

최신 리눅스 커널 소스를 직접 실행하면서 분석할 수 있는

리눅스 커널 소스 실행 분석기

Quick Start Guide

커널연구회 (www.kernel.bz)

정재준 (rgbi3307@nate.com)

목차

Table of Contents

● 리눅스 커널 소스 실행 분석기.....	1
● 목차.....	2
● 커널 소스 실행 분석기 (Quick Start Guide).....	3
1.1 소스 경로 구조.....	5
1.2 qt 프로젝트.....	7
1.3 userspace RCU 패키지.....	11
1.4 소스 실행 및 디버깅.....	14
1.5 소스 공유(github).....	20

커널 소스 실행 분석기 (Quick Start Guide)

"리눅스 커널 소스 실행 분석기"는 커널연구회에서 제작한 어플리케이션입니다. 리눅스 커널 소스를 qt 어플리케이션 프로젝트로 구성하여 사용자 콘솔에서 직접 실행하면서 분석할 수 있습니다. 커널 소스는 커널 모드에서 실행해야 하는 특성으로 인해서 소스를 빌드하고 실행 및 디버깅하기가 상당히 까다롭습니다. 커널연구회는 그동안 커널 소스를 분석하면서 이런 불편한 점들을 해결하기 위해서 많은 노력을 했습니다. 커널 소스 중에서 하드웨어 의존적인 소스는 커널 모드에서 하드웨어를 직접 제어해야 하므로 따로 분리해서 실행하기 어렵습니다. 그러나 하드웨어에 상관없이 소프트웨어 알고리즘으로 동작하는 소스들(자료구조, 구조체, 공통 라이브러리, 스케줄러 등)은 별도로 분리하여 사용자 어플리케이션으로 실행할 수 있습니다. 이렇게 하면 소스를 분석 및 디버깅하기 좋아지고 소스도 쉽게 익힐 수 있습니다. 커널연구회는 다년간의 노력과 노하우로 커널 소스 중에서 단독으로 분리할 수 있는 소스들을 파일 단위로 분리하여 qt 프로젝트로 구성하여 사용자 콘솔 어플리케이션으로 실행할 수 있도록 했습니다. 이렇게 하면 커널 소스를 행단위로 하나씩 실행하면서 자세히 분석 및 디버깅 할 수 있습니다.

리눅스 커널 소스는 다음과 같이 분류할 수 있는데, 아래에서 굵은 글자로 된 항목이 사용자 어플리케이션 소스로 따로 분리할 수 있는 것들입니다.

- 환경설정 소스(Kconfig)
- 초기화 소스(init)
- 자료구조 구조체 소스(include/linux)
- 공통 라이브러리 소스(lib)
- 공유자원 접근제어(Locking) 소스

커널 소스 실행 분석기 (Quick Start Guide)

- CPU 아키텍처 의존 소스(arch)
- 메모리 의존 소스(mm)
- 스케줄러 소스(sched)
- 디바이스 트리 파싱 소스(drivers/of)
- 디바이스 코어 소스(drivers/base)
- 하드웨어 의존 디바이스 드라이버 소스들(drivers)

커널연구회에서 제작한 "리눅스 커널 소스 실행 분석기"는 리눅스 커널 소스들중에서 환경설정, 초기화, 자료구조, 구조체, 공통 라이브러리, 스케줄러, 디바이스 트리 파싱, 디바이스 코어 소스를 따로 분리 했습니다. 분리한 소스들은 파일 단위로 qt 어플리케이션 프로젝트로 구성하여 qt 에서 실행 및 디버깅할 수 있습니다. 또한 사용자 콘솔 화면에서 직접 실행하여 소스를 분석할 수 있습니다.

소스 파일 경로 구조는 리눅스 커널 소스 구조와 동일하게 구성하여 소스를 학습할때 혼선없이 일관성을 유지할 수 있도록 했습니다. 리눅스 커널 소스는 Long Term 안정화 버전인 v5.4 을 기준으로 분석기에 적용 했습니다.

1.1 소스 경로 구조

“리눅스 커널 실행 분석기” 소스는 다음과 같이 구성되어 있습니다. 경로 명칭은 리눅스 커널 소스와 동일하게 했습니다. 소스 경로별 의미들은 아래 목록에서 주석문 형태로 설명 합니다.

분석기 소스 구조 (소스 목록)

```
kernel-bz/linux-kernel-test$ ll
total 144
drwxr-xr-x 13 jungjaejoon 4096 1월 30 21:58 ./
drwxr-xr-x 7 jungjaejoon 4096 1월 3 09:45 ../
/* CPU 아키텍처에 관련된 소스 경로 */
drwxr-xr-x 6 jungjaejoon 4096 1월 23 14:23 arch/
/* 문서(매뉴얼) 경로 */
drwxr-xr-x 4 jungjaejoon 4096 1월 23 14:06 docs/
/* 디바이스 드라이버 관련 소스 경로 */
drwxr-xr-x 4 jungjaejoon 4096 1월 23 14:06 drivers/
/* git 관련 경로 */
drwxr-xr-x 8 jungjaejoon 4096 1월 30 14:54 .git/
-rw-r--r-- 1 jungjaejoon 462 1월 23 14:06 .gitignore
/* 인클루드 헤더파일(자료구조 구조체) 경로 */
drwxr-xr-x 11 jungjaejoon 4096 1월 23 14:23 include/
/* 초기화 관련 경로 */
drwxr-xr-x 2 jungjaejoon 4096 1월 28 12:27 init/
/* 커널 스케줄러 관련 경로 */
drwxr-xr-x 6 jungjaejoon 4096 1월 28 12:27 kernel/
/* qt 프로젝트 파일 */
-rw-r--r-- 1 jungjaejoon 20244 1월 28 12:27 kernel-test.pro
-rw-r--r-- 1 jungjaejoon 24109 1월 30 21:58 kernel-test.pro.user
/* 공통 라이브러리 관련 경로 */
drwxr-xr-x 8 jungjaejoon 4096 1월 30 14:53 lib/
/* 라이선스(GPL3) 파일 */
-rw-r--r-- 1 jungjaejoon 35149 9월 27 08:00 LICENSE
/* 메인 시작 소스 파일 */
-rw-r--r-- 1 jungjaejoon 2030 1월 30 14:53 main.c
/* 메모리 관련 소스 경로 */
```

커널 소스 실행 분석기 (Quick Start Guide)

```
drwxr-xr-x  2 jungjaejoon  4096  1월 30 14:53 mm/
/* git 도움말 파일 */
-rw-r--r--  1 jungjaejoon   296  9월 27 08:00 README.md
/* 각종 스크립트 관련 소스 경로 */
drwxr-xr-x  3 jungjaejoon  4096  1월 23 14:06 scripts/
/* 사용자 테스트 소스 경로 */
drwxr-xr-x 11 jungjaejoon  4096  1월 28 12:27 test/
```

위의 소스 경로중에서 test/ 경로에 있는 소스들은 커널연구회에서 새롭게 추가한 소스들이고 나머지 소스들은 리눅스 커널 소스 구조와 동일합니다.

커널연구회에서 추가한 test 경로 소스 목록

```
kernel-bz/linux-kernel-test$ ll test/

drwxr-xr-x 11 jungjaejoon 4096  1월 28 12:27 ./
drwxr-xr-x 13 jungjaejoon 4096  2월  1 11:08 ../
drwxr-xr-x  2 jungjaejoon 4096  1월 30 14:53 algorithm/
drwxr-xr-x  2 jungjaejoon 4096  1월 30 14:53 basic/
drwxr-xr-x  2 jungjaejoon 4096  1월 30 14:53 config/
drwxr-xr-x  2 jungjaejoon 4096  1월 30 14:53 drivers/
drwxr-xr-x  2 jungjaejoon 4096  1월 28 12:27 init/
drwxr-xr-x  2 jungjaejoon 4096  1월 30 14:53 menu/
drwxr-xr-x  5 jungjaejoon 4096  1월 23 14:06 sample/
drwxr-xr-x  2 jungjaejoon 4096  1월 28 12:27 sched/
drwxr-xr-x  2 jungjaejoon 4096  1월 28 12:27 user/
```

위의 소스들에 대해서는 다음장부터 설명 됩니다.

1.2 qt 프로젝트

위의 소스 파일들은 qt 에서 프로젝트로 구성 했습니다. qt 프로젝트 파일명은 "kernel-test.pro" 입니다. qt 는 윈도우즈의 비주얼 스튜디오 처럼 리눅스 환경의 어플리케이션 통합개발환경 입니다. 리눅스에서 qt 를 사용해본 독자분들은 금방 이것의 효용성을 알고 있을 것입니다. 혹시 처음 접하신다면 다음과 같이 qt 을 설치합니다.(우분투 배포본에서 실행)

qt 설치하기

```
$ sudo apt-get install build-essential qtcreator qt5-default qt5-doc
```

qt 를 설치한후 정보를 확인해보면 다음과 같습니다.

qt5-default 패키지 설치 정보

```
$ apt show qt5-default

Package: qt5-default
Version: 5.9.5+dfsg-0ubuntu2.5
Priority: optional
Section: universe/libs
Source: qtbase-opensource-src
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian Qt/KDE Maintainers <debian-qt-kde@lists.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 159 kB
Depends: qtbase5-dev, qtchooser (>= 55-gc9562a1-1~)
Suggests: qt5-qmake, qtbase5-dev-tools
Conflicts: qt4-default
Homepage: http://qt-project.org/
Download-Size: 20.9 kB
APT-Manual-Installed: yes
APT-Sources: http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
Description: Qt 5 development defaults package
```

커널 소스 실행 분석기 (Quick Start Guide)

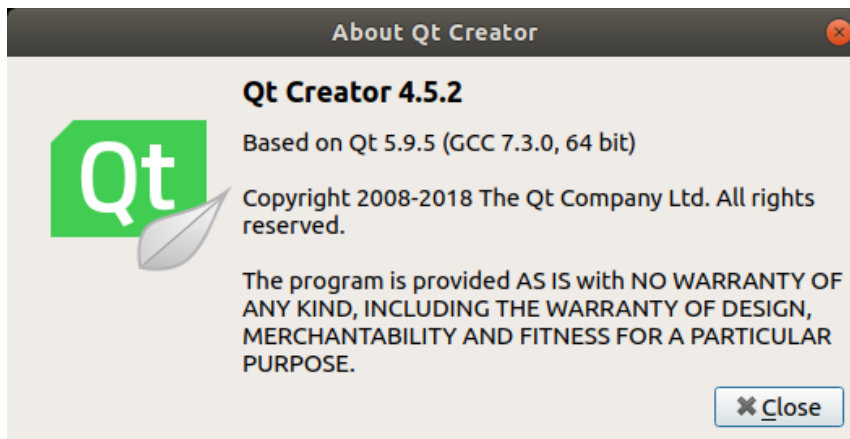
qtcreator 패키지 설치 정보

```
$ apt show qtcreator
```

```
Package: qtcreator
Version: 4.5.2-3ubuntu2
Priority: optional
Section: universe/devel
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian Qt/KDE Maintainers <debian-qt-kde@lists.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 56.3 MB
Depends: libqt5sql5-sqlite, qml-module-qtqml-models2, qml-module-qtquick-controls (>= 5.5.0~), qml-module-qtquick2 (>= 5.5.0~), qtchooser, qtcreator-data (= 4.5.2-3ubuntu2), libbotan-2-4, libc6 (>= 2.14), libclang1-6.0 (>= 1:5.0~+rc1~), libgcc1 (>= 1:3.0), libqbscore1.10 (>= 1.10.0), libqbsqtprofilessetup1.10 (>= 1.2.1), libqt5concurrent5 (>= 5.6.2~), libqt5core5a (>= 5.9.0~beta), libqt5designer5 (>= 5.6.2~), libqt5designercomponents5 (>= 5.6.2~), libqt5gui5 (>= 5.8.0), libqt5help5 (>= 5.6.2~), libqt5network5 (>= 5.6.2~), libqt5printsupport5 (>= 5.6.2~), libqt5qml5 (>= 5.1.0), libqt5quick5 (>= 5.9.0~beta), libqt5quickwidgets5 (>= 5.9.0~beta), libqt5sql5 (>= 5.6.2~), libqt5widgets5 (>= 5.7.0), libqt5xml5 (>= 5.6.2~), libstdc++6 (>= 7), qtdeclarative-abi-5-9-5
Recommends: clang, gdb, make, qmlscene, qt5-doc, qt5-qmltooling-plugins, qtbase5-dev-tools, qtcreator-doc, qtdeclarative5-dev-tools, qttools5-dev-tools, qttranslations5-l10n, qtxmlpatterns5-dev-tools, xterm | x-terminal-emulator
Suggests: cmake, g++, git, kate-data, subversion
Breaks: qtcreator-data (<< 4.5.2-2~)
Replaces: qtcreator-data (<< 4.5.2-2~)
Homepage: https://doc.qt.io/qt-5/topics-app-development.html
Download-Size: 14.0 MB
APT-Manual-Installed: yes
APT-Sources: http://kr.archive.ubuntu.com/ubuntu bionic/universe amd64
Packages
```

Qt 는 5.9 이상 버전, Qt Creator 는 4.5 이상 버전을 권장 합니다. 버전이 너무 낮으면 GCC 컴파일러를 자동으로 검색하지 못하기 때문에 컴파일러 설정하는데 어려움이 생길 수 있습니다.

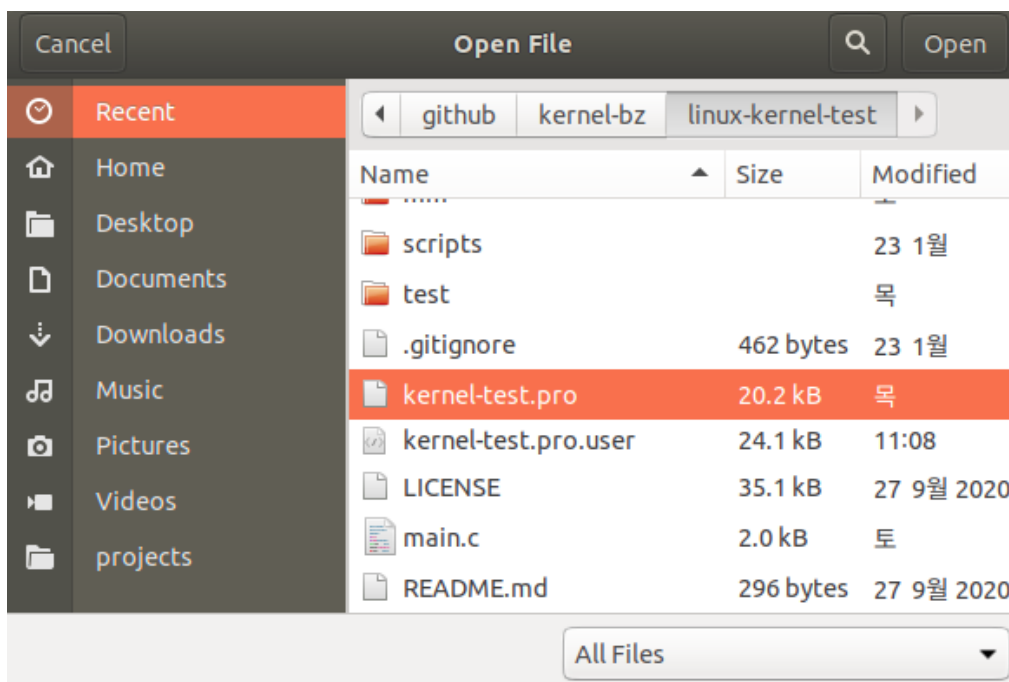
커널 소스 실행 분석기 (Quick Start Guide)



다음과 같이 qtcreeator 을 실행하여 프로그램 화면 상단의 File 메뉴를 클릭한후 “Open File or Project”을 실행하여 소스경로에 있는 kernel-test.pro 파일을 오픈 합니다.

qt 실행 및 프로젝트 파일 열기

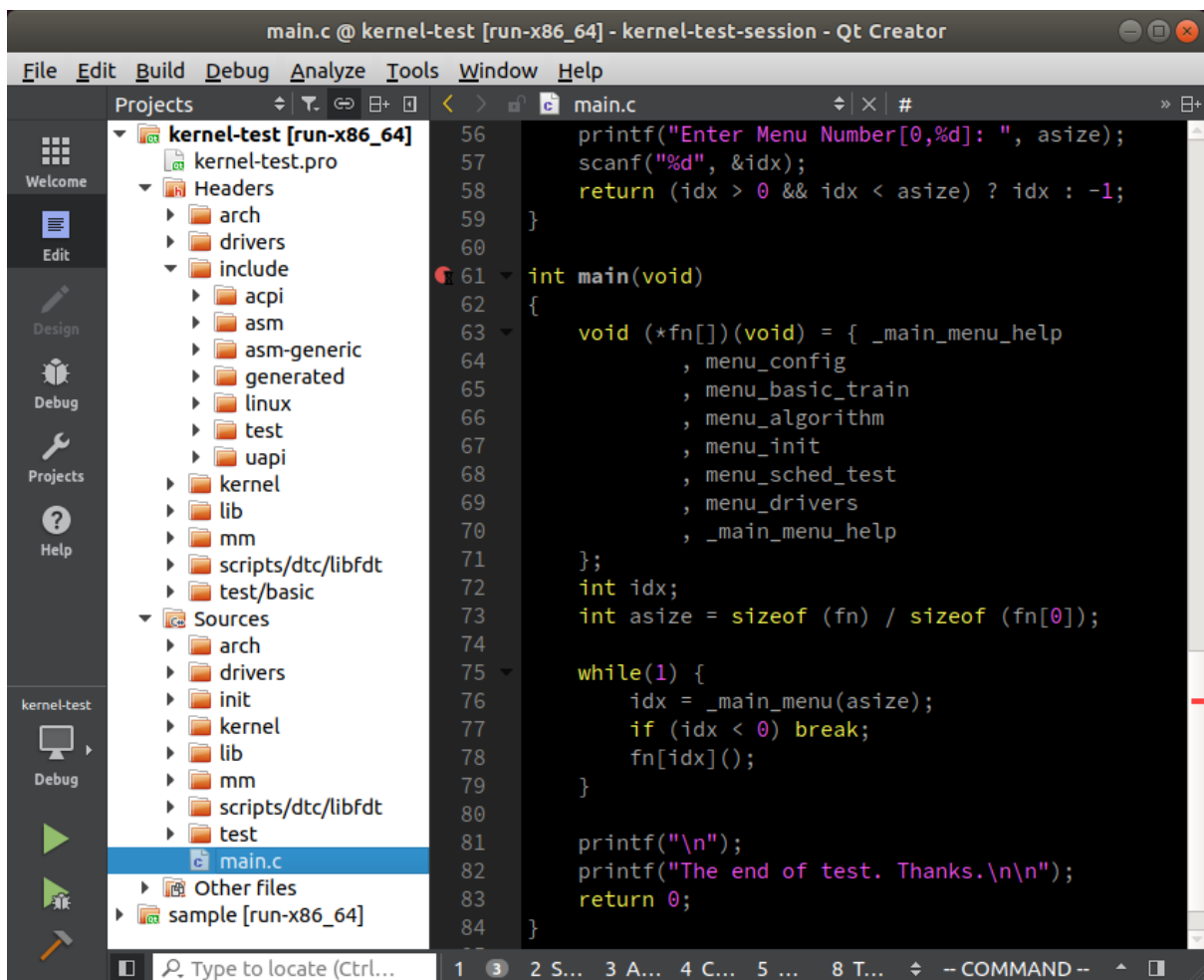
```
$ qtcreeator
```



커널 소스 실행 분석기 (Quick Start Guide)

qt 프로젝트 파일인 "kernel-test.pro" 파일을 오픈하면 다음과 같이 qt 통합개발 환경에 소스 파일들이 적재 됩니다. 커널 소스 분석기 소스는 화면 좌측의 "Projects" 리스트에 트리구조로 나타납니다. 이곳에 있는 파일들을 클릭하면 화면 우측 편집기에 소스가 나타나고 이것을 qt 어플리케이션으로 바로 실행 및 디버깅할 수 있습니다.

qt 실행 화면



1.3 userspace RCU 패키지

RCU(Read-copy update)는 공유된 자원 보호(Locking)를 위해서 2002년부터 커널에 적용되었습니다. 일반적인 Locking 보다 성능을 개선하는 방향으로 발전했으며 자세한 내용은 아래 웹사이트에서 확인 가능합니다.

<https://liburcu.org/>

RCU 는 사용자(User Level) 어플리케이션에서도 라이브러리를 설치하여 사용할 수 있습니다. 필자의 커널 소스 실행 분석기는 사용자 공간에서 실행 되므로 urcu(userspace RCU) 패키지를 apt-get 을 사용하여 설치 합니다.

userspace RCU 패키지 설치

```
$ sudo apt-get install liburcu-dev
```

위와 같이 설치를 쉽게 할 수 있습니다. 설치된 정보를 확인해 보면 다음과 같습니다.

userspace RCU 설치 내용

```
$ apt show liburcu-dev

Package: liburcu-dev
Version: 0.10.1-1
Priority: extra
Section: libdevel
Source: liburcu
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Jon Bernard <jbernard@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
```

커널 소스 실행 분석기 (Quick Start Guide)

```

Installed-Size: 677 kB
Depends: liburcu6 (= 0.10.1-1)
Homepage: http://liburcu.org/
Supported: 5y
Download-Size: 96.4 kB

```

urcu 라이브러리 및 헤더파일이 있는 경로는 아래와 같이 확인할 수 있습니다.

urcu 라이브러리 및 헤더파일

```

$ ls /usr/local/lib/liburcu*

/usr/local/lib/liburcu.a                /usr/local/lib/liburcu-common.so.6
/usr/local/lib/liburcu-qsbr.la
/usr/local/lib/liburcu-bp.a             /usr/local/lib/liburcu-
common.so.6.0.0 /usr/local/lib/liburcu-qsbr.so
/usr/local/lib/liburcu-bp.la            /usr/local/lib/liburcu.la
/usr/local/lib/liburcu-qsbr.so.6
/usr/local/lib/liburcu-bp.so            /usr/local/lib/liburcu-mb.a
/usr/local/lib/liburcu-qsbr.so.6.0.0
/usr/local/lib/liburcu-bp.so.6          /usr/local/lib/liburcu-mb.la
/usr/local/lib/liburcu-signal.a
/usr/local/lib/liburcu-bp.so.6.0.0      /usr/local/lib/liburcu-mb.so
/usr/local/lib/liburcu-signal.la
/usr/local/lib/liburcu-cds.a            /usr/local/lib/liburcu-mb.so.6
/usr/local/lib/liburcu-signal.so
/usr/local/lib/liburcu-cds.la           /usr/local/lib/liburcu-mb.so.6.0.0
/usr/local/lib/liburcu-signal.so.6
/usr/local/lib/liburcu-cds.so           /usr/local/lib/liburcu-memb.a
/usr/local/lib/liburcu-signal.so.6.0.0
/usr/local/lib/liburcu-cds.so.6         /usr/local/lib/liburcu-memb.la
/usr/local/lib/liburcu.so
/usr/local/lib/liburcu-cds.so.6.0.0     /usr/local/lib/liburcu-memb.so
/usr/local/lib/liburcu.so.6
/usr/local/lib/liburcu-common.a         /usr/local/lib/liburcu-memb.so.6
/usr/local/lib/liburcu.so.6.0.0
/usr/local/lib/liburcu-common.la        /usr/local/lib/liburcu-memb.so.6.0.0
/usr/local/lib/liburcu-common.so        /usr/local/lib/liburcu-qsbr.a

$ ls /usr/local/include/urcu/

```

커널 소스 실행 분석기 (Quick Start Guide)

```

arch          compiler.h  flavor.h    list.h       rculfhash.h  ref.h
tls-compat.h  urcu-bp.h    urcu-memb.h wfcqueue.h
arch.h        config.h   futex.h     map          rculfqueue.h static
uatomic       urcu-futex.h urcu-qsbr.h wfqueue.h
call-rcu.h    debug.h     hlist.h     pointer.h    rculfstack.h syscall-
compat.h      uatomic_arch.h urcu.h       urcu_ref.h   wfstack.h
cds.h         defer.h     lfstack.h   rcuhlist.h   rculist.h    system.h
uatomic.h     urcu-mb.h    urcu-signal.h

```

urcu 헤더파일들은 “/usr/local/include/urcu” 경로에 모두 모여 있습니다. urcu 라이브러리를 어플리케이션에서 사용할때는 다음과 같이 urcu 헤더파일을 #include 로 포함 시키면 됩니다. urcu 라이브러리 함수 사용은 분석기 소스를 설명할때 함수가 나올때 마다 설명하도록 하겠습니다.

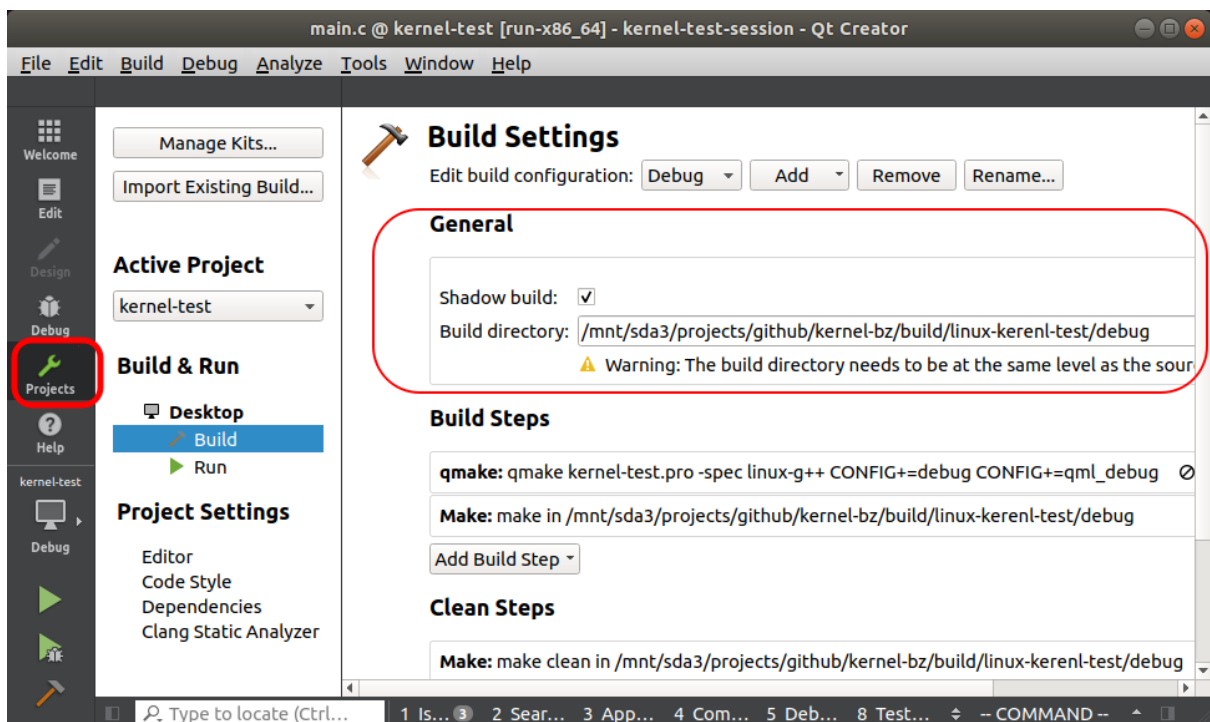
urcu 헤더 파일 인클루드

```
#include <urcu.h>
```

1.4 소스 실행 및 디버깅

qt 프로젝트에 등록되어 있는 소스 파일들이 빌드되어 실행되는 경로는 다음과 같이 qt 화면에서 제일 좌측 가장자리 중간에 있는 Projects 아이콘을 클릭하면 나타납니다.

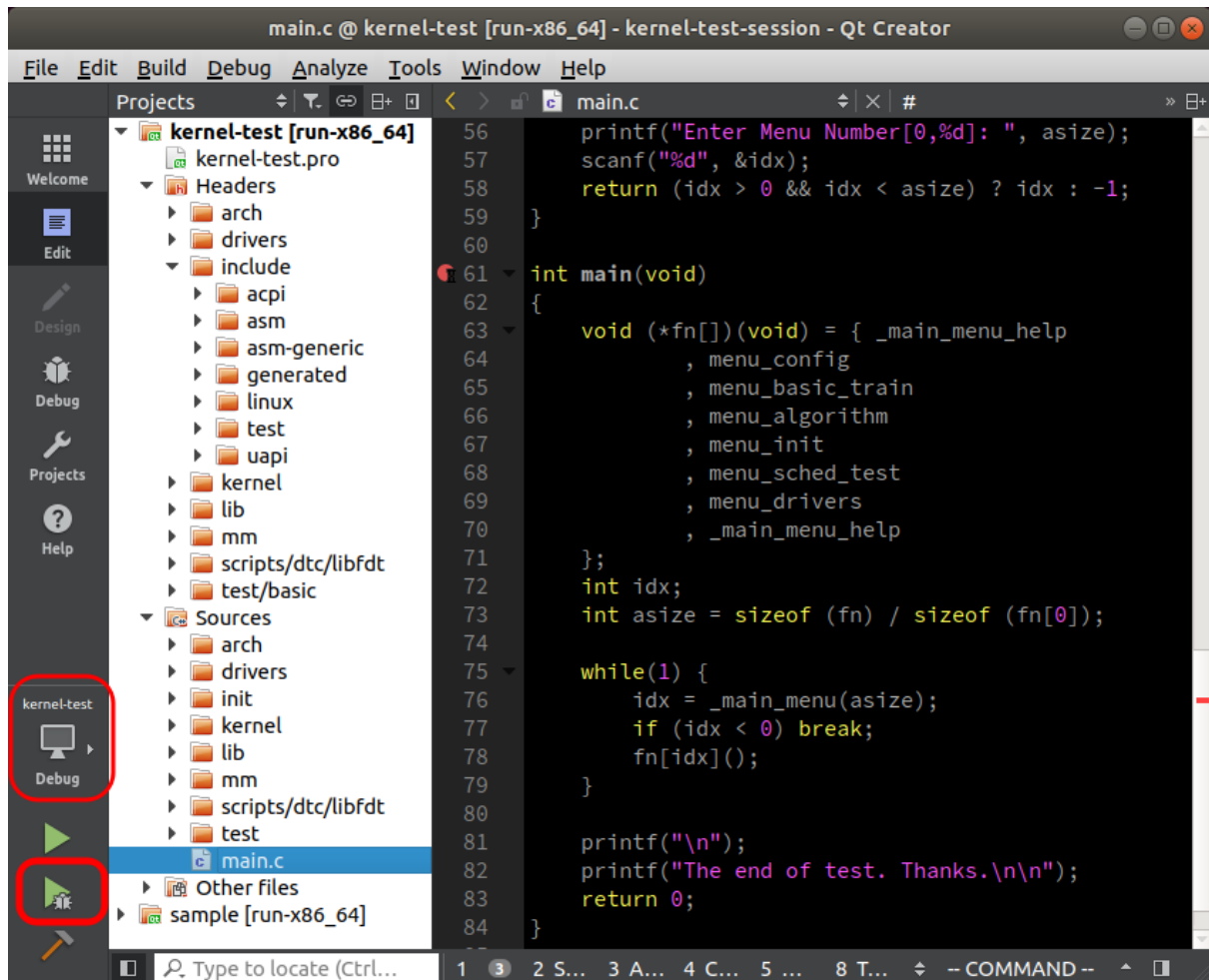
qt 소스 빌드 및 실행 경로



qt 에서 소스를 빌드 및 실행하는 방법은 프로젝트를 Release 모드로 설정하는 방법과 Debug 모드로 설정하는 방법이 있습니다. Release 모드는 소스를 실행하기 가장 좋은 조건으로 최적화(컴파일러 최적화 옵션 작용됨)하여 실행합니다. 이렇게 하면 실행파일 크기가 작아지고 속도는 빨라 집니다만, 소스를 디버깅할 수 없게 됩니다. 그래서 소스를 분석하고자 하는 단계에서는 프로젝트를 Debug 모드로 설정하여 실행합니다. 이것의 설정은 다음과 같이 qt 작업화면에서 제일 왼쪽 하단부 구석에 있는 Debug 아이콘을 클릭하여 선택한후 디버그 버튼을 클릭하여 진행하면 됩니다.

커널 소스 실행 분석기 (Quick Start Guide)

Debug 모드로 설정

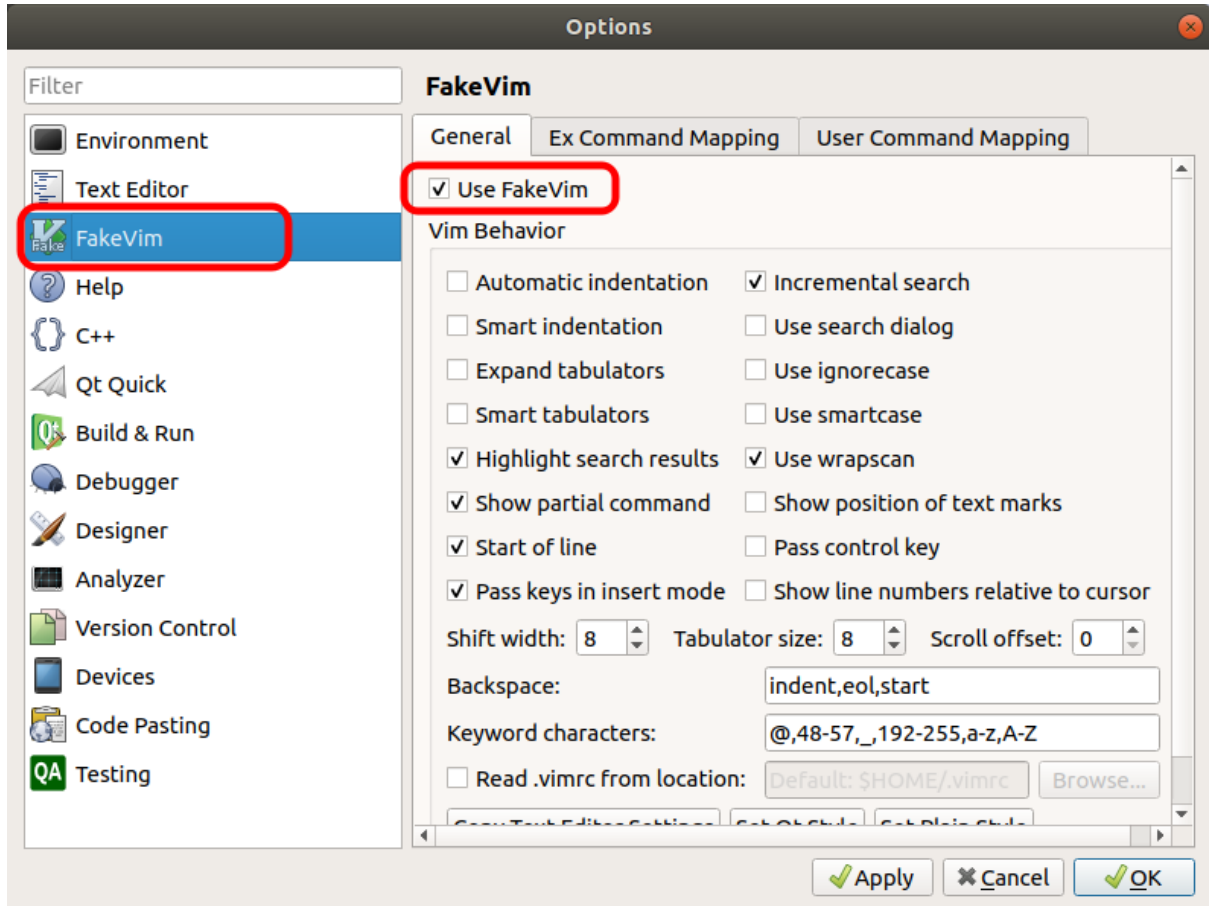


qt 을 사용해본 독자분들은 소스 디버깅을 바로 하면서 분석 진행할 수 있을 것입니다. 혹시 qt 를 처음 접하신다면 인터넷에서 “qt 로 디버깅 하는 방법”으로 검색해 보시면 많은 자료들이 있으므로 참고하시면 되겠습니다.

qt 편집기는 vim 과도 호환됩니다. vim 에 익숙한 독자분들은 qt 편집기 옵션에서 “Use FakeVim”을 선택하면 됩니다. 이렇게 하면 qt 편집 명령들이 vim 편집 명령들과 호환됩니다. vim 에 익숙한 독자분들은 이렇게 설정하여 vim 과 qt 를 같이 열어 놓고 작업 하시면 편리 합니다.

커널 소스 실행 분석기 (Quick Start Guide)

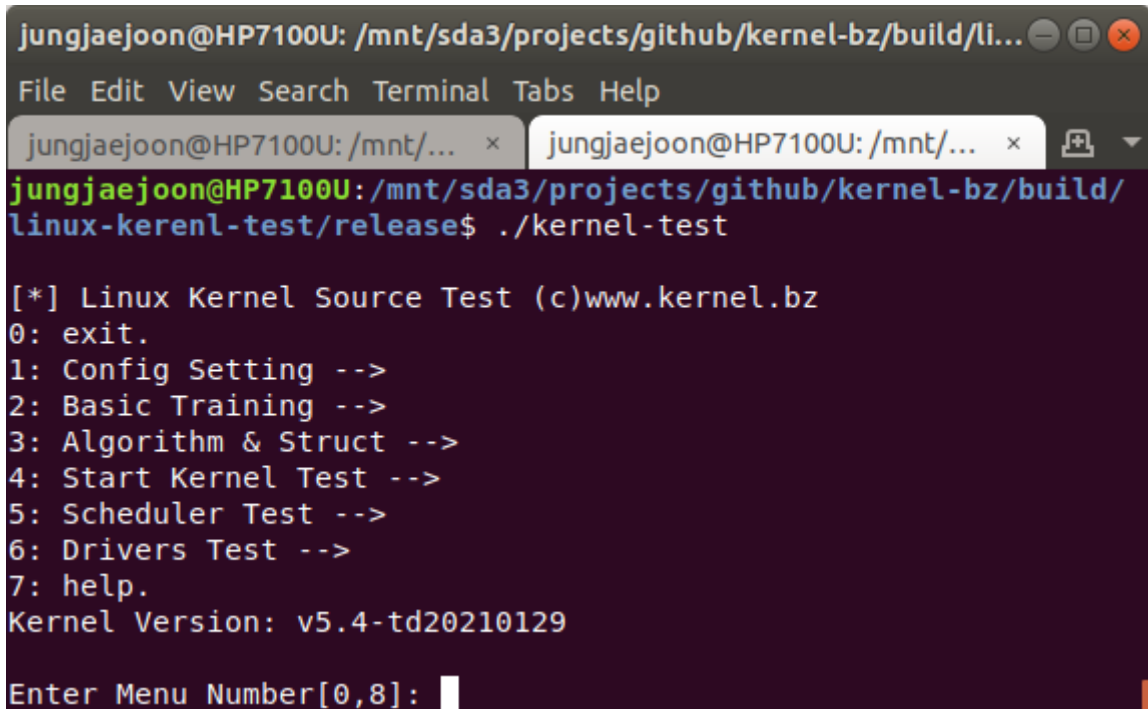
qt 편집기 옵션에서 vim 호환 선택



빌드된 실행 파일(kernel-test)을 리눅스 콘솔 화면에서 다음과 같이 바로 실행할 수도 있습니다.

커널 소스 실행 분석기 (Quick Start Guide)

리눅스 콘솔 화면에서 실행



```
jungjaejoon@HP7100U: /mnt/sda3/projects/github/kernel-bz/build/li...
File Edit View Search Terminal Tabs Help
jungjaejoon@HP7100U: /mnt/... x jungjaejoon@HP7100U: /mnt/... x
jungjaejoon@HP7100U: /mnt/sda3/projects/github/kernel-bz/build/
linux-kerenl-test/release$ ./kernel-test

[*] Linux Kernel Source Test (c)www.kernel.bz
0: exit.
1: Config Setting -->
2: Basic Training -->
3: Algorithm & Struct -->
4: Start Kernel Test -->
5: Scheduler Test -->
6: Drivers Test -->
7: help.
Kernel Version: v5.4-td20210129

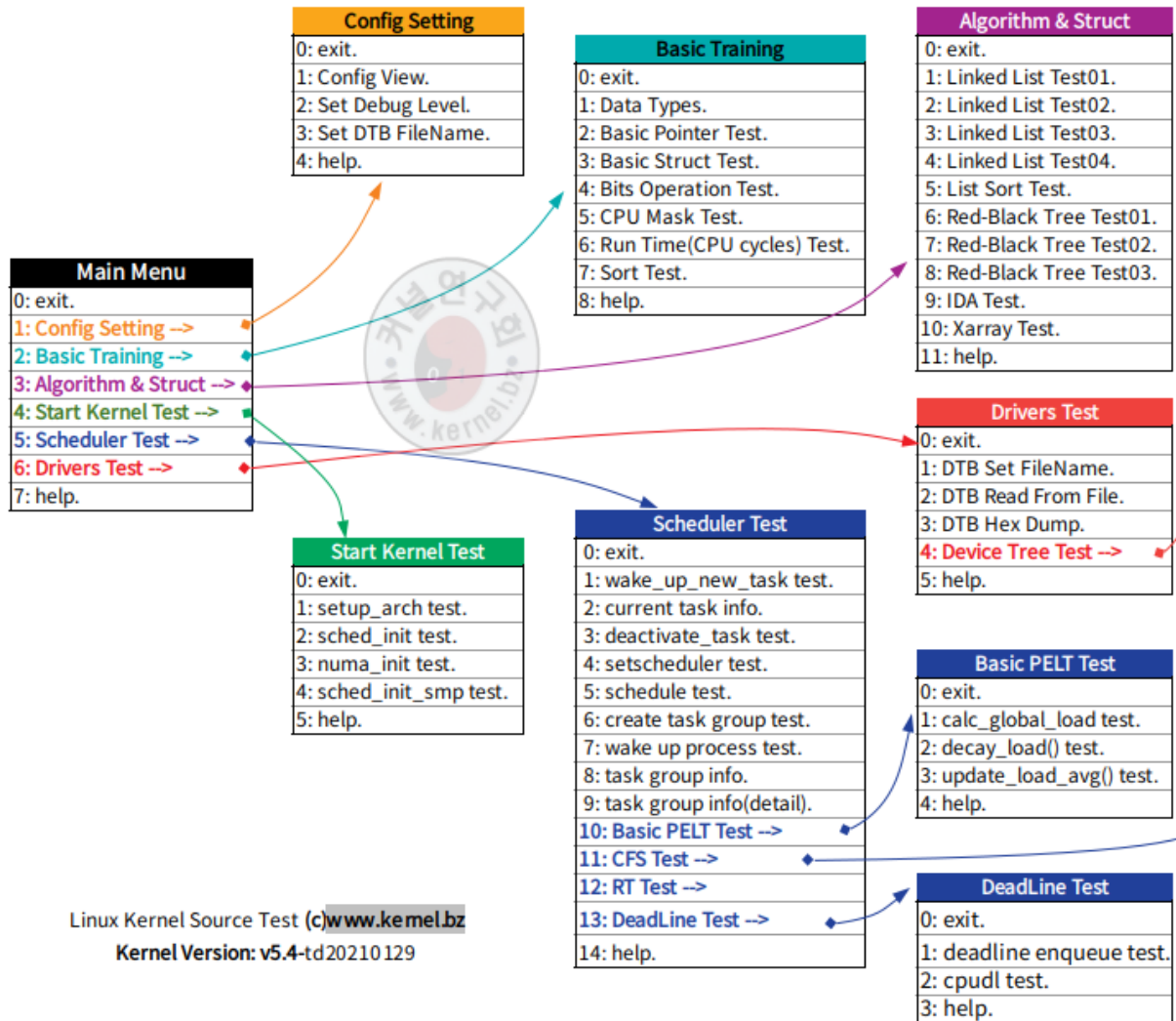
Enter Menu Number[0,8]:
```

커널연구회의 "리눅스 커널 소스 실행 분석기"는 콘솔 화면에 실행된 메뉴별로 커널 소스를 분석할 수 있도록 구성되어 있습니다. 작업 메뉴는 번호를 입력 하도록 되어 있으며, 해당 번호의 메뉴가 실행되면 그것에 해당하는 커널 소스가 실행되는 과정(함수 호출)이 트리 구조 형태로 콘솔 화면에 출력 됩니다. 함수가 실행 되면서 출력되는 데이터들도 자세히 확인하여 소스와 비교하면서 분석할 수 있습니다.

콘솔 화면에서 실행되는 메뉴 구조는 다음과 같이 Main Menu 에서부터 메뉴 순번을 입력하여 실행을 전개할 수 있습니다. 해당 메뉴에서 0 번을 입력하면 앞메뉴로 이동(exit) 합니다.

커널 소스 실행 분석기 (Quick Start Guide)

리눅스 커널 소스 실행 분석기 메뉴



Main Menu 에서부터 Config Setting, Basic Training, Algorithm & Struct, Start Kernel Test, Scheduler Test, Drivers Test 순으로 메뉴번호를 입력하여 실행할 수 있습니다.

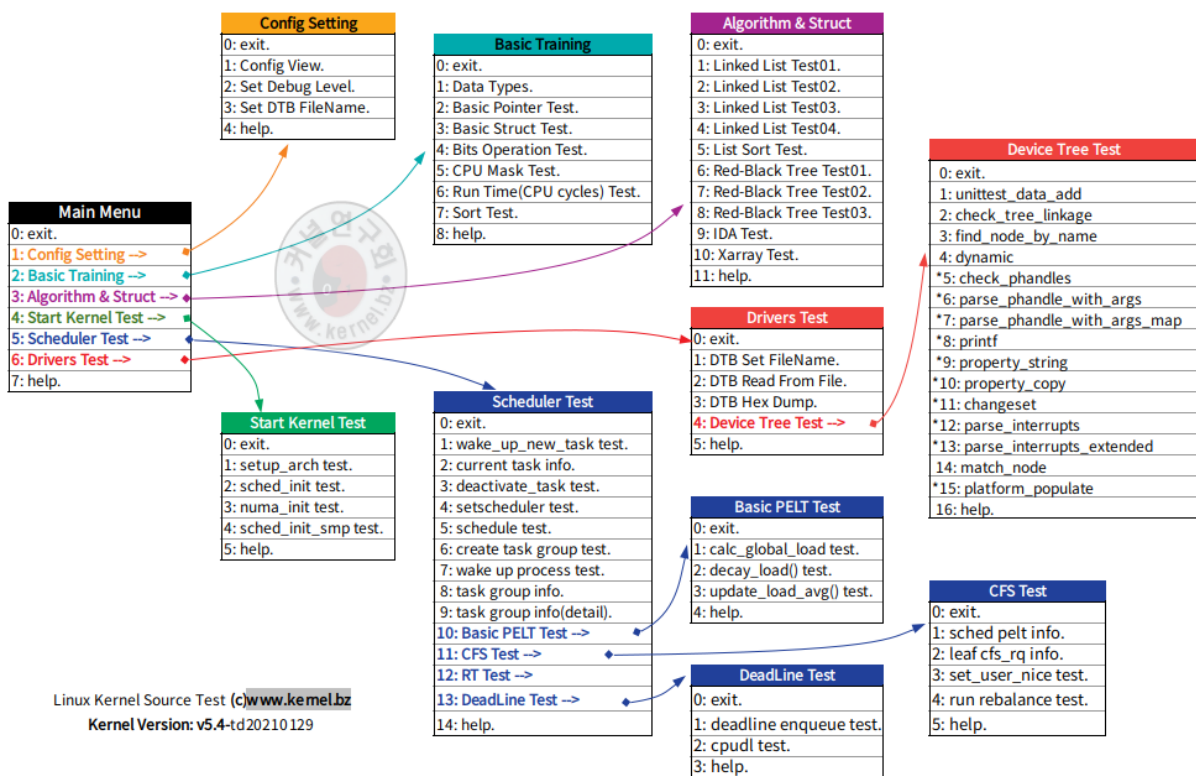
Config Setting 은 소스에 대한 환경 설정을 하는 메뉴이고 Basic Training 과 Algorithm & Struct 은 커널 소스를 이해하기 위해서 반드시 선행 학습해야 하는 소스들로 구성되어 있습니다. 이것이 숙지되면 Start Kernel Test 메뉴에서 커널 초기화 루틴들에 대한 소스를 익힐 수 있고, Scheduler

커널 소스 실행 분석기 (Quick Start Guide)

Test 메뉴에서 커널 스케줄러에 관련되는 각종 소스들을 테스트하며 익힐 수 있습니다. 마지막으로 Drivers Test 메뉴에서 디바이스 트리와 드라이버 소스들을 테스트하고 이해할 수 있습니다.

위에서 요약 설명한 메뉴들을 전체적으로 모두 전개하면 다음과 같습니다.

리눅스 커널 소스 실행 분석기 전체 메뉴



위의 메뉴들에서 실행되는 소스에 대한 내용들은 다음장부터 단계적으로 설명됩니다.

1.5 소스 공유(github)

지금까지 요약 설명한 커널연구회의 "리눅스 커널 소스 실행 분석기"의 소스는 github 을 통해서 공유하며 다같이 참여하여 기능을 지속적으로 개선하는 방향으로 진행 합니다.

소스 다운로드

```
$ git clone https://github.com/kernel-bz/linux-kernel-test
```

위와 같이 git clone 으로 다운로드 하면 "linux-kernel-test"라는 경로가 생성되고 그곳에서 다음과 같은 소스를 확인할 수 있습니다.

다운로드 된 소스 내용

```
$ cd linux-kernel-test
$ ll

drwxr-xr-x 13 jungjaejoon 4096 2월 10 15:56 ./
drwxr-xr-x  7 jungjaejoon 4096 1월  3 09:45 ../
drwxr-xr-x  6 jungjaejoon 4096 1월 23 14:23 arch/
drwxr-xr-x  2 jungjaejoon 4096 2월 10 16:13 docs/
drwxr-xr-x  4 jungjaejoon 4096 1월 23 14:06 drivers/
drwxr-xr-x  8 jungjaejoon 4096 2월 10 17:24 .git/
-rw-r--r--  1 jungjaejoon  462 1월 23 14:06 .gitignore
drwxr-xr-x 11 jungjaejoon 4096 1월 23 14:23 include/
drwxr-xr-x  2 jungjaejoon 4096 2월  3 11:43 init/
drwxr-xr-x  6 jungjaejoon 4096 2월 10 10:09 kernel/
-rw-r--r--  1 jungjaejoon 20880 2월 10 15:56 kernel-test.pro
-rw-r--r--  1 jungjaejoon 24111 2월 10 13:49 kernel-test.pro.user
drwxr-xr-x  8 jungjaejoon 4096 2월 10 13:36 lib/
-rw-r--r--  1 jungjaejoon 35149 9월 27 08:00 LICENSE
-rw-r--r--  1 jungjaejoon  2060 2월 10 13:36 main.c
drwxr-xr-x  2 jungjaejoon 4096 2월 10 10:09 mm/
-rw-r--r--  1 jungjaejoon  296 9월 27 08:00 README.md
```

커널 소스 실행 분석기 (Quick Start Guide)

```
drwxr-xr-x  3  jungjaejoon  4096  1월 23 14:06 scripts/
drwxr-xr-x 11  jungjaejoon  4096  2월  3 11:43 test/
```

아울러, 커널연구회 멤버십에 가입하시면 다음과 같은 혜택이 주어 집니다.

멤버십 혜택:

- ➔ 커널연구회 제작 커널 소스 분석기 제공.
- ➔ 커널 소스 분석기 github 접근 계정(commit 권한) 제공.
- ➔ 커널연구회 소스 실시간 동영상 강의 제공.
- ➔ 오프라인 강화 교육 참여권 제공.
- ➔ 커널 소스 메인테이너 맞춤형 성장 지원.
- ➔ 커널 엔지니어 인증 수료증서 제공.
- ➔ 커널 엔지니어로 성장할 수 있도록 멤버십 지원.
- ✓ 커널연구회 멤버십에는 가입비 및 월회비가 있습니다.(상담 및 면접시 비용 결정)

또한, 자립형 사립으로 개설되어 있는 커널연구회에 입소하실 수도 있습니다. 입소생은 학기별로 일년에 두번(1 월과 7 월) 소수 정예인원을 모집합니다. 입소생에게는 커널연구회 멤버십 혜택과 함께 연구소에 상주하면서 커널 엔지니어로 성장할 수 있는 맞춤형 교육 프로그램들을 진행합니다. 입소비용은 면접을 통해서 결정하며, 좀더 자세한 사항들은 커널연구회 홈페이지(www.kernel.bz) 참조 바랍니다.