

Part 04. 리눅스 커널 입문

# Chapter 01.

# 리눅스 커널 구조 및 컴파일

## 진행 순서

Chapter 01\_01

리눅스 커널 구조

Chapter 01\_02

리눅스 커널 컴파일

Chapter 01\_01    리눅스 커널 구조

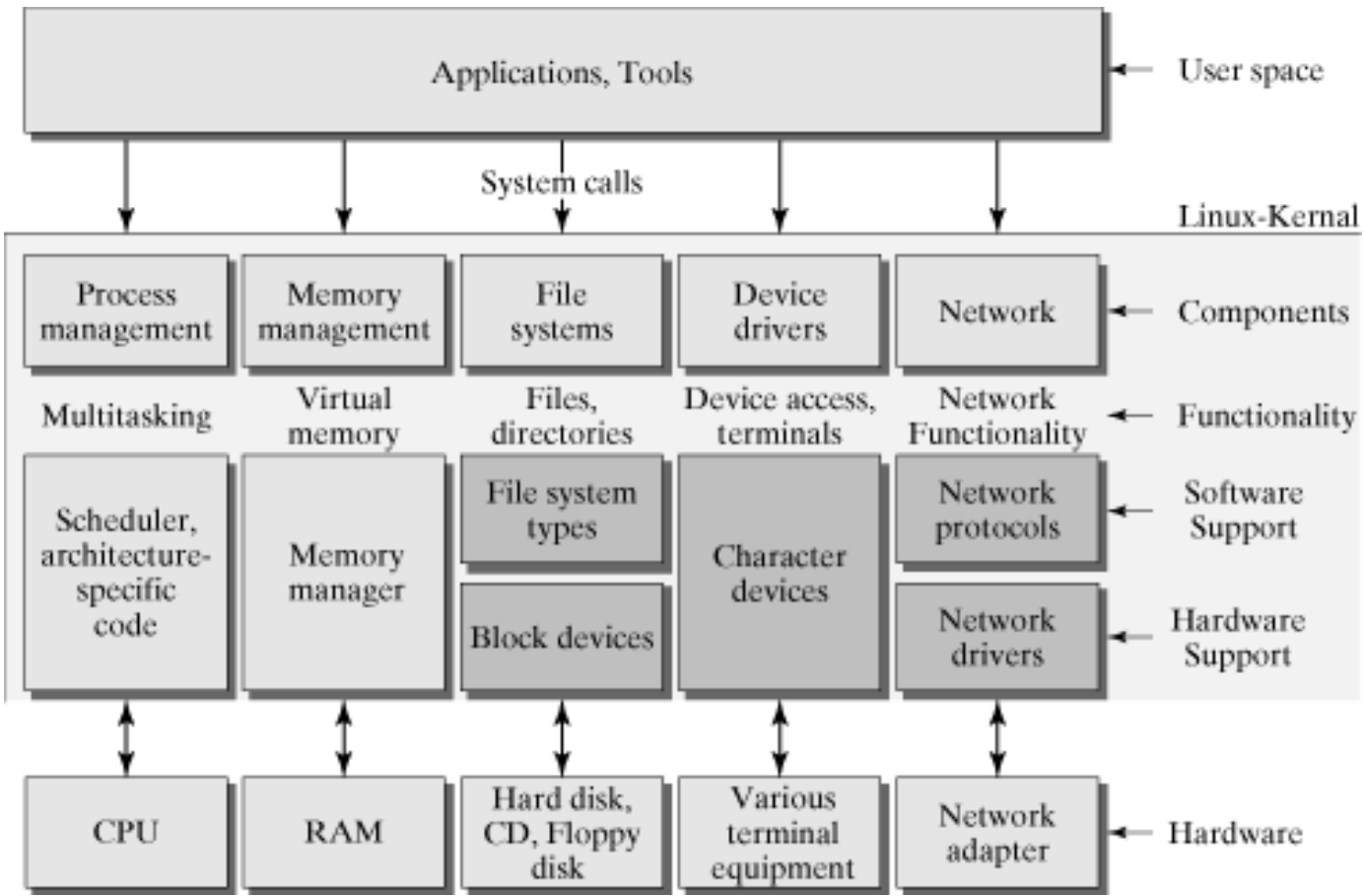


Figure 2-1. Structure of the Linux kernel according to [RuCo01].  
출처: <https://flylib.com/books/en/3.475.1.15/1/>

Chapter 01\_01     리눅스 커널 구조

# 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 Release

5.7.10

mainline:	5.8-rc6	2020-07-19	<a href="#">[tarball]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>
stable:	5.7.10	2020-07-22	<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="#">[change log]</a>
longterm:	5.4.53	2020-07-22	<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="#">[change log]</a>
longterm:	4.19.134	2020-07-22	<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="#">[change log]</a>
longterm:	4.14.189	2020-07-22	<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="#">[change log]</a>
longterm:	4.9.231	2020-07-22	<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="#">[change log]</a>
longterm:	4.4.231	2020-07-22	<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="#">[change log]</a>
linux-next:	next-20200724	2020-07-24					<a href="#">[browse]</a>

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



<https://elixir.bootlin.com/linux/latest/source>

HOME   ENGINEERING   TRAINING   DOCS   COMMUNITY   COMPANY

Stuck at home?  
All Bootlin training courses  
are now available  
through on-line seminars

bootlin

Elixir Cross Referencer

Linux

Filter tags

v5

v5.8

v5.7

v5.7.8

v5.7.7

v5.7.6

v5.7.5

v5.7.4

v5.7.3

v5.7.2

v5.7.1

v5.7

v5.7-rc7

v5.7-rc6

v5.7-rc5

v5.7-rc4

v5.7-rc3

v5.7-rc2

Documentation

LICENSES

arch

block

certs

crypto

drivers

fs

include

init

ipc

kernel

lib

mm

net

linux   v5.7.8

powered by Elixir 2.1

Chapter 01\_01      리눅스 커널 구조

Linux

Filter tags

v5

v5.8

v5.7

v5.7.8

v5.7.7

v5.7.6

v5.7.5

v5.7.4

v5.7.3

v5.7.2

v5.7.1

v5.7

v5.7-rc7

v5.7-rc6

v5.7-rc5

v5.7-rc4

v5.7-rc3

v5.7-rc2

v5.7-rc1

v5.6

v5.5

v5.4

v5.3

v5.2

v5.1

v5.0

v4

v3

v2

v1

v0

Documentation

LICENSES

arch

block

certs

crypto

drivers

fs

include

init

ipc

kernel

lib

mm

net

samples

scripts

security

sound

tools

usr

virt

COPYING496 bytes

CREDITS99724 bytes

Kbuild1327 bytes

Kconfig595 bytes

MAINTAINERS552385 bytes

Makefile61731 bytes

README727 bytes

linux v5.7.8

powered by Elixir 2.1

## Chapter 01\_01 리눅스 커널 구조

- Documentation
  - 커널 소스 문서화
- LICENSES
  - 커널 소스에 적용 할 라이선스와 라이선스가 다른 개별 소스 파일을 포함합니다. 이 디렉토리에는 기본 (GNU GPL), COPYING 파일에 나열된 예외 및 기타의 3 가지 하위 디렉토리가 있습니다.
- arch
  - 특정 아키텍처의 소스 코드 (예 : powerpc, x86 등). 이 디렉토리에는 i386, sparc, arm 등 지원되는 각 아키텍처에 속하는 서브 디렉토리가 있습니다.
- block
  - 블록 I/O 계층은 블록 장치 (하드 디스크, DVD, 플로피 디스크 등) 및 해당 요청을 관리하기 위한 코드를 포함합니다.
- certs
  - 커널이 서명 된 모듈을 로드 할 수 있도록 모듈 서명을 활성화하는 인증서 및 서명 파일
- crypto
  - 암호화 API. 암호화 및 압축 작업을 처리하는 암호화 암호를 포함합니다.

## Chapter 01\_01 리눅스 커널 구조

- drivers
  - 하드웨어 장치 드라이버. 하드웨어를 지원하기 위한 장치 드라이버 코드가 들어 있습니다. 이 디렉토리에는 비디오, 블루투스 등과 같은 각 하드웨어 (하위 하드웨어 지원)와 같은 하위 디렉토리가 있습니다.
- fs
  - 가상 파일 시스템 및 추가 파일 시스템에 대한 코드. 이 디렉토리에는 파일 시스템을 지원하고 읽고 쓰는 코드가 들어 있습니다.
- include
  - 커널 헤더. 이 디렉토리에는 코드를 컴파일하는 함수와 같은 커널 파일의 c 헤더가 있습니다.
- init
  - 커널 부팅. 커널 초기화와 관련된 소스 코드를 포함합니다. 소스 코드는 /init 디렉토리 내의 mainc.c라는 파일에 저장됩니다. 코드는 커널과 일부 초기 프로세스를 초기화합니다.
- ipc
  - 신호 및 파이프와 같은 프로세스 간 통신
- kernel
  - 스케줄러 신호 처리 코드 등과 같은 핵심 서브 시스템

## Chapter 01\_01 리눅스 커널 구조

- lib
  - 라이브러리 루틴 공통 문자열 조작, 하드웨어 종속 조작, 디버깅 루틴 및 명령 행 구문 분석 코드.
- mm
  - 메모리 관리 및 가상 메모리. 커널은 하드웨어와 가상 메모리(스왑)를 모두 관리합니다. 이 디렉토리는 메모리 관리를 위한 코드를 저장합니다.
- net
  - 네트워크 스택. IP, TCP, UP 등과 같은 통신 프로토콜과 관련된 코드를 포함합니다.
- samples
  - 샘플 코드 및 구성 파일.
- scripts
  - 커널을 빌드하는 스크립트
- security
  - LSM (Linux Security Module)은 보안 정책이 모듈 제어에 액세스 할 수 있도록 하는 프레임 워크입니다.
- sound
  - 사운드 하위 시스템에는 ALSA와 같은 사운드와 관련된 사운드 드라이버 및 코드가 있습니다.



## Chapter 01\_01     리눅스 커널 구조

- tools
  - ACPI, cgroup, USB 테스트 도구, vhost 테스트 모듈, GPIO, IIO 및 spi 도구와 같은 압축 커널 개발 도구, 기타 에너지 정책 도구.
- usr
  - 파일 시스템을 루트화하고 커널 메모리 캐시에서 초기화하는 initramfs.
- virt
  - 가상화, 이 디렉토리에는 하이퍼바이저 용 KVM (Kernel Virtual Machine) 모듈이 포함되어 있습니다.

## Chapter 01\_02 리눅스 커널 컴파일

## 리눅스 커널 생성 과정 3단계

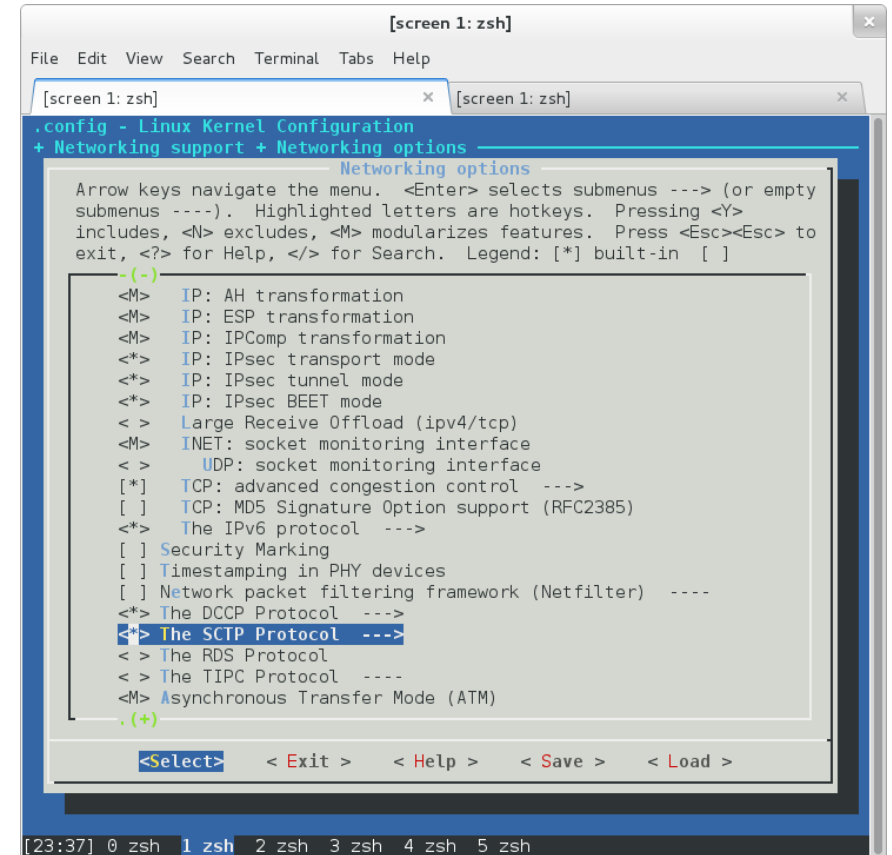
- 1) 커널 구성 (Kernel Configuration)
- 2) 커널 컴파일 (Kernel Compile)
- 3) 커널 인스톨 (Kernel Installation)

- 커널 구성

커널 구성은 새로 리눅스 커널 컴파일 시 사용할 환경 변수 등을 설정하는 과정이다. 하드웨어 특성, 커널 특성 등 설정 정보를 지정하는 과정이다.

make config, make menuconfig, make xconfig 등의 방법이 있다.

커널 구성 단계에서 사용자가 선택한 사항은 .config 이라는 파일에 저장되며 이후 커널 컴파일 단계에서 사용된다.



make menuconfig 예시

## Chapter 01\_02 리눅스 커널 컴파일

- 커널 컴파일

커널 소스 파일을 이용해 실행 가능한 커널 이미지를 만드는 과정  
make bzImage, make zImage 등의 명령으로 생성할 수 있다.

(2.6버전 이후 make 명령으로 생성 가능)

커널 컴파일이 완료되면 시스템 아키텍처에 따라

kernel/arch/x86/boot/ 디렉토리 이하에 커널 이미지가 생성된다.

- 커널 인스톨

생성된 커널 이미지로 시스템이 부팅될 수 있도록 만드는 과정  
생성된 이미지를 파일시스템으로 복사, 커널 모듈 인스톨,  
부트로더(grub) 수정 등의 과정으로 이루어진다.

make install

## Chapter 01\_02     리눅스 커널 컴파일

- CentOS 커널 소스

```
[root@localhost kernels]# pwd
/usr/src/kernels
[root@localhost kernels]# ls
3.10.0-1127.8.2.el7.x86_64 3.10.0-862.11.6.el7.x86_64
[parallels@localhost kernels]$ ls -hl
total 8.0K
drwxr-xr-x. 22 root root 4.0K May 30 22:40 3.10.0-1127.8.2.el7.x86_64
drwxr-xr-x. 22 root root 4.0K May 30 22:40 3.10.0-862.11.6.el7.x86_64
```

```
[root@localhost kernels]# yum install kernel-devel
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
* base: mirror.kakao.com
* extras: mirror.kakao.com
* updates: mirror.kakao.com
base | 3.6 kB 00:00:00
extras | 2.9 kB 00:00:00
updates | 2.9 kB 00:00:00
(1/2): extras/7/x86_64/primary_db | 205 kB 00:00:00
(2/2): updates/7/x86_64/primary_db | 3.0 MB 00:00:00
...
```

Chapter 01\_02     리눅스 커널 컴파일

```
=====
Package                Arch                Version                Repository                Size
=====
Installing:
kernel-devel            x86_64              3.10.0-1127.13.1.el7    updates                    18 M

Transaction Summary
=====
Install 1 Package

...

Complete!
[root@localhost kernels]#
[root@localhost kernels]# pwd
/usr/src/kernels
[root@localhost kernels]# ls
3.10.0-1127.13.1.el7.x86_64 3.10.0-1127.8.2.el7.x86_64 3.10.0-862.11.6.el7.x86_64
```

커널 드라이버 (모듈)를 컴파일해야 할 경우로 전체 커널 소스 트리를 실제로 설치하지는 않습니다.

## Chapter 01\_02 리눅스 커널 컴파일

- CentOS 커널 소스

```
[root@localhost ~]# rpm -i http://vault.centos.org/7.8.2003/updates/Source/SPackages/kernel-3.10.0-1127.13.1.el7.src.rpm
2>&1 | grep -v exist
[root@localhost ~]# ls
anaconda-ks.cfg Desktop Documents Downloads initial-setup-ks.cfg Music Pictures Public rpmbuild Templates Videos
[root@localhost ~]# cd rpmbuild/
[root@localhost rpmbuild]# ls
SOURCES SPECS
[root@localhost rpmbuild]# cd SOURCES/
[root@localhost SOURCES]# ls
centos-ca-secureboot.der  extra_certificates      kernel-3.10.0-x86_64-debug.config  Module.kabi_dup_x86_64
centos-kpatch.x509       find-debuginfo.sh        kernel-abi-whitelists-1127.tar.bz2  Module.kabi_ppc64
centos-ldup.x509         kernel-3.10.0-ppc64.config  kernel-kabi-dw-1127.tar.bz2      Module.kabi_ppc64le
centossecureboot001.crt  kernel-3.10.0-ppc64-debug.config  linux-3.10.0-1127.13.1.el7.tar.xz  Module.kabi_s390x
check-kabi               kernel-3.10.0-ppc64le.config  linux-kernel-test.patch          Module.kabi_x86_64
cpupower.config          kernel-3.10.0-ppc64le-debug.config  Makefile.common                  sign-modules
cpupower.service        kernel-3.10.0-s390x.config  modprobe-dccp-blacklist.conf    x509.genkey
debrand-rh-i686-cpu.patch  kernel-3.10.0-s390x-debug.config  Module.kabi_dup_ppc64
debrand-rh_taint.patch    kernel-3.10.0-s390x-kdump.config  Module.kabi_dup_ppc64le
debrand-single-cpu.patch  kernel-3.10.0-x86_64.config      Module.kabi_dup_s390x
```

## Chapter 01\_02     리눅스 커널 컴파일

- CentOS 커널 소스

```
[root@host]# yum install asciidoc audit-libs-devel bash binutils binutils-devel bison bzip2 diffutils elfutils-devel
[root@host]# yum install elfutils-libelf-devel findutils flex gawk gcc gnupg gzip hmaccalc m4 make module-init-tools
[root@host]# yum install net-tools newt-devel patch patchutils perl perl-ExtUtils-Embed python python-devel
[root@host]# yum install redhat-rpm-config rpm-build sh-utils tar xmlto zlib-devel
```

```
[user@host]$ cd ~/rpmbuild/SPECS
[user@host SPECS]$ rpmbuild -bp --target=$(uname -m) kernel.spec
```

```
[root@localhost rpmbuild]# cd BUILD
[root@localhost BUILD]# ls
kernel-3.10.0-1127.13.1.el7
[root@localhost BUILD]# cd kernel-3.10.0-1127.13.1.el7/
[root@localhost kernel-3.10.0-1127.13.1.el7]# cd linux-3.10.0-1127.13.1.el7.x86_64/
[root@localhost linux-3.10.0-1127.13.1.el7.x86_64]# ls
arch  COPYING  Documentation  fs  ipc  kernel  Makefile  net  samples  sound  virt
block  CREDITS  drivers  include  Kbuild  lib  Makefile  qlock  README  scripts  tools
configs  crypto  firmware  init  Kconfig  MAINTAINERS  mm  REPORTING-BUGS  security  usr
```

참조: [https://wiki.centos.org/HowTos/I\\_need\\_the\\_Kernel\\_Source](https://wiki.centos.org/HowTos/I_need_the_Kernel_Source)

- 수정한 커널 설치

참조: [https://wiki.centos.org/HowTos/Custom\\_Kernel](https://wiki.centos.org/HowTos/Custom_Kernel)