라즈베리파이 커널 컴파일

목표

- 커널 컴파일이 무엇인지 간단히 살펴본 뒤 커널 컴파일에 필요한 패키지들을 호스트환경에서 설치한다.
- Raspberry Pi(RPI)의 커널을 호스트 PC 환경에서 컴파일한 뒤 생성된 이미지를 RPI로 옮겨 부팅하는 것을 진행해본다.

준비물

- 호스트 환경 Linux PC (가상머신에 올려져 있는 Linux PC라도 무관)
- Raspberry Pi (Raspbian OS가 올라가 있는)

***중요)** 본 실습은 **RPI 3B+ 이상의 모델**에 맞추어 설명이 진행된다.

커널 컴파일(Kernel Compile)이란?

- 운영체제(Operating System)란?
  - 컴퓨터 시스템의 자원들을 효율적으로 관리하며, **사용자가 컴퓨터를 편리하게, 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임**이다.
  - 운영체제는 **컴퓨터 사용자와 컴퓨터 하드웨어 간의 인터페이스**로서 동작하는 시스템 소프트웨어의 일종으로, 다른 응용프로그램이 유용한 작업을 할 수 있도록 환경을 제공해준다.

- 커널(Kernel)이란?
  - 운영체제는 커널을 감싸고 있는 존재이다. 즉, 다시 말해 운영체제의 한 요소에는 커널이 포함되는 것이다.
  - 운영체제와 같이 덩치가 큰 프로그램은 메모리에 항상 상주해 있을 수 없다. 그래서 그때 그때 필요한 모듈들은 메모리에 올라왔다가 다시 꺼지기를 반복한다. 하지만, 하드웨어와 밀접한 관계를 가지는 프로그램(Memory, Storage 등을 제어하는)은 항시 메모리 내에 상주해 있어야 한다. 이러한 **하드웨어와 밀접한 관계를 가지며 컴퓨터 시스템에 반드시 필요한 동작을 수행해야 하는 프로그램**을 커널이라 한다.

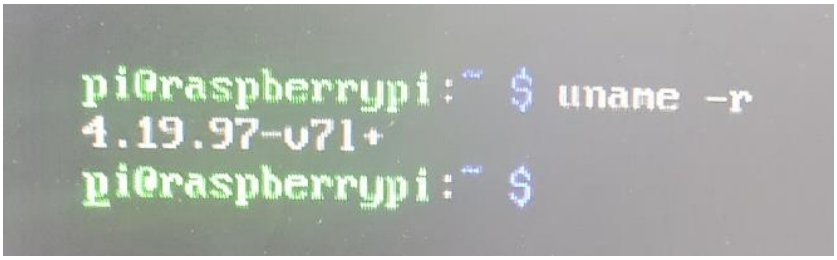
- 본 실습에서는 이러한 커널을 **호스트 환경**에서 직접 컴파일해보고 RPI로 옮긴 뒤 정상적으로 부팅되는지 살펴본다.
  실습을 진행하기 전에 앞서 필요한 패키지들을 설치해준다.

  - $ sudo apt-get install git
  - $ sudo apt-get install make
  - $ sudo apt-get install bison flex libssl-dev bc
  - $ sudo apt-get install gcc-arm-linux-gnueabihf

호스트환경에서 RPI 전용 커널소스 내려받기

- RPI 3B+와 4 모델에서 지원하는 커널 4.19 버전을 사용해 컴파일을 진행해볼것이다.

먼저, 현재 RPI의 커널 버전을 확인해준다. (명령어 uname –r)



- 현재 4.19.97로 커널 버전이 나타난다.
- 커널 컴파일을 끝난 뒤에는 앞의 4.19 버전은 동일하며 뒤의 97이 변경될 것이다.

- 이제 RPI 커널 4.19 버전 소스를 Git을 통해 다운받아 본다. 명령어는 다음과 같다.

  $ git clone --branch rpi-4.19.y https://github.com/raspberrypi/linux.git ~/rpi-linux

```
wlgns12www@VM:~$ git clone --branch rpi-4.19.y https://github.com/raspberrypi/linux.git ~/rpi-linux
Cloning into '/home/wlgns12www/rpi-linux'...
remote: Enumerating objects: 9156575, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 9156575 (delta 3), reused 1 (delta 0), pack-reused 9156562
Receiving objects: 100% (9156575/9156575), 2.69 GiB | 4.67 MiB/s, done.
Resolving deltas: 100% (7659813/7659813), done.
Checking out files: 100% (62377/62377), done.
wlgns12www@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  rpi-linux  Templates  test  Videos
wlgns12www@VM:~$
```

- 본 실습에서는 커널 소스를 홈 디렉터리(~/)의 rpi-linux로 내려 받도록 하였다.

- 생성된 커널 소스 디렉터리로 이동 (본 실습에서는 rpi-linux로 생성을 하였으니 이 디렉터리를 기준으로 설명)

  $ cd ~/rpi-linux

```
wlgns12www@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  rpi-linux  Templates  test  Videos
wlgns12www@VM:~$ cd ~/rpi-linux
wlgns12www@VM:~/rpi-linux$ ls
arch      CREDITS        firmware  ipc       lib          mm        scripts   usr
block     crypto         fs        Kbuild    LICENSES     net       security  virt
certs     Documentation  include   Kconfig   MAINTAINERS  README    sound
COPYING   drivers        init      kernel    Makefile     samples   tools
```

- 이제 내려 받은 커널 소스를 컴파일 하도록 한다.

호스트환경에서 커널 컴파일 진행

- **.config 파일 생성

  $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2711_defconfig (**RPI 4 이상 모델**)

  $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2709_defconfig (**RPI 2 이상 모델**)

```
wlgns12www@VM:~/rpi-linux$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2711_defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  YACC    scripts/kconfig/zconf.tab.c
  LEX     scripts/kconfig/zconf.lex.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
```

**.config 파일 : 커널 컴파일 시 필요한 커널 모듈들을 컴파일 할 수 있게 지정해주는 파일이다.

- 커널 컴파일 진행

  $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j$(nproc) zImage modules dtbs

```
wlgns12www@VM:~/rpi-linux$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j$(nproc) zImage modules dtbs
  SYSHDR  arch/arm/include/generated/uapi/asm/unistd-common.h
  SYSHDR  arch/arm/include/generated/uapi/asm/unistd-oabi.h
  UPD     include/config/kernel.release
  SYSHDR  arch/arm/include/generated/uapi/asm/unistd-eabi.h
  WRAP    arch/arm/include/generated/uapi/asm/bitsperlong.h
  WRAP    arch/arm/include/generated/uapi/asm/bpf_perf_event.h
  WRAP    arch/arm/include/generated/uapi/asm/errno.h
  WRAP    arch/arm/include/generated/uapi/asm/ioctl.h
  WRAP    arch/arm/include/generated/uapi/asm/ipcbuf.h
  WRAP    arch/arm/include/generated/uapi/asm/msgbuf.h
  WRAP    arch/arm/include/generated/uapi/asm/param.h
  UPD     include/generated/uapi/linux/version.h
  WRAP    arch/arm/include/generated/uapi/asm/poll.h
  WRAP    arch/arm/include/generated/uapi/asm/resource.h
  WRAP    arch/arm/include/generated/uapi/asm/sembuf.h
  WRAP    arch/arm/include/generated/uapi/asm/shmbuf.h
  WRAP    arch/arm/include/generated/uapi/asm/siginfo.h
  WRAP    arch/arm/include/generated/uapi/asm/socket.h
  WRAP    arch/arm/include/generated/uapi/asm/sockios.h
  WRAP    arch/arm/include/generated/uapi/asm/termbits.h
  WRAP    arch/arm/include/generated/uapi/asm/termios.h
```

  ➢ 이 작업은 최소 20분이 소요된다. (호스트 PC의 성능에 따라 더 빨리 끝날 수도 있고 늦게 끝날 수도 있음)

- 커널 컴파일이 완료될 시 다음과 같은 파일이 생성되어 있을 것이다.

```
wlgns12www@VM:~/rpi-linux$ ls
arch         crypto         init      LICENSES         Module.symvers  sound       vmlinux.o
block        Documentation  ipc       MAINTAINERS      net             System.map
built-in.a   drivers        Kbuild    Makefile         README          tools
certs        firmware       Kconfig   mm               samples         usr
COPYING      fs             kernel    modules.builtin  scripts         virt
CREDITS      include        lib       modules.order    security        vmlinux
wlgns12www@VM:~/rpi-linux$ ls ./arch/arm/boot/
bootp  compressed  deflate_xip_data.sh  dts  Image  install.sh  Makefile  zImage
```

  ➢ zImage가 메모리에 올라갈 커널 이미지이다.

- 커널 모듈 컴파일 진행

  $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- INSTALL_MOD_PATH=~/modules -j$(nproc) modules_install

```
wlgns12www@VM:~/rpi-linux$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- INSTALL_MOD_PATH=~/
modules -j$(nproc) modules_install
  INSTALL arch/arm/crypto/aes-arm-bs.ko
  INSTALL arch/arm/crypto/aes-arm.ko
  INSTALL arch/arm/crypto/sha1-arm-neon.ko
  INSTALL arch/arm/crypto/sha1-arm.ko
  INSTALL arch/arm/lib/xor-neon.ko
  INSTALL arch/arm/oprofile/oprofile.ko
  INSTALL crypto/af_alg.ko
  INSTALL crypto/algif_skcipher.ko
  INSTALL crypto/arc4.ko
  INSTALL crypto/async_tx/async_memcpy.ko
  INSTALL crypto/async_tx/async_pq.ko
  INSTALL crypto/async_tx/async_raid6_recov.ko
  INSTALL crypto/async_tx/async_tx.ko
  INSTALL crypto/async_tx/async_xor.ko
  INSTALL crypto/authenc.ko
  INSTALL crypto/authencesn.ko
  INSTALL crypto/cast5_generic.ko
```

- 커널 모듈 컴파일이 완료되었다면 홈 디렉터리(~/)의 modules에 커널 모듈들이 생성되어 있을 것이다.

```
wlgns12www@VM:~/rpi-linux$ ls ~/modules/lib/modules/
4.19.127-v7l+
```

  ➢ 다음 디렉터리 안에 커널 모듈들이 들어있다.

- 커널 모듈 디렉터리에서 불필요한 링크 파일 삭제

$ rm ~/module/lib/modules/[kernel version]/build

$ rm ~/module/lib/modules/[kernel version]/source

```
wlgns12www@VM:~$ ls ~/modules/lib/modules/4.19.127-v7l+/
build           modules.alias.bin    modules.dep      modules.order    modules.symbols.bin
kernel          modules.builtin      modules.dep.bin  modules.softdep  source
modules.alias   modules.builtin.bin  modules.devname  modules.symbols
wlgns12www@VM:~$ rm ~/modules/lib/modules/4.19.127-v7l+/build
wlgns12www@VM:~$ rm ~/modules/lib/modules/4.19.127-v7l+/source
wlgns12www@VM:~$ ls ~/modules/lib/modules/4.19.127-v7l+/
kernel             modules.builtin      modules.dep.bin  modules.softdep
modules.alias      modules.builtin.bin  modules.devname  modules.symbols
modules.alias.bin  modules.dep          modules.order    modules.symbols.bin
```

➢ 해당 링크는 커널 컴파일 시 사용된 소스파일들을 가르키는 링크 파일이다.
방대한 소스파일들을 다 보내게 되면 시간이 너무 오래 걸리니 위 두 링크를 삭제해주도록 한다.

- 커널 이미지와 모듈의 컴파일이 완료 되었으니 RPI로 이동시켜 적용시켜 보도록 한다.

RPI에 커널 적용 및 부팅

- 홈 디렉터리에 생성된 커널 이미지와 모듈을 저장할 디렉터리를 하나 생성한다.

$ mkdir ~/kernel

```
wlgns12www@VM:~/rpi-linux$ mkdir ~/kernel
wlgns12www@VM:~/rpi-linux$ ls ~
Desktop    Downloads  modules  Pictures  rpi-linux  test
Documents  kernel     Music    Public    Templates  Videos
```

- 생성한 디렉터리로 커널 이미지와 모듈을 복사한다.

$ cp ~/rpi-linux/arch/arm/boot/zImage ~/kernel

$ cp ~/rpi-linux/arch/arm/boot/dts/*.dtb ~/kernel

$ cp ~/rpi-linux/arch/arm/boot/dts/overlays/*.dtb* ~/kernel

$ cp -r ~/modules/lib/modules/[kernel version] ~/kernel

```
wlgns12www@VM:~$ ls
Desktop    Downloads  modules  Pictures  rpi-linux  test
Documents  kernel     Music    Public    Templates  Videos
wlgns12www@VM:~$ cp ~/rpi-linux/arch/arm/boot/zImage ~/kernel
wlgns12www@VM:~$ cp ~/rpi-linux/arch/arm/boot/dts/*.dtb ~/kernel
wlgns12www@VM:~$ cp ~/rpi-linux/arch/arm/boot/dts/overlays/*.dtb* ~/kernel
wlgns12www@VM:~$ cp -r ~/modules/lib/modules/4.19.127-v7l+ ~/kernel
```

- 정상적으로 복사되었는지 확인한다.
  ➢ 본 실습의 결과와는 차이가 존재할 수도 있다.
  { *.dtb, *.dtb*, [kernel module directory], zImage } -> 이러한 확장자와 커널 모듈 디렉터리, 커널 이미지만 대략적으로 존재한다면 정상적으로 복사가 완료된 것이다.

```
wlgns12www@VM:~$ ls ~/kernel/
4.19.127-v7l+                              i-sabre-q2m.dtbo
act-led.dtbo                               jedec-spi-nor.dtbo
adau1977-adc.dtbo                          justboom-both.dtbo
adau7002-simple.dtbo                       justboom-dac.dtbo
ads1015.dtbo                               justboom-digi.dtbo
ads1115.dtbo                               ltc294x.dtbo
ads7846.dtbo                               max98357a.dtbo
adv7282m.dtbo                              mbed-dac.dtbo
adv728x-m.dtbo                             mcp23017.dtbo
akkordion-iqdacplus.dtbo                   mcp23s17.dtbo
allo-boss-dac-pcm512x-audio.dtbo           mcp2515-can0.dtbo
allo-digione.dtbo                          mcp2515-can1.dtbo
allo-katana-dac-audio.dtbo                 mcp3008.dtbo
allo-piano-dac-pcm512x-audio.dtbo          mcp3202.dtbo
allo-piano-dac-plus-pcm512x-audio.dtbo     mcp342x.dtbo
anyspi.dtbo                                media-center.dtbo
apds9960.dtbo                              merus-amp.dtbo
applepi-dac.dtbo                           midi-uart0.dtbo
at86rf233.dtbo                             midi-uart1.dtbo
audioinjector-addons.dtbo                  miniuart-bt.dtbo
audioinjector-isolated-soundcard.dtbo      mmc.dtbo
audioinjector-ultra.dtbo                   mpu6050.dtbo
audioinjector-wm8731-audio.dtbo            mz61581.dtbo
audiosense-pi.dtbo                         ov5647.dtbo
audremap.dtbo                              papirus.dtbo
balena-fin.dtbo                            pi3-act-led.dtbo
bcm2708-rpi-b.dtb                          pi3-disable-bt.dtbo
bcm2708-rpi-b-plus.dtb                     pi3-disable-wifi.dtbo
bcm2708-rpi-cm.dtb                         pi3-miniuart-bt.dtbo
bcm2708-rpi-zero.dtb                       pibell.dtbo
bcm2708-rpi-zero-w.dtb                     piglow.dtbo
bcm2709-rpi-2-b.dtb                        piscreen2r.dtbo
bcm2710-rpi-2-b.dtb                        piscreen.dtbo
bcm2710-rpi-3-b.dtb                        pisound.dtbo
```

⋮

```
i2c-sensor.dtbo                            vc4-kms-kippah-7inch.dtbo
i2s-gpio28-31.dtbo                         vc4-kms-v3d.dtbo
ilitek251x.dtbo                            vga666.dtbo
imx219.dtbo                                w1-gpio.dtbo
iqaudio-codec.dtbo                         w1-gpio-pullup.dtbo
iqaudio-dac.dtbo                           w5500.dtbo
iqaudio-dacplus.dtbo                       wittypi.dtbo
iqaudio-digi-wm8804-audio.dtbo             zImage
irs1125.dtbo
```

- 커널 이미지와 모듈 복사를 완료하였다면 RPI로 해당 디렉터리를 전송한다.

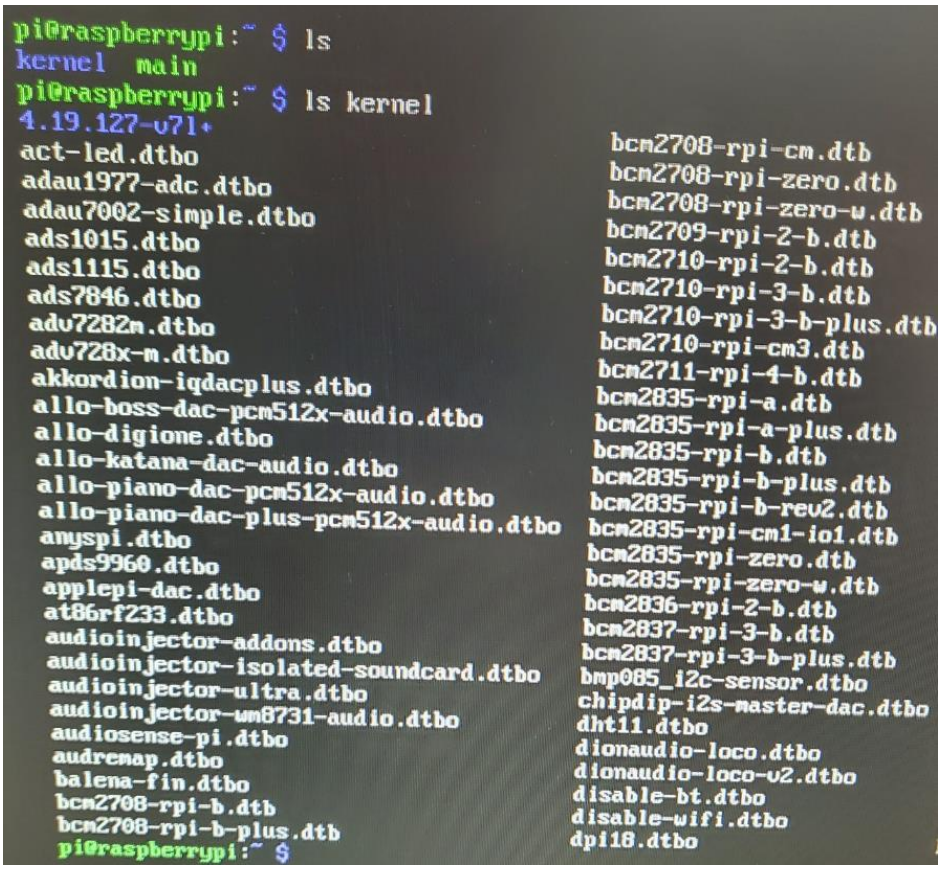
  $ scp -r ~/kernel pi@[RPI IP 주소]:/home/pi

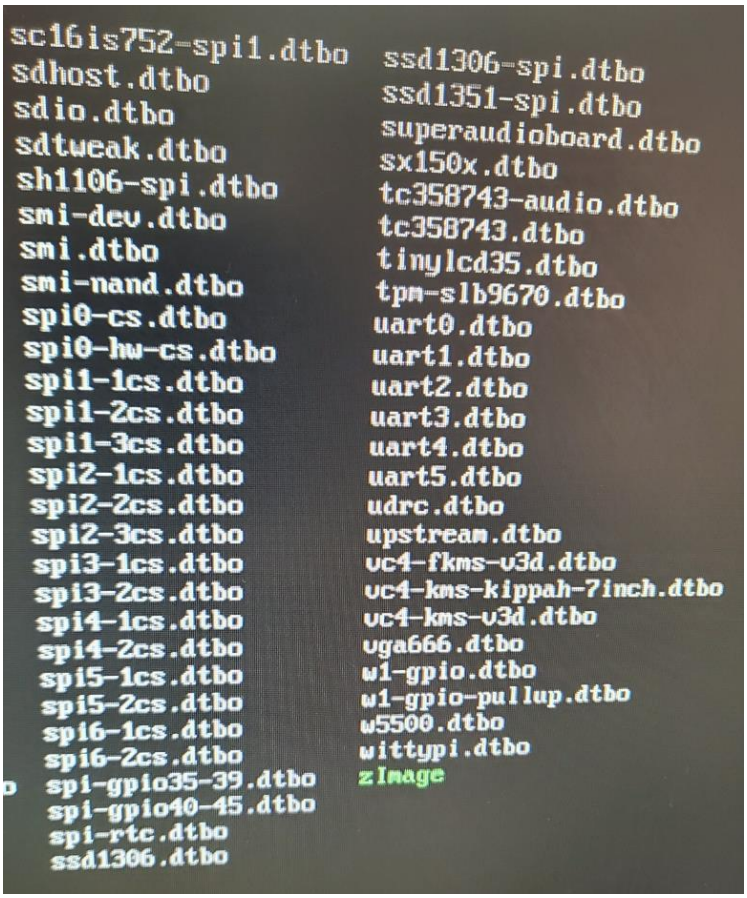
```
wlgns12www@VM:~$ scp -r ~/kernel pi@192.168.1.26:/home/pi
pi@192.168.1.26's password:
i-sabre-q2m.dtbo                              100%  893   497.9KB/s   00:00
pwm-2chan.dtbo                                100% 1096   236.7KB/s   00:00
imx219.dtbo                                   100% 3293   480.8KB/s   00:00
uart3.dtbo                                    100%  589   155.0KB/s   00:00
tc358743.dtbo                                 100% 2465   371.3KB/s   00:00
googlevoicehat-soundcard.dtbo                 100% 1259   235.2KB/s   00:00
spi5-1cs.dtbo                                 100% 1289   178.5KB/s   00:00
ov5647.dtbo                                   100% 2597   242.4KB/s   00:00
sdio.dtbo                                     100% 1889   232.5KB/s   00:00
```

⋮

```
vc4-kms-kippah-7inch.dtbo                     100% 1112   875.5KB/s   00:00
justboom-both.dtbo                            100% 1640     1.1MB/s   00:00
audiosense-pi.dtbo                            100% 2187     1.3MB/s   00:00
iqaudio-digi-wm8804-audio.dtbo                100% 1326   883.9KB/s   00:00
hifiberry-dac.dtbo                            100%  655   548.4KB/s   00:00
bcm2708-rpi-cm.dtb                            100%   23KB   3.5MB/s   00:00
```

- RPI에 정상적으로 커널 이미지와 모듈이 수신되었는지 확인한다.





***중요)** 이후부터는 RPI에서 진행되는 명령어이다.

- 전달받은 커널 관련 파일들을 다음과 같은 명령어들로 각 디렉터리로 이동 시킨다.

➢ 커널 이미지 이동

$ sudo mv ~/kernel/zImage /boot/kernel7l.img (**RPI 4 이상 모델, l(소문자 L)**)

$ sudo mv ~/kernel/zImage /boot/kernel7.img (**RPI 2 이상 모델**)

➢ Device tree blob 이동

$ sudo mv ~/kernel/*.dtb /boot/.

$ sudo mv ~/kernel/*.dtb* /boot/overlays/.

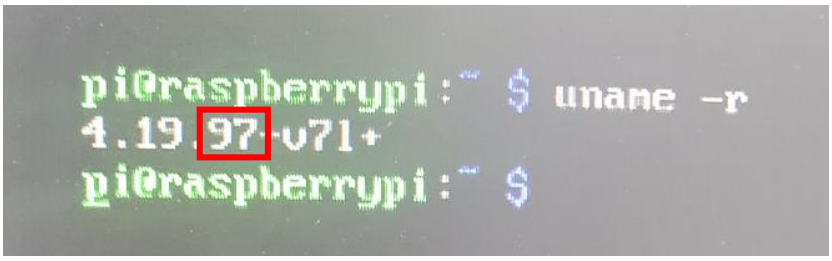
➢ 커널 모듈 디렉터리 이동

$ sudo mv ~/kernel/[kernel version]/ /lib/modules/.

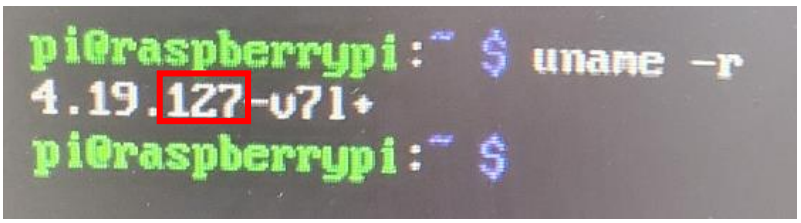
- *.dtb, *.dtb*, [kernel module directory], zImage를 모두 각자의 위치로 옮겼다면 리부팅을 진행한다.

  $ sudo reboot

  ➢ 리부팅 전 (아직 바뀐 커널로 변경되지 않음)



  ➢ 리부팅 후 (리부팅 전과 다른 커널 버전인 것을 확인가능)



- 본 실습에서는 RPI 툴체인을 활용해 RPI의 커널을 직접 컴파일 후 적용시켜 보았다.
  다음 시간에는 RPI에 시스템 콜을 올려보는 것을 진행한다.