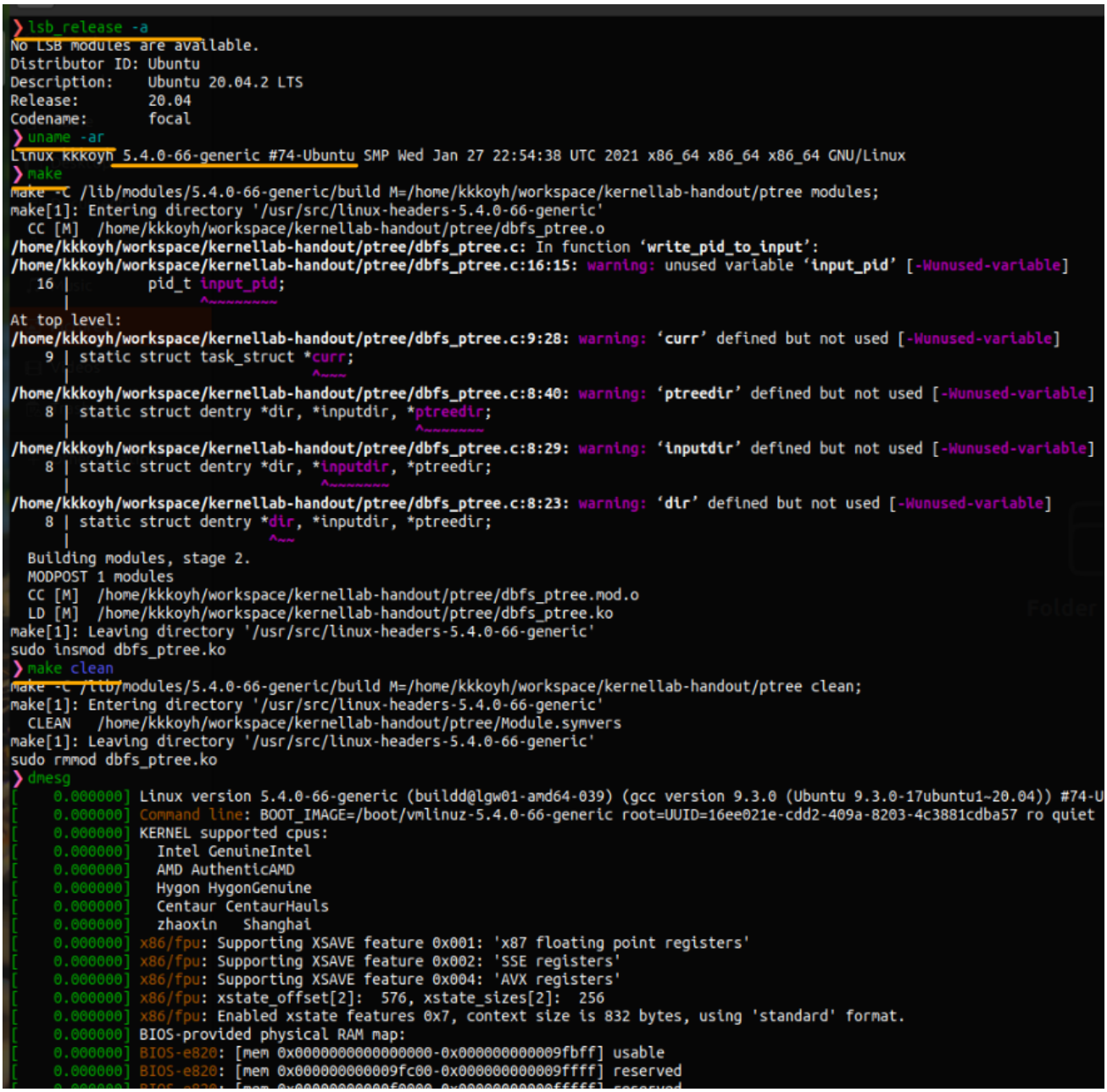
2021 SYSTEM PROGRAMMING

Lab5 Report – Kernel Lab

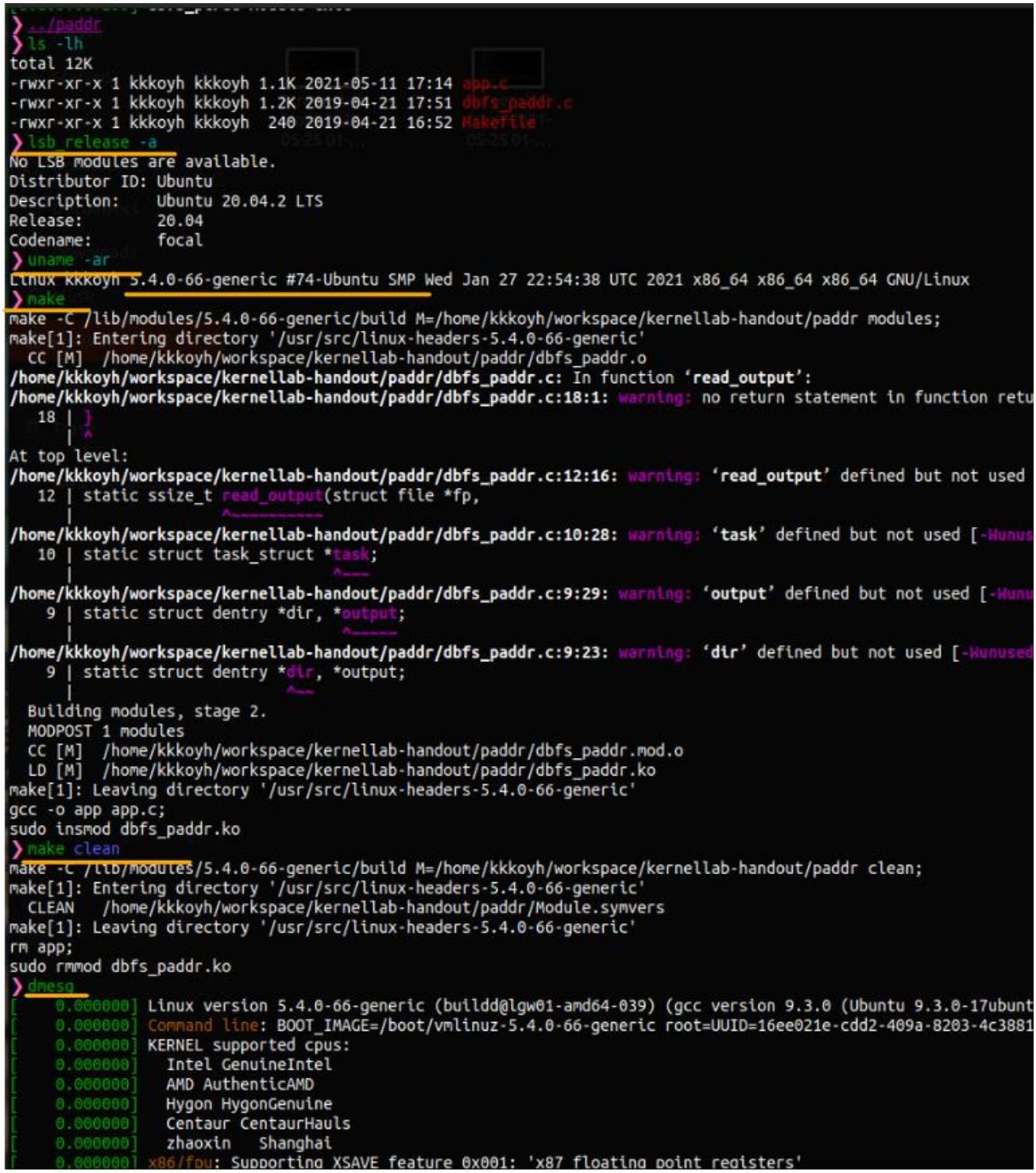
자유전공학부 2012-13311 안 효 지

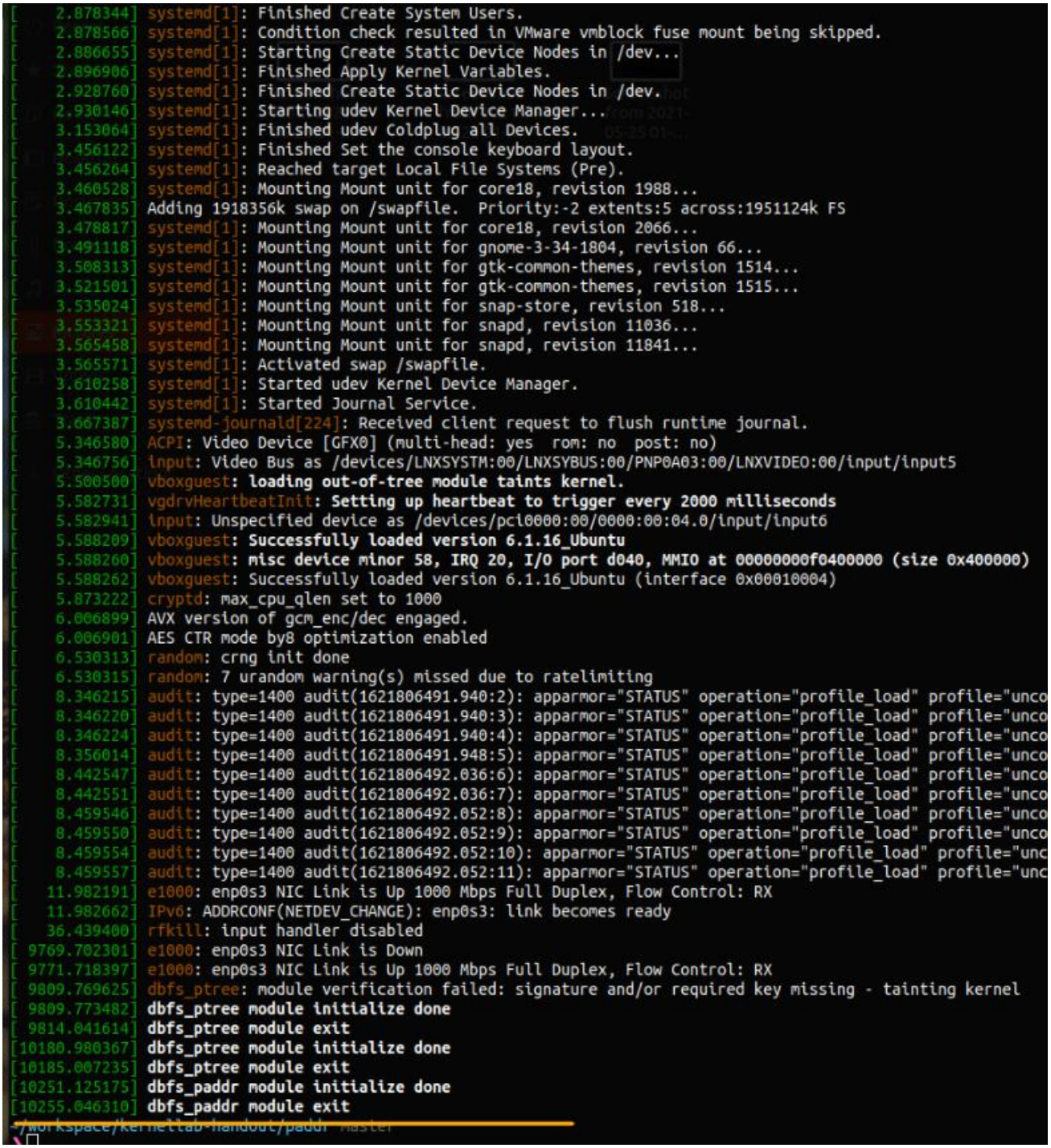
<Preparing screen shot>



```
❯ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal
❯ uname -ar
Linux kkkoyh 5.4.0-66-generic #74-Ubuntu SMP Wed Jan 27 22:54:38 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
❯ make
make -C /lib/modules/5.4.0-66-generic/build M=/home/kkkoyh/workspace/kernellab-handout/ptree modules;
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-66-generic'
  CC [M]  /home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.o
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c: In function ‘write_pid_to_input’:
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c:16:15: warning: unused variable ‘input_pid’ [-Wunused-variable]
   16 |         pid_t input_pid;
      |               ^~~~~~~~~
At top level:
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c:9:28: warning: ‘curr’ defined but not used [-Wunused-variable]
    9 | static struct task_struct *curr;
      |                            ^~~~
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c:8:40: warning: ‘ptreedir’ defined but not used [-Wunused-variable]
    8 | static struct dentry *dir, *inputdir, *ptreedir;
      |                                        ^~~~~~~~
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c:8:29: warning: ‘inputdir’ defined but not used [-Wunused-variable]
    8 | static struct dentry *dir, *inputdir, *ptreedir;
      |                             ^~~~~~~~
/home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.c:8:23: warning: ‘dir’ defined but not used [-Wunused-variable]
    8 | static struct dentry *dir, *inputdir, *ptreedir;
      |                       ^~~
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.mod.o
  LD [M]  /home/kkkoyh/workspace/kernellab-handout/ptree/dbfs_ptree.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-66-generic'
sudo insmod dbfs_ptree.ko
❯ make clean
make -C /lib/modules/5.4.0-66-generic/build M=/home/kkkoyh/workspace/kernellab-handout/ptree clean;
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-66-generic'
  CLEAN   /home/kkkoyh/workspace/kernellab-handout/ptree/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-66-generic'
sudo rmmod dbfs_ptree.ko
❯ dmesg
[    0.000000] Linux version 5.4.0-66-generic (buildd@lgw01-amd64-039) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #74-U
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-66-generic root=UUID=16ee021e-cdd2-409a-8203-4c3881cdba57 ro quiet
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin   Shanghai
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
[    0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
```

```
[    2.858762] systemd[1]: Mounted FUSE Control File System.
[    2.867275] systemd[1]: Mounted Kernel Configuration File System.
[    2.878344] systemd[1]: Finished Create System Users.
[    2.878566] systemd[1]: Condition check resulted in VMware vmblock fuse mount being skipped.
[    2.886655] systemd[1]: Starting Create Static Device Nodes in /dev...
[    2.896906] systemd[1]: Finished Apply Kernel Variables.
[    2.928760] systemd[1]: Finished Create Static Device Nodes in /dev.
[    2.930146] systemd[1]: Starting udev Kernel Device Manager...
[    3.153064] systemd[1]: Finished udev Coldplug all Devices.
[    3.456122] systemd[1]: Finished Set the console keyboard layout.
[    3.456264] systemd[1]: Reached target Local File Systems (Pre).
[    3.460528] systemd[1]: Mounting Mount unit for core18, revision 1988...
[    3.467835] Adding 1918356k swap on /swapfile.  Priority:-2 extents:5 across:1951124k FS
[    3.478817] systemd[1]: Mounting Mount unit for core18, revision 2066...
[    3.491118] systemd[1]: Mounting Mount unit for gnome-3-34-1804, revision 66...
[    3.508313] systemd[1]: Mounting Mount unit for gtk-common-themes, revision 1514...
[    3.521501] systemd[1]: Mounting Mount unit for gtk-common-themes, revision 1515...
[    3.535024] systemd[1]: Mounting Mount unit for snap-store, revision 518...
[    3.553321] systemd[1]: Mounting Mount unit for snapd, revision 11036...
[    3.565458] systemd[1]: Mounting Mount unit for snapd, revision 11841...
[    3.565571] systemd[1]: Activated swap /swapfile.
[    3.610258] systemd[1]: Started udev Kernel Device Manager.
[    3.610442] systemd[1]: Started Journal Service.
[    3.667387] systemd-journald[224]: Received client request to flush runtime journal.
[    5.346580] ACPI: Video Device [GFX0] (multi-head: yes  rom: no  post: no)
[    5.346756] input: Video Bus as /devices/LNXSYSTM:00/LNXSYBUS:00/PNP0A03:00/LNXVIDEO:00/input/input5
[    5.500500] vboxguest: loading out-of-tree module taints kernel.
[    5.582731] vgdrvHeartbeatInit: Setting up heartbeat to trigger every 2000 milliseconds
[    5.582941] input: Unspecified device as /devices/pci0000:00/0000:00:04.0/input/input6
[    5.588209] vboxguest: Successfully loaded version 6.1.16_Ubuntu
[    5.588260] vboxguest: misc device minor 58, IRQ 20, I/O port d040, MMIO at 00000000f0400000 (size 0x400000)
[    5.588262] vboxguest: Successfully loaded version 6.1.16_Ubuntu (interface 0x00010004)
[    5.873222] cryptd: max_cpu_qlen set to 1000
[    6.006899] AVX version of gcm_enc/dec engaged.
[    6.006901] AES CTR mode by8 optimization enabled
[    6.530313] random: crng init done
[    6.530315] random: 7 urandom warning(s) missed due to ratelimiting
[    8.346215] audit: type=1400 audit(1621806491.940:2): apparmor="STATUS" operation="profile_load" profile="un
[    8.346220] audit: type=1400 audit(1621806491.940:3): apparmor="STATUS" operation="profile_load" profile="un
[    8.346224] audit: type=1400 audit(1621806491.940:4): apparmor="STATUS" operation="profile_load" profile="un
[    8.356014] audit: type=1400 audit(1621806491.948:5): apparmor="STATUS" operation="profile_load" profile="un
[    8.442547] audit: type=1400 audit(1621806492.036:6): apparmor="STATUS" operation="profile_load" profile="un
[    8.442551] audit: type=1400 audit(1621806492.036:7): apparmor="STATUS" operation="profile_load" profile="un
[    8.459546] audit: type=1400 audit(1621806492.052:8): apparmor="STATUS" operation="profile_load" profile="un
[    8.459550] audit: type=1400 audit(1621806492.052:9): apparmor="STATUS" operation="profile_load" profile="un
[    8.459554] audit: type=1400 audit(1621806492.052:10): apparmor="STATUS" operation="profile_load" profile="u
[    8.459557] audit: type=1400 audit(1621806492.052:11): apparmor="STATUS" operation="profile_load" profile="u
[   11.982191] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[   11.982662] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[   36.439400] rfkill: input handler disabled
[ 9769.702301] e1000: enp0s3 NIC Link is Down
[ 9771.718397] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 9809.769625] dbfs_ptree: module verification failed: signature and/or required key missing - tainting kernel
[ 9809.773482] dbfs_ptree module initialize done
[ 9814.041614] dbfs_ptree module exit
[10180.980367] dbfs_ptree module initialize done
[10185.007235] dbfs_ptree module exit
~/workspace/kernellab-handout/ptree master
❯
```


../paddr
ls -lh
total 12K
-rwxr-xr-x 1 kkkoyh kkkoyh 1.1K 2021-05-11 17:14 app.c
-rwxr-xr-x 1 kkkoyh kkkoyh 1.2K 2019-04-21 17:51 dbfs_paddr.c
-rwxr-xr-x 1 kkkoyh kkkoyh 240 2019-04-21 16:52 Makefile
lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 20.04.2 LTS
Release: 20.04
Codename: focal
uname -ar
Linux kkkoyh 5.4.0-66-generic #74-Ubuntu SMP Wed Jan 27 22:54:38 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
make
make -C /lib/modules/5.4.0-66-generic/build M=/home/kkkoyh/workspace/kernellab-handout/paddr modules;
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-66-generic'
CC [M] /home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.o
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c: In function 'read_output':
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c:18:1: warning: no return statement in function retu
18 | }
| ^
At top level:
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c:12:16: warning: 'read_output' defined but not used
12 | static ssize_t read_output(struct file *fp,
| ^~~~~~~~~~~
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c:10:28: warning: 'task' defined but not used [-Wunus
10 | static struct task_struct *task;
| ^~~~
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c:9:29: warning: 'output' defined but not used [-Wunu
9 | static struct dentry *dir, *output;
| ^~~~~~
/home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.c:9:23: warning: 'dir' defined but not used [-Wunused
9 | static struct dentry *dir, *output;
| ^~~
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.mod.o
LD [M] /home/kkkoyh/workspace/kernellab-handout/paddr/dbfs_paddr.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-66-generic'
gcc -o app app.c;
sudo insmod dbfs_paddr.ko
make clean
make -C /lib/modules/5.4.0-66-generic/build M=/home/kkkoyh/workspace/kernellab-handout/paddr clean;
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-66-generic'
CLEAN /home/kkkoyh/workspace/kernellab-handout/paddr/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-66-generic'
rm app;
sudo rmmod dbfs_paddr.ko
dmesg
[0.000000] Linux version 5.4.0-66-generic (buildd@lgw01-amd64-039) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubunt
[0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-66-generic root=UUID=16ee021e-cdd2-409a-8203-4c3881
[0.000000] KERNEL supported cpus:
[0.000000] Intel GenuineIntel
[0.000000] AMD AuthenticAMD
[0.000000] Hygon HygonGenuine
[0.000000] Centaur CentaurHauls
[0.000000] zhaoxin Shanghai
[0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'



0. Goal of this lab

Loadable kernel module을 통해 커널에 접근하는 모듈을 만들 수 있다, debug file system을 이용하여 실제 kernel information에 접근해보는 경험을 할 수 있다.

1. How to implement

<Part A-ptree.c>

1) __init dbfs_module_init(void) : `make`를 하면 실행되는 init module

(1) dir = debugfs_create_dir(“ptree”, NULL)

“ptree”라는 dir을 만들라는 의미이다. /sys/kernel/debug에 위치한다. 디렉토리 생성에 실패한 경우 0이 리턴된다.

(2) inputfile = debugfs_create_file(“input”, 00700, dir, NULL, &dbfs_fops_write)

dir 안에 “input”이라는 파일을 생성한다. 00700은 파일접근권한자중 하나인데, owner가 read, write, execute 할 수 있다는 의미이다. Dbfs_fops_write에 등록된 function들을 사용하여 해당 작업을 수행하라는 의미이다. 이 inputfile은 아래에서 설명할 write_pid_to_input()에 따라 pid를 input으로 받고, init(1)까지 tracing한 자료를 inputfile에 넣는다.

(3) ptreefile = debugfs_create_file(“ptree”, 00700, dir, NULL, &dbfs_fops_read)

“ptree”라는 파일을 dir에 만들고, (2)와 동일한 권한을 owner에게 주며, dbfs_fops_read에 등록된 함수인 read_output_from_ptreefile()에 따른 역할을 수행한다. Write_pid_to_input()에서 user 버퍼에 담아놓았던 자료들을 커널버퍼로 읽어들이는 작업을 수행한다.

2) __exit dbfs_module_exit(void): `make clean`을 하면 실행되는 모듈

debugfs_remove_recursive(dir)을 통해 만들어놓았던 dir과 그 하위 파일들을 삭제한다.

3) write_pid_to_input()

(1) User buffer로부터 받은 값을 input_pid에 저장하고, 해당 pid를 가진 task structure을 찾는다. 그리고 재귀적으로 root를 찾아가는 trace_process()를 따로 만들어서 user buffer에 담도록 한다.

(2) trace_process(task_struct* curr)

Struct pointer인 curr을 사용하여 parent process의 정보들을 tracing할 수 있다. Curr->comm을 통해 프로세스의 이름을 최대 15byte까지 받을 수 있으며, curr->pid로 해당 프로세스의 pid를 알 수 있다. 원하는 formation으로 버퍼에 넣기 위해 snprintf()를 사용하였으며, root까지 추적하기 위해 recursive function으로 구성했다. MAXLEN은 1000으로 하였는데 1024로 하면 stack overflow가 일어났기 때문에 그보다 작은 수로 결정하였다.
output buffer에 init(1)부터 담겨야하기 때문에 recursive가 다 끝난 후에 결과물들이 stack에서 빠져나오면서 buffer에 담기도록 코드 순서를 구성하였다. 굳이 번거롭게 most child를 가장 먼저 buffer의 맨 끝에 담는 것을 지양하기 위해서이다.

<Part B – paddr.c>

1) module_init / module_exit : part A와 동일

2) read_output(struct file* fp, char __user* user_buf, size_t length, loff_t* position)

입력된 pid와 vaddr을 통해 paddr을 찾아 user buffer로 보내는 역할까지 수행한다.

(1) app.c 살피기

Argument에 length가 있는데 이게 과연 어디서 들어오는지 찾다가 app.c를 살펴보게 되었다. App.c에는 packet structure가 존재했고, 그것이 버퍼를 통해 커널로 들어오고, 또 packet 사이즈만큼을 읽어서 결과를 비교한다는 것을 알게 되었다. 그래서 dbfs_paddr.c에도 동일한 사이즈와 element의 structure를 만들어 거기에 결과가 담기도록 하였다.

(2) pid와 vaddr 추출하기

User_buf를 통해 입력이 들어온다. 이것을 copy_from_user를 통해 커널버퍼로 옮긴다. 커널버퍼는 kzalloc을 통해 처음부터 0으로 초기화시켰으며, length만큼 할당하였다. 나중에 kfree를 해주는 것을 잊지 않아야 한다.

옮긴 후에는 pid와 vaddr를 추출한다. 이를 위해 일단 length 값을 printk하여 보았더니 24가 나왔다. 그래서 struct packet의 원소들인 pid, vaddr, paddr의 사이즈를 찍어보았더니 4, 8, 8이 나왔다. 더하면 24가 아니라서 찾아보았더니, pid의 뒤에는 8 byte alignment를 맞추기 위해 4byte의 padding이 추가되어 그렇다는 것이었다. 그리하여 24byte의 structure를 buffer를 통해 받았고, 4byte의 pid가 little endian으로 저장되어 있으므로 [0]~[3]의 정보를 bitwise operator를 사용해 추출하였다. 그리고 vaddr은 [8]~[15]에 저장되어 있지만 실제로는 48bit이므로 [8]~[13]을 역시 bitwise operator를 이용해 little endian을 해석하여 추출하였다. 추출한 정보들은 paddr.c에 만들어놓았다고 언급한 packet structure의 element 값에 넣어주었다.

그리고 pid_task(find_get_pid())를 통해 pid의 정보를 갖고 있는 structure를 받았다. Part A에서는 find_vpid를 사용하였는데, 검색해보니 find_get_pid와 동일하나 후자는 ref count가 1식 증가한다는 차이점만 있어서 둘 중 아무거나 사용하였다.

(3) paddr 찾기

Pgd -> p4d -> pud -> pmd -> pte 순으로 접근하여 결국엔 Page Frame Number를 받을 수 있었다. Offset이 LSB 3bit라 맨 마지막이 000이겠지만, 그래도 VPO와 PPO가 같은 것을 assert하기 위하여 paddr = PFN << 12 ¦ (vaddr&0xfff) 를 해주었다.

(4) user buffer로 보내기

Packet structure에 모든 정보를 담아놓았으니 그것을 simple_read_from_buffer를 통해 user buffer로 보낸다. 보내기 전에 kfree를 해준다.

2. What was difficult

커널에 대해 수업시간에 배운 적이 없던 것 같은데 갑자기 커널이 나와서 당황했다. 구글링을 통해 어찌저찌 해결하기는 했으나, 수업시간에 관련 개념을 다룬 후 수행했던 이전 랩들보다는 지식이 덜 정리된 느낌이다. 아직도 커널의 개념이 두루뭉술하다.

특히 dmesg만을 이용해 디버깅하는 게 좀 힘들었다. Printk()를 사용해서 하는 방법만 사용했는데 또 다른 방법이 있는지 모르겠다.

3. What was surprising

1) vaddr->paddr 로 바꾸기 위해서 하나하나 주소를 다 변환해주어야 하는 줄 알았는데 pgd, p4d … pte 등등 유용한 함수들이 있다는 것을 알았고, 그 덕에 과제 수행이 편했다.

2) file_operations dbfs_fops 와 같은 struct는 파일별로 하나씩 존재해야한다는 것을 처음 알았다. 그냥 read, write를 등록하면 서로 다른 debugfs_create_file에서 read따로, write따로 갖다 쓸 수 있는 줄 알았는데 그게 아니었다.

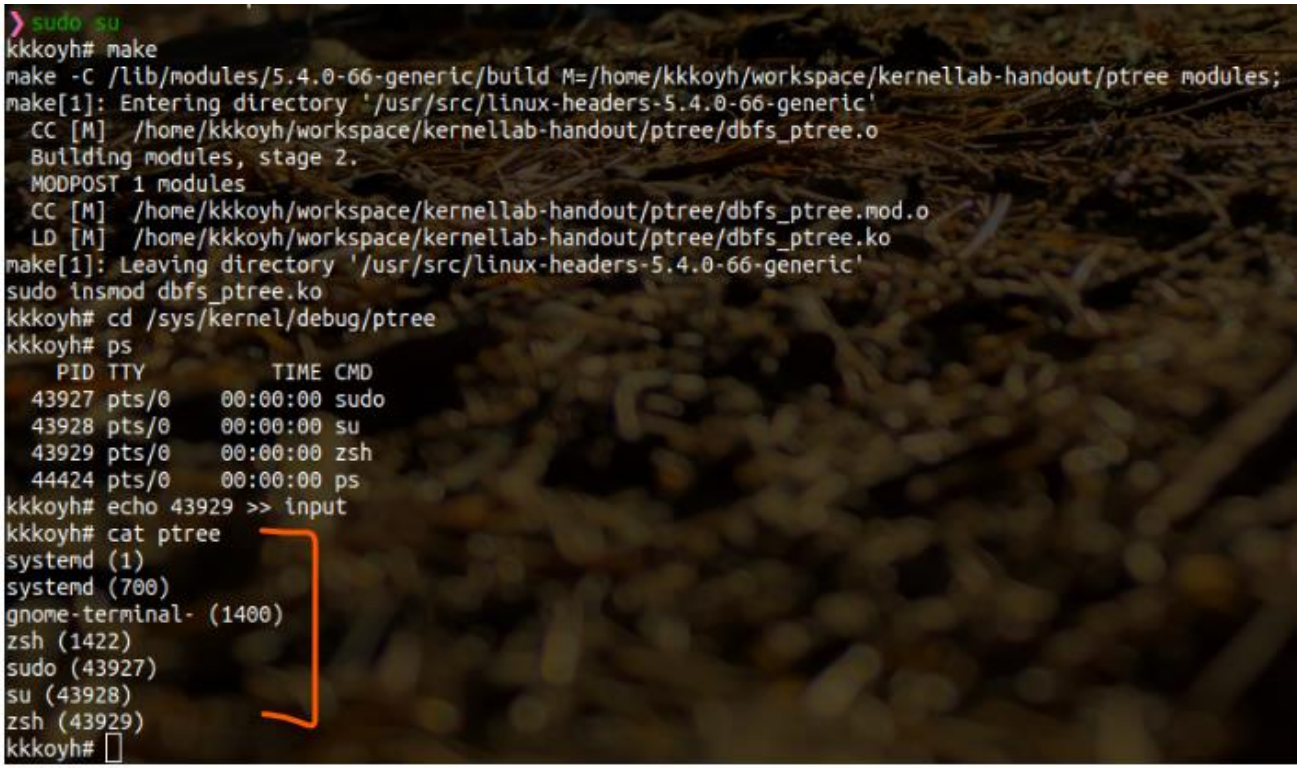
3) kmalloc, kzalloc 과 같이 커널용 함수가 따로 있다는 것을 처음 알았다.

4) 커널에선 C99와 같은 것을 지원하지 않아서 body의 맨 위에 선언을 먼저 해주어야 한다는 것을 처음 알았다.

5) struct의 size는 원소들의 size의 합과 같지 않을 수도 있다는 것을 알았다.

4. Result screen shot

<Part1. Ptree>



<Part 2. Paddr>