Month 1, Day 2

Stealien 김도현

Contents

- Linux kernel versioning.
- Linux kernel compile.
- Linux kernel module compile.
- Make initramfs
- Qemu – Run kernel on virtual environment.
- Qemu – Debugging linux kernel.
- Sample prob

Linux kernel versioning

Ex) Linux **5.3.0**

Major revision
Version - (짝수: Stable, 홀수: Develop)
Minor revision - (패치 횟수)

Linux kernel compile

1) Get linux kernel source.
2) Configuration
3) Make
    1) Optional: install kernel to current machine

Linux kernel compile

Get linux kernel source

Name	Address
Git	https://git.kernel.org/
Github	https://github.com/torvalds/linux
Mirror	https://mirrors.edge.kernel.org/pub/

Linux kernel compile
Configuration

```
$ make help
…
Configuration targets:
  config          - Update current config utilising a line-oriented program
  nconfig         - Update current config utilising a ncurses menu based program
  menuconfig      - Update current config utilising a menu based program
…
```

Linux kernel compile

Demo

```
$ make menuconfig            # Kernel configuration.
$ make –j64                  # Make kernel with 64 threads.
…
$ ls arch/x86/boot/bzImage   # Kernel image places here.
bzImage
```

Linux kernel module compile

1) Write code
2) Write makefile
3) Make

Linux kernel module compile

Write code

```
/* my_module.c */

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

// Executes on module load
int my_init(void) {
    return 0;
}

// Executes on module exit
void my_cleanup(void) {
    return;
}

// Register init & exit
module_init(my_init);
module_exit(my_cleanup);
```

Linux kernel module compile

Write makefile **(tab != space)**

```
############
# Makefile #
############

# Which source? Ex) hello.c -> hello.o
obj-m := module.o

KERN := /mnt/hgfs/project/2020_probs/confidence2020/forgotten_module/debug/linux/linux-5.5
PWD  := $(shell pwd)

all:
	make -C $(KERN) M=$(PWD) modules
clean:
	make -C $(KERN) M=$(PWD) clean
```

Make initramfs

```
$ mkdir /tmp/initramfs
$ cd /tmp/initramfs
$ mkdir bin tmp proc dev
$ touch init
$ # Get busybox to bin
$ ./bin/busybox --install ./bin
$ find . | cpio -H newc -ov | gzip -cf > /tmp/initramfs.cpio.gz
```

Qemu

Run kernel on virtual environment.

```
#!/bin/bash

IMG="./bzImage"
RAM="./initramfs.cpio.gz"

qemu-system-x86_64              \
    -kernel $IMG                \
    -initrd $RAM                \
    -enable-kvm                 \
    -m 64m                      \
    -nographic                  \
    -monitor /dev/null          \
    -append 'console=ttyS0' -s
```

Any questions?