→ 이 경우 파란색 창이 뜨면서 메뉴를 선택할 수 있다.

1) LOAD -> [.config] OK

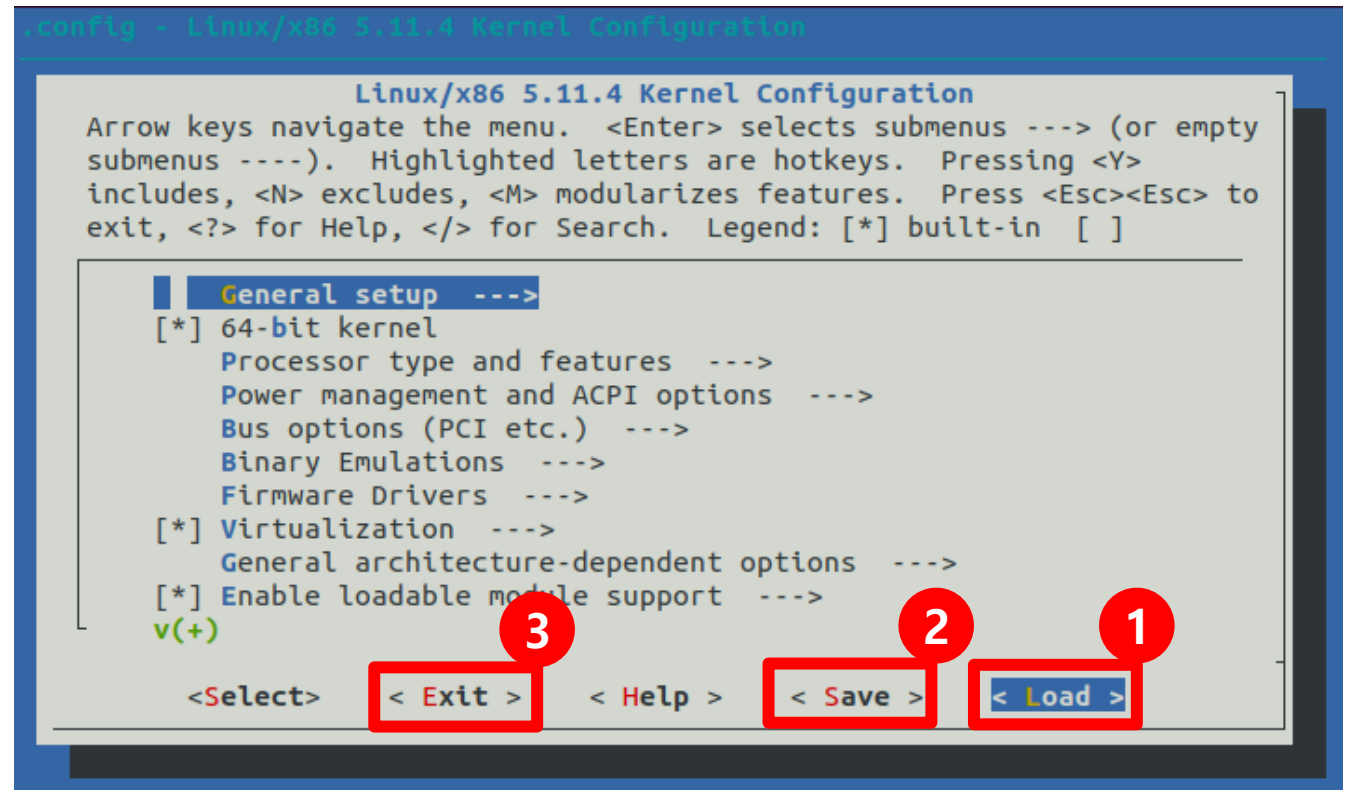
→ 현재 커널의 설정을 읽기

2) SAVE -> [.config] OK :

→ 현 상태의 커널의 설정을 저장

3) Exit

→ 터미널로 돌아가기



# Kernel 컴파일(4)

## 6) 커널 컴파일하기

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

## 7) 커널 모듈 설치하기

- `$ sudo make modules_install`

## 8) 커널 설치하기

- `$ sudo make install`

## 9) 설치완료 후 재부팅하기

- `$ sudo reboot`

# 모듈(Module) 프로그래밍

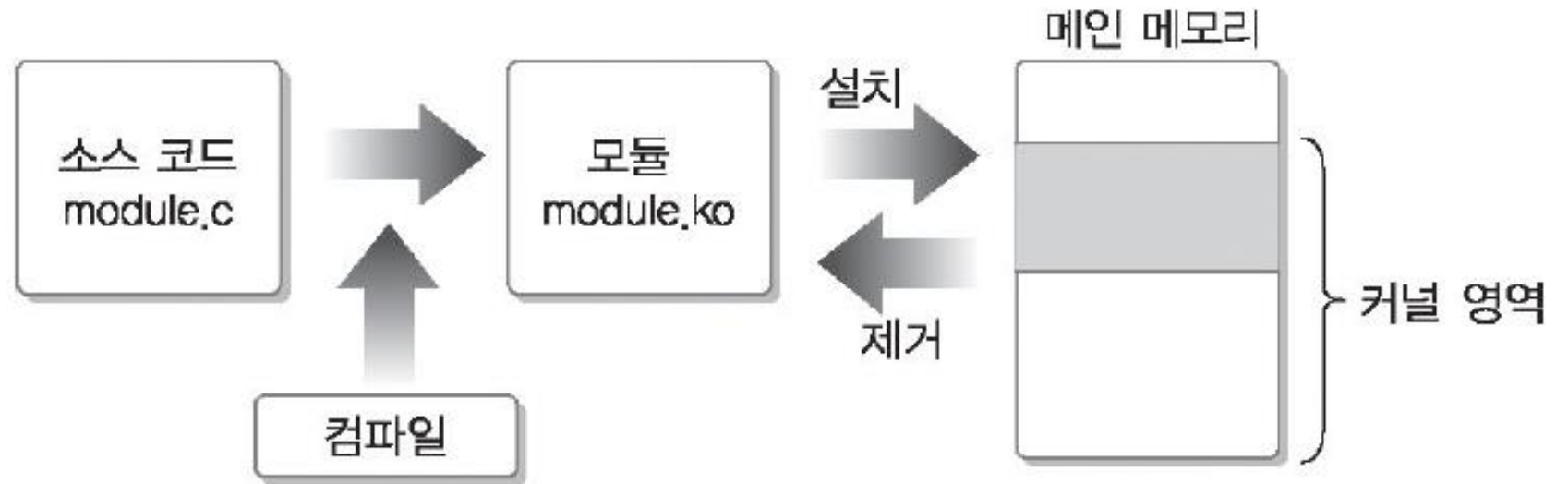
## 1) 모듈 프로그래밍

- 커널을 변경할 시 커널 전체를 다시 컴파일 해야함 ex) system call 등록
- 모듈 프로그램으로 개발하면, **해당 모듈만 컴파일** 후 **필요할 때만 동적으로 연결**하여 커널의 일부로 사용 가능
- 자주 사용하지 않는 커널 기능을 메모리에 상주시키지 않아도 됨
- **확장성과 재사용성**을 높일 수 있음
- **사건 구동형 (event-driven program)** 방식으로 작성
- 내부에 main() 함수가 존재하지 않음
- 커널에 적재/제거하기 위한 규칙과 유틸리티가 필요
- 커널에 적재된 모듈 프로그램은 시스템 내부에서 **모든 특권**을 가지게 되므로, 신중하게 작성해야 함

# 모듈(Module) 프로그래밍

## 2) 모듈 프로그래밍

- 모듈 프로그램 작성
- 모듈 프로그램 컴파일
- 모듈 설치
- 설치된 모듈 확인
- 모듈 제거



# 모듈(Module) 프로그래밍

## 3) 모듈 프로그래밍 코드 작성

```
GNU nano 2.9.3 mymodule.c

#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

int simple_init(void)                                초기화 루틴
{
    printk(KERN_INFO "Loading My Module.....\n");
    return 0;
}

void simple_exit(void)                                종료 루틴
{
    printk(KERN_INFO "Removing My Module.....\n");
}

module_init(simple_init);                             초기화, 종료 루틴 설정
module_exit(simple_exit);
```

# 모듈(Module) 프로그래밍

## 4) 모듈 Makefile 코드 작성

```
GNU nano 2.9.3 Makefile

KDIR= /lib/modules/$(shell uname -r)/build
PWD= $(shell pwd)

obj-m := mymodule.o

default:
    $(MAKE) -C $(KDIR) KBUILD_EXTMOD=$(PWD) modules

clean:
    rm -rf *~
```

# 모듈(Module) 프로그래밍

## 5) 모듈 프로그램 컴파일

- \$ sudo make

```
dongmin1@dongmin1-VirtualBox:~/week4$ sudo make
make -C /lib/modules/5.5.13/build KBUILD_EXTMOD=/home/dongmin1/week4 modules
make[1]: Entering directory '/usr/src/linux-5.5.13'
  CC [M]  /home/dongmin1/week4/mymodule.o
  Building modules, stage 2.
  MODPOST 1 modules
WARNING: modpost: missing MODULE_LICENSE() in /home/dongmin1/week4/mymodule.o
see include/linux/module.h for more information
  CC [M]  /home/dongmin1/week4/mymodule.mod.o
  LD [M]  /home/dongmin1/week4/mymodule.ko
make[1]: Leaving directory '/usr/src/linux-5.5.13'
```

```
dongmin1@dongmin1-VirtualBox:~/week4$ ls
Makefile      Module.symvers  mymodule.ko     mymodule.mod.c  mymodule.o
modules.order mymodule.c      mymodule.mod    mymodule.mod.o
```



# 모듈(Module) 프로그래밍

## 6) 모듈 설치

- \$ sudo insmod {*modulename*}.ko
- \$ dmesg

```
[ 89.714377] 04:41:36.586061 main      OS Product: Linux
[ 89.714472] 04:41:36.586164 main      OS Release: 5.5.13
[ 89.714573] 04:41:36.586258 main      OS Version: #3 SMP Tue Apr 7 09:50:56 KS
T 2020
[ 89.714721] 04:41:36.586359 main      Executable: /opt/VBoxGuestAdditions-6.1.
4/sbin/VBoxService
          04:41:36.586363 main      Process ID: 959
          04:41:36.586366 main      Package type: LINUX_64BITS_GENERIC
[ 89.718947] 04:41:36.590600 main      6.1.4 r136177 started. Verbose level = 0
[ 89.723660] 04:41:36.595298 main      vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[ 120.096227] systemd-journald[275]: File /var/log/journal/fbf0c302b00f4680905c
0220a8f589b6/user-1000.journal corrupted or uncleanly shut down, renaming and re
placing.
[ 122.298655] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 128.366872] rfkill: input handler disabled
[ 129.549405] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 136.538498] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 136.564392] ISO 9660 Extensions: RRIP_1991A
[ 155.205207] SYSCALL HELLO IS CALLED!
[19468.727579] mymodule: module license 'unspecified' taints kernel.
[19468.727581] Disabling lock debugging due to kernel taint
[19468.727900] Loading My Module.....
dongmin1@dongmin1-VirtualBox:~/week4$
```

# 모듈(Module) 프로그래밍

## 7) 설치된 모듈 확인

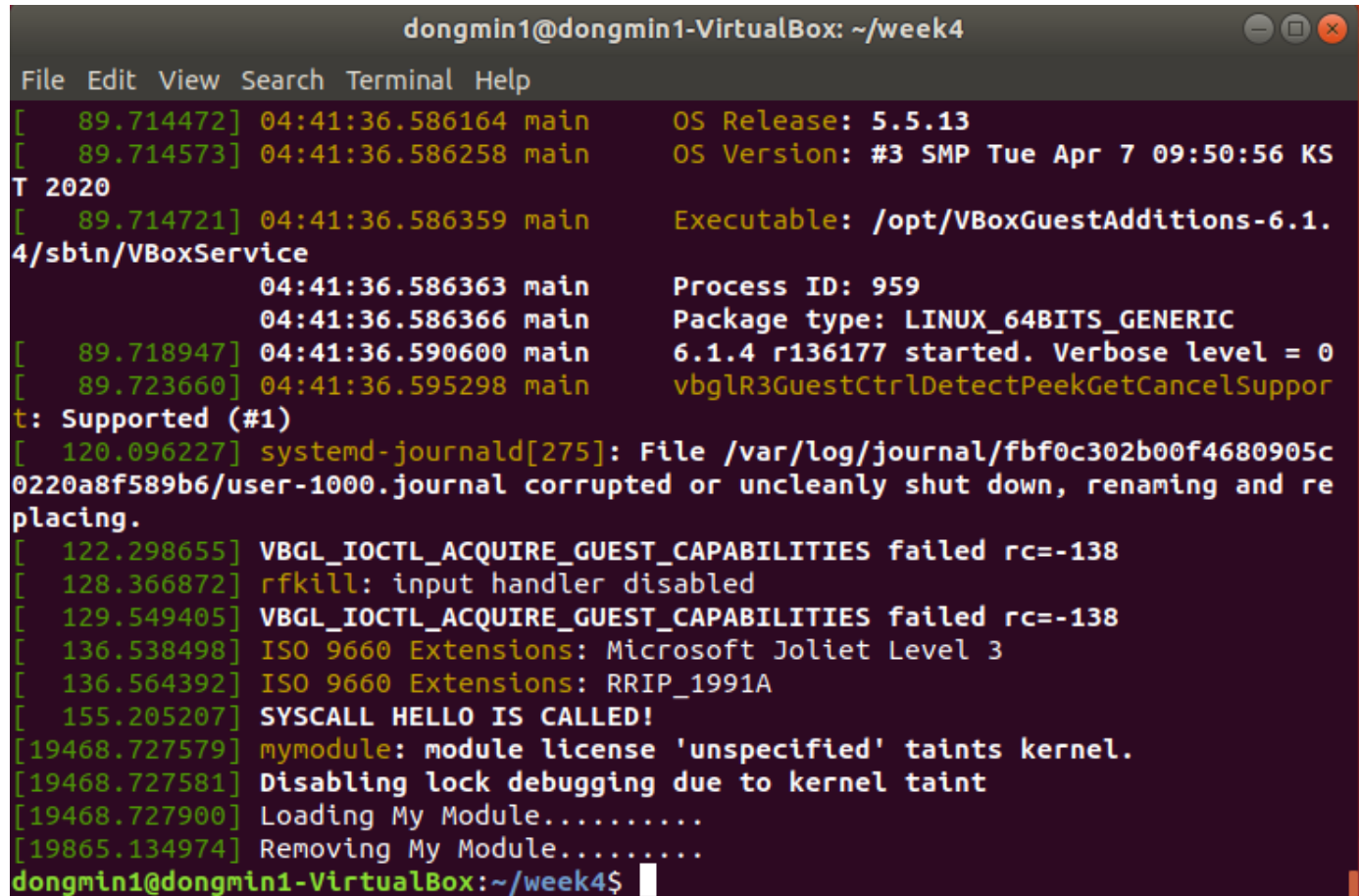
- \$ lsmod

```
dongmin1@dongmin1-VirtualBox:~/week4$ sudo insmod mymodule.ko
dongmin1@dongmin1-VirtualBox:~/week4$ lsmod
Module                Size  Used by
mymodule              16384  0
nls_utf8              16384  1
isofs                 49152  1
vboxvideo             36864  0
intel_rapl_msr        20480  0
intel_rapl_common     24576  1 intel_rapl_msr
intel_powerclamp      20480  0
crct10dif_pclmul      16384  1
crc32_pclmul          16384  0
vmwgfx                303104  4
```

# 모듈(Module) 프로그래밍

## 8) 모듈 제거

- `$ sudo rmmod {modulename}`
- `$ dmesg`



```
dongmin1@dongmin1-VirtualBox: ~/week4
File Edit View Search Terminal Help
[ 89.714472] 04:41:36.586164 main OS Release: 5.5.13
[ 89.714573] 04:41:36.586258 main OS Version: #3 SMP Tue Apr 7 09:50:56 KS
T 2020
[ 89.714721] 04:41:36.586359 main Executable: /opt/VBoxGuestAdditions-6.1.
4/sbin/VBoxService
04:41:36.586363 main Process ID: 959
04:41:36.586366 main Package type: LINUX_64BITS_GENERIC
[ 89.718947] 04:41:36.590600 main 6.1.4 r136177 started. Verbose level = 0
[ 89.723660] 04:41:36.595298 main vbglR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[ 120.096227] systemd-journald[275]: File /var/log/journal/fbf0c302b00f4680905c
0220a8f589b6/user-1000.journal corrupted or uncleanly shut down, renaming and re
placing.
[ 122.298655] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 128.366872] rfkill: input handler disabled
[ 129.549405] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[ 136.538498] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 136.564392] ISO 9660 Extensions: RRIP_1991A
[ 155.205207] SYSCALL HELLO IS CALLED!
[19468.727579] mymodule: module license 'unspecified' taints kernel.
[19468.727581] Disabling lock debugging due to kernel taint
[19468.727900] Loading My Module.....
[19865.134974] Removing My Module.....
dongmin1@dongmin1-VirtualBox:~/week4$
```