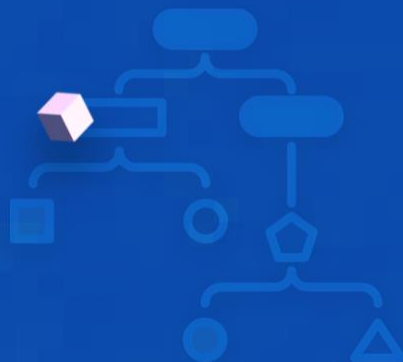


강원혁신플랫폼

리눅스프로그래밍

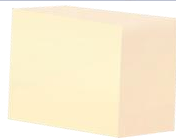
리눅스 커널과 커널 생성





리눅스 운영 체제의 핵심 부분으로 시스템의 하드웨어와 관련된 작업, 프로세스 관리, 메모리 관리, 파일 시스템, 디바이스 드라이버 등 운영 체제의 핵심 기능을 담당하는 영역을 무엇이라고 하나요?

커널 영역 (Kernel Space)





리눅스 커널 빌드 순서에서, 메뉴를 사용하여
커널 구성을 설정할 수 있는 단계는 무엇인가요?

menuconfig





학습 내용

- 1 리눅스 커널과 모듈
- 2 리눅스 디바이스 드라이버
- 3 커널 생성

학습 목표

- 리눅스 커널과 모듈에 대해 설명할 수 있다.
- 리눅스 디바이스 드라이버에 대해 설명할 수 있다.
- 커널 생성에 대해 설명할 수 있다.

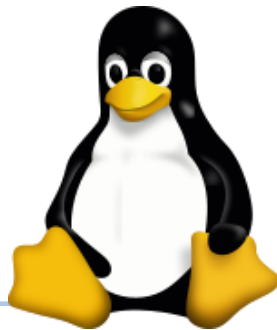
강원혁신플랫폼

리눅스프로그래밍



리눅스 커널과 모듈





Linux kernel 리눅스 커널

운영 체제의 핵심 부분

시스템의 하드웨어와 소프트웨어 리소스를 관리하고 프로세스 관리,
메모리 관리, 파일 시스템 등 다양한 기능을 수행
주로 C 언어로 작성되며, 운영 체제의 모든 기능과 서비스에 대한
인터페이스를 제공
리눅스 커널은 모듈이라는 개념을 지원



모듈

[커널의 일부 기능을 동적으로 추가하거나
제거할 수 있는 작은 소프트웨어 단위]



모듈

특징

- ◆ 필요한 경우에만 로드되어 실행되며, 필요하지 않은 경우에는 언로드 가능
- ◆ 커널을 재 컴파일하지 않고도 커널에 새로운 기능을 추가하거나 기존 기능을 제거



모듈



특징

디바이스
드라이버

파일 시스템

네트워킹
프로토콜

특정 기능을 구현하는데 사용



모듈

특징

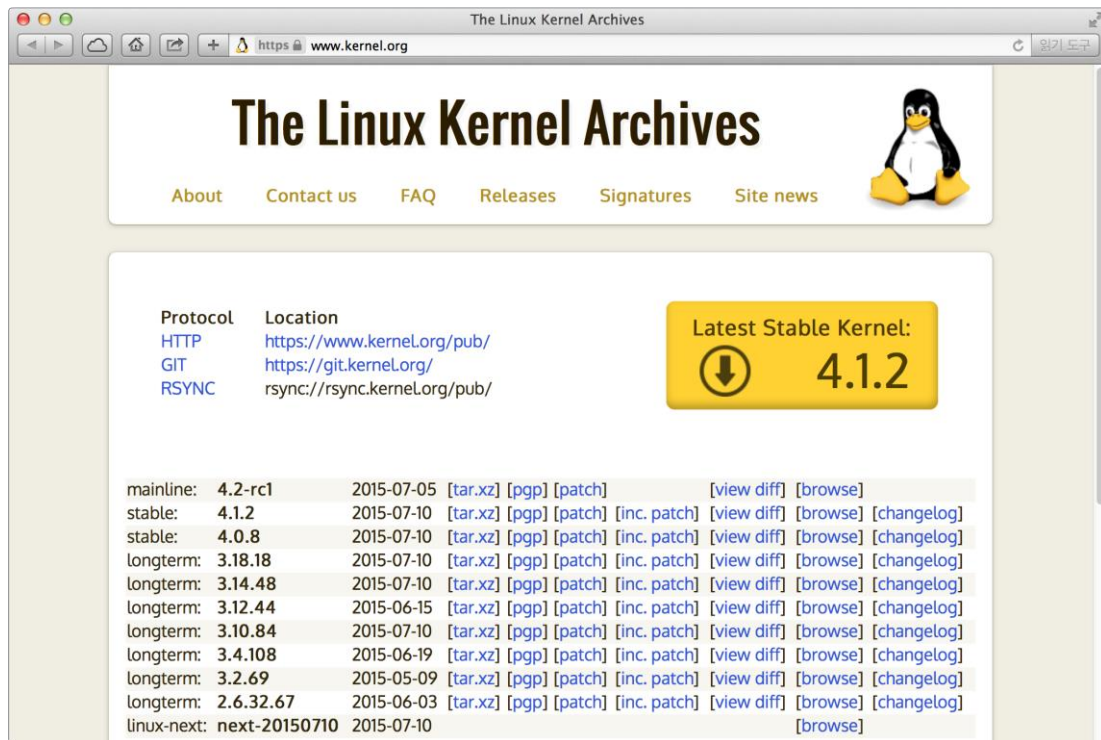
- ◆ 모듈은 일반적으로 커널의 동적 로딩 기능을 사용





리눅스 커널 아카이브

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



The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news


Protocol Location

HTTP <https://www.kernel.org/pub/>

GIT <https://git.kernel.org/>

RSYNC <rsync://rsync.kernel.org/pub/>

Latest Stable Kernel:

 4.1.2

mainline:	4.2-rc1	2015-07-05	[tar.xz]	[pgp]	[patch]	[view diff]	[browse]
stable:	4.1.2	2015-07-10	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
stable:	4.0.8	2015-07-10	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.18.18	2015-07-10	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.14.48	2015-07-10	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.12.44	2015-06-15	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.10.84	2015-07-10	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.4.108	2015-06-19	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.2.69	2015-05-09	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	2.6.32.67	2015-06-03	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
linux-next:	next-20150710	2015-07-10					[browse]

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020



모노리식 커널과 마하 커널



모노리식 커널(Monolithic Kernel)



- ◆ 리눅스는 모노리식 커널 아키텍처
- ◆ 리눅스 커널이 운영 체제의 모든 기능과 서비스를 단일 커널 공간에서 제공
- ◆ 리눅스 모노리식 커널은 프로세스 관리, 메모리 관리, 파일 시스템, 디바이스 드라이버 등 모든 운영 체제 기능을 하나의 단일 실행 파일로 구현
- ◆ 커널의 크기가 크고, 모든 기능이 커널 내부에서 직접 실행



모노리식 커널과 마하 커널

01

02

마하(Mach) 커널



- ◆ 마하 커널은 마이크로 커널 아키텍처(macOS)
- ◆ 마하는 커널의 핵심 기능을 최소화하고, 추가적인 서비스는 유저 모드의 서버나 프로세스로 분리하여 제공
- ◆ 마하 커널은 프로세스 관리, 메모리 관리, IPC(Inter-Process Communication) 등의 기본적인 기능만 제공
- ◆ 파일 시스템이나 디바이스 드라이버와 같은 서비스는 유저 모드의 서버에서 실행



모노리식 커널과 마하 커널

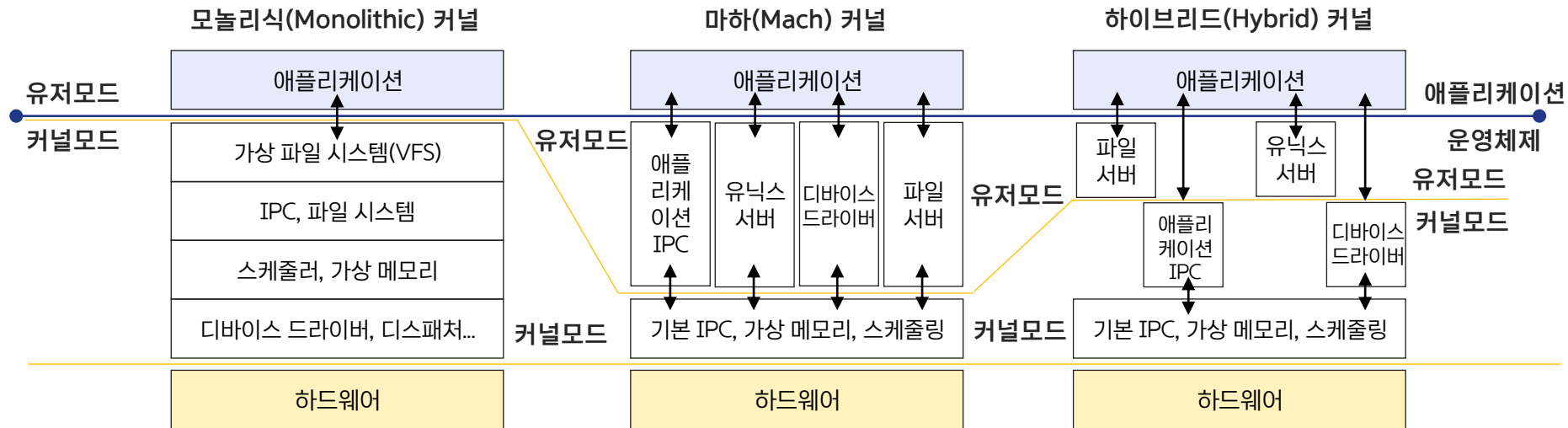
01

02

마하(Mach) 커널



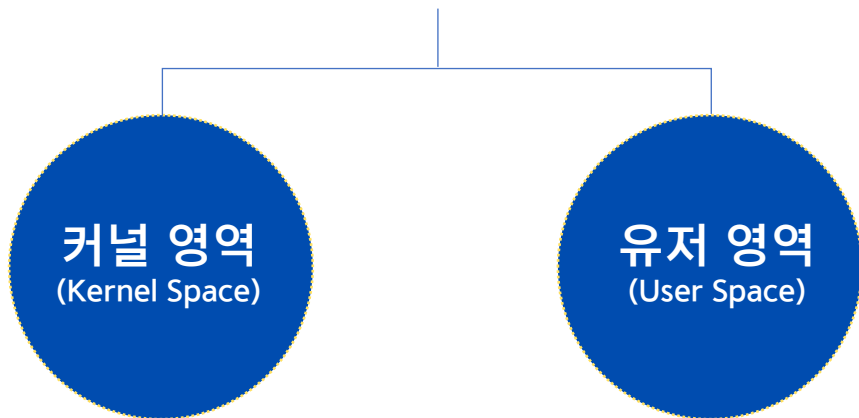
- ◆ 커널 자체의 크기를 줄이고, 기능을 서버로 분리
- ◆ 유연성과 확장성을 강조





커널 영역과 유저 영역

리눅스 운영 체제





커널 영역과 유저 영역

📦 커널 영역(Kernel Space)

- ◆ 운영 체제의 핵심 부분
- ◆ 리눅스 커널의 코드와 데이터가 위치하는 영역
- ◆ 시스템의 하드웨어와 관련된 작업, 프로세스 관리, 메모리 관리, 파일 시스템, 디바이스 드라이버 등 운영 체제의 핵심 기능을 담당
- ◆ 특권 모드(Privileged Mode)에서 실행
- ◆ 커널 자체의 메모리 공간을 사용



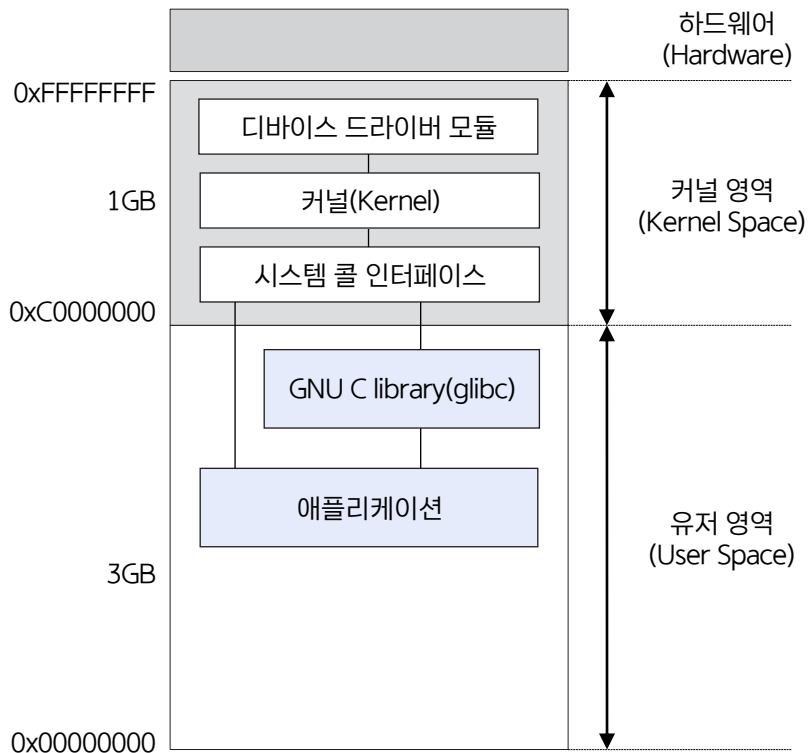
커널 영역과 유저 영역

유저 영역(User Space)

- ◆ 일반 응용 프로그램이 실행되는 영역
- ◆ 응용 프로그램의 코드와 데이터가 위치
- ◆ 유저 영역에서는 직접 커널 영역에 접근할 수 없으며, 시스템 호출(System Call)이나 인터럽트를 통해 커널에 요청을 보내고 응답 받음
- ◆ 운영 체제의 안정성과 보안을 유지하는 데 중요한 역할



커널 영역과 유저 영역



강원혁신플랫폼

리눅스프로그래밍



리눅스 디바이스 드라이버





Linux Device Driver 리눅스 디바이스 드라이버

리눅스 운영 체제에서 하드웨어 장치를 제어하고
관리하기 위한 소프트웨어 모듈
디바이스 드라이버는 특정한 하드웨어 장치와 상호 작용하여
사용자 애플리케이션과 운영 체제 간의 인터페이스 역할 담당



리눅스 디바이스 드라이버





리눅스 디바이스 드라이버

개발 단계



[해당 하드웨어 장치의 동작 원리와 특성에 대한 이해]



리눅스 디바이스 드라이버

개발 단계

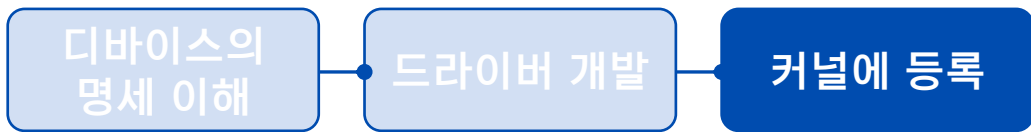


[C 또는 C++과 같은 언어를 사용하여
디바이스 드라이버를 개발]



리눅스 디바이스 드라이버

개발 단계



[개발한 디바이스 드라이버(모듈 파일(.ko))를
커널에 등록]



리눅스 디바이스 드라이버

관련 용어



디바이스(Device)



- ◆ 컴퓨터 시스템에서 [하드웨어](#)
- ◆ 입력장치(키보드, 마우스), 출력장치(모니터, 프린터), 저장장치(하드 디스크, USB 드라이브), 네트워크 인터페이스 카드 등



리눅스 디바이스 드라이버

관련 용어

01

02

03

04

디바이스 드라이버(Device Driver)



- ◆ 디바이스 드라이버는 컴퓨터 시스템에서 특정한 디바이스를 제어하고 관리하기 위한 [소프트웨어 모듈](#)



리눅스 디바이스 드라이버

관련 용어

01

02

03

04

디바이스 파일(Device File)



- ◆ 디바이스 파일은 리눅스에서 디바이스에 접근하기 위한 인터페이스로, 파일 시스템의 일부로 간주
- ◆ '/dev/sda'는 첫 번째 하드 디스크를 나타냄



리눅스 디바이스 드라이버

관련 용어

01

02

03

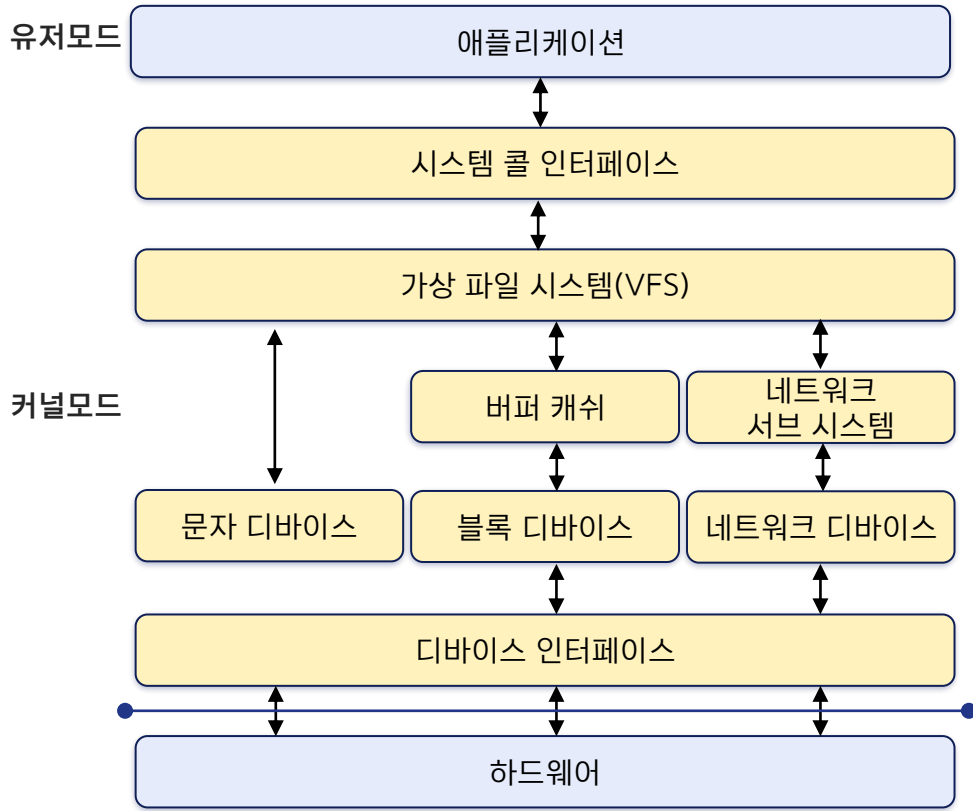
04

디바이스 테스트 프로그램(Device Testing Program)

- ◆ 디바이스의 동작을 확인하고 테스트하기 위한 프로그램



리눅스 커널과 디바이스 드라이버



(major#, minor#)

```
$ cat /proc/devices
```

Character devices:

1 mem

4 /dev/vc/0

4 tty

4 ttyS

5 /dev/tty

5 /dev/console

5 /dev/ptmx

7 vcs

10 misc

13 input

...중간생략

166 ttyACM

180 usb

188 ttyUSB

189 usb_device

...중간생략

254 ndctl



Block devices:

1 ramdisk

7 loop

8 sd

9 md

65 sd

66 sd

67 sd

68 sd

69 sd

70 sd

71 sd



...중간생략

252 device-mapper

253 virtblk

254 mdp

259 blkext

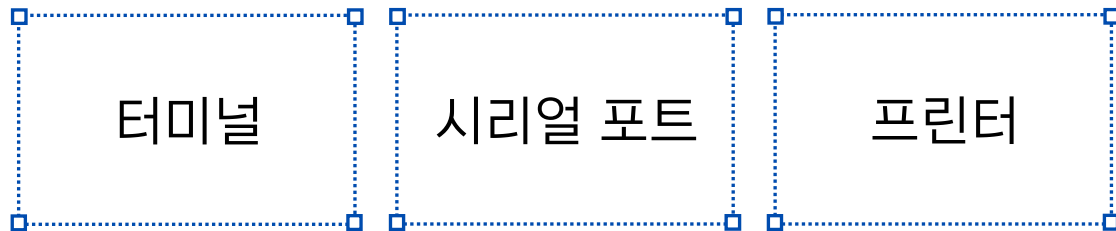
\$



문자 디바이스 파일

문자 디바이스 드라이버(Character Device Driver)

- ◆ 리눅스 운영 체제에서 문자 기반 디바이스를 제어하고 관리하기 위한 드라이버
- ◆ 데이터를 문자 단위로 전송하는 디바이스





```
$ ls -l /dev/ | grep ^c
crw-r--r-- 1 root root 10, 235 Jul 13 15:15 autofs
crw--w---- 1 root tty 5, 1 Jul 13 15:15 console
crw-r--r-- 1 root root 1, 11 Jul 13 15:15 kmsg
crw-rw---- 1 root kvm 10, 232 Jul 13 15:15 kvm
crw-rw---- 1 root disk 10, 237 Jul 13 15:15 loop-control
crw-rw-rw- 1 root root 1, 3 Jul 13 15:15 null
crw-r----- 1 root kmem 10, 144 Jul 13 15:15 nvram
crw----- 1 root root 249, 0 Jul 13 15:15 ptp0
crw-rw-rw- 1 root tty 5, 0 Jul 13 15:15 tty
crw--w---- 1 root tty 4, 0 Jul 13 15:15 tty0
crw--w---- 1 root tty 4, 1 Jul 13 15:15 tty1
```



```
crw-rw---- 1 root dialout  4, 64 Jul 13 15:15 ttyS0
crw-rw---- 1 root dialout  4, 65 Jul 13 15:15 ttyS1
crw-rw---- 1 root dialout  4, 66 Jul 13 15:15 ttyS2
crw-rw-rw- 1 root root    10, 126 Jul 13 15:15 vsock
```





블록 디바이스 파일

블록 디바이스 드라이버(Block Device Driver)

- ◆ 리눅스 운영 체제에서 블록 단위로 데이터를 전송하는 디바이스
- ◆ 저장 장치





```
$ ls -l /dev/ | grep ^b
```

```
brw-rw---- 1 root disk 7, 0 Jul 13 15:15 loop0
```

```
brw-rw---- 1 root disk 7, 1 Jul 13 15:15 loop1
```

```
brw-rw---- 1 root disk 1, 0 Jul 13 15:15 ram0
```

```
brw-rw---- 1 root disk 1, 1 Jul 13 15:15 ram1
```

```
brw-rw---- 1 root disk 1, 2 Jul 13 15:15 ram2
```

```
brw-rw---- 1 root disk 1, 3 Jul 13 15:15 ram3
```

```
brw-rw---- 1 root disk 1, 4 Jul 13 15:15 ram4
```

```
brw-rw---- 1 root disk 1, 5 Jul 13 15:15 ram5
```

```
brw-rw---- 1 root disk 8, 0 Jul 13 15:15 sda
```

```
brw-rw---- 1 root disk 8, 16 Jul 13 15:15 sdb
```

```
brw-rw---- 1 root disk 8, 32 Jul 13 15:15 sdc
```



네트워크 디바이스

- 네트워크 디바이스 드라이버(Network Device Driver)
 - 네트워크 인터페이스 카드(Network Interface Card, NIC)와 같은 네트워크 기기를 제어하고 관리하기 위한 드라이버



```
$ ifconfig
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
inet 172.18.93.102 netmask 255.255.240.0 broadcast 172.18.95.255
```

```
inet6 fe80::215:5dff:fe45:d9ec prefixlen 64 scopeid 0x20<link>
```

```
ether 00:15:5d:45:d9:ec txqueuelen 1000 (Ethernet)
```

```
RX packets 2611 bytes 540675 (540.6 KB)
```

```
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 214 bytes 66476 (66.4 KB)
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

```
inet 127.0.0.1 netmask 255.0.0.0
```

```
inet6 ::1 prefixlen 128 scopeid 0x10<host>
```

```
loop txqueuelen 1000 (Local Loopback)
```

```
RX packets 0 bytes 0 (0.0 B)
```

```
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 0 bytes 0 (0.0 B)
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

강원혁신플랫폼

리눅스프로그래밍



커널 생성





```
$ uname -a
```

```
Linux ATwin 5.15.90.1-microsoft-standard-WSL2 #1 SMP Fri Jan 27 02:56:13 UTC  
2023 x86_64 x86_64 x86_64 GNU/Linux
```





GitHub이용을 위한 git설치

```
$ sudo apt-get install git
```

```
[sudo] password for freetime:
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
git is already the newest version (1:2.34.1-1ubuntu1.9).
```

```
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
```

```
$
```



❏ 라즈베리파이 커널 소스코드 다운로드

```
$ wget https://github.com/raspberrypi/linux/archive/rpi-5.3.y.tar.gz
```

❏ 라즈베리파이 커널 소스코드 압축풀기

```
$ tar zxvf rpi-5.3.y.tar.gz
```

❏ 라즈베리파이 커널 소스코드 이동

```
$ sudo mv linux-rpi-5.3.y/ /usr/src
```

❏ 라즈베리파이 커널 소스코드 확인

```
$ cd /usr/src
```

```
$ ls
```

```
linux-rpi-5.3.y
```



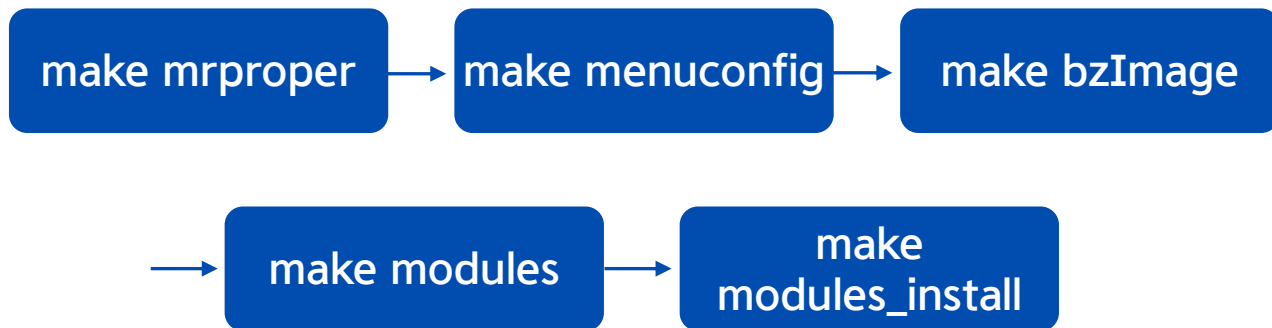
- 라즈베리파이 커널 소스코드 이름을 linux로 변경

```
$ sudo ln -s linux-rpi-5.3.y/ linux
```

- 라즈베리파이 커널 소스코드 수정, 추가, 커널 재컴파일(빌드)



리눅스 커널 빌드 순서





리눅스 커널 빌드 순서

옵션	내용	비고
mrproper	커널의 설정을 초기화함	
menuconfig	커널을 설정할 수 있는 메뉴를 띄움	xconfig, gconfig
dep	커널을 컴파일하기 전에 의존관계(dependency)를 검사함	
bzImage	압축된 커널 이미지를 생성한함 (빌드된 이미지는 /usr/src/linux/arch/i386/boot에 위치)	zImage
modules	커널의 모듈들을 빌드함	
modules_install	빌드된 커널 모듈을 설치함(/lib/modules에 설치)	
depmod	모듈 간의 의존관계를 검사해서 /lib/modules/modules.dep 파일을 생성함	
clean	컴파일 과정에서 생성되었던 중간 파일들을 삭제함	



라즈베리 파이의 리눅스 커널의 디렉터리 구조(버전 3.18.y)

raspberrypi/linux

483,138 commits 19 branches 0 releases 4,500 contributors

branch: rpi-3.18.y linux / +

Merge pull request #838 from jburgess777/rpi-3.18.y-i2c-rtc-overlay

pehwell authored an hour ago latest commit 2594f991ce

Documentation	Merge remote-tracking branch 'stable/linux-3.18.y' into rpi-3.18.y	20 days ago
arch	Merge pull request #838 from jburgess777/rpi-3.18.y-i2c-rtc-overlay	an hour ago
block	genhd: check for int overflow in disk_expand_part_tbl()	a month ago
crypto	crypto: add missing crypto module aliases	18 days ago
drivers	lirc-rpi: use getnstimeofday instead of read_current_timer	a day ago
firmware	kbuild: remove obj-n and lib-n handling	5 months ago
fs	nillfs2: fix deadlock of segment constructor over I_SYNC flag	7 days ago
include	Merge remote-tracking branch 'stable/linux-3.18.y' into rpi-3.18.y	6 days ago
init	param: fix crash on bad kernel arguments	3 months ago
ipc	ipc/sem.c: fully initialize sem_array before making it visible	3 months ago

Issues 144

Pull Requests 17

Wiki

Pulse

Graphs

HTTPS clone URL

<https://github.com/r>

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP



리눅스 커널의 주요 디렉터리

디렉터리	내용
arch	특정 하드웨어/아키텍처에 종속되어 있는 소스 코드들이 위치
arch/arm/kernel	ARM CPU와 관련된 소스 코드들이 위치
drivers	디바이스 드라이버들과 관련된 소스 코드들이 위치
fs	파일 시스템과 관련된 소스 코드들이 위치
init	시스템의 초기화와 관련된 소스 코드들이 위치
include	커널과 관련된 헤더 파일들이 위치
ipc	프로세스 간의 통신을 위한 IPC 관련 소스 코드들이 위치
kernel	태스크나 시그널과 같은 프로세스와 관련된 소스 코드들이 위치



리눅스 커널의 주요 디렉터리

디렉터리	내용
mm	메모리 관리 관련된 소스 코드들이 위치
net	네트워킹과 관련된 소스 코드들이 위치
script	커널 빌드와 관련된 스크립트 파일들이 위치
sound	사운드와 관련된 소스 코드들이 위치



01 • 리눅스 커널과 모듈

02 • 리눅스 디바이스 드라이버

03 • 커널 생성