

1. 실행 결과

1) part 1

```

healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/ptree$ make
make -C /lib/modules/5.13.0-44-generic/build M=/home/healthykim/Desktop/kernellab-handout/ptree modules;
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-44-generic'
  CC [M] /home/healthykim/Desktop/kernellab-handout/ptree/dbfs_ptree.o
  MODPOST /home/healthykim/Desktop/kernellab-handout/ptree/Module.symvers
  CC [M] /home/healthykim/Desktop/kernellab-handout/ptree/dbfs_ptree.mod.o
  LD [M] /home/healthykim/Desktop/kernellab-handout/ptree/dbfs_ptree.ko
  BTF [M] /home/healthykim/Desktop/kernellab-handout/ptree/dbfs_ptree.ko
Skipping BTF generation for /home/healthykim/Desktop/kernellab-handout/ptree/dbfs_ptree.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-44-generic'
sudo insmod dbfs_ptree.ko
healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/ptree$ sudo su
root@healthykim-VirtualBox:/home/healthykim/Desktop/kernellab-handout/ptree# cd /sys/kernel/debug/ptree/
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# echo $$
5586
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# echo 5586 > input
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# cat ptree
systemd (1)
systemd (709)
gnome-terminal- (1816)
bash (1824)
sudo (5584)
su (5585)
bash (5586)
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# ps
  PID TTY          TIME CMD
  5584 pts/0      00:00:00 sudo
  5585 pts/0      00:00:00 su
  5586 pts/0      00:00:00 bash
  7641 pts/0      00:00:00 ps
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# echo 5584 > input
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# cat ptree
systemd (1)
systemd (709)
gnome-terminal- (1816)
bash (1824)
sudo (5584)
root@healthykim-VirtualBox:/sys/kernel/debug/ptree# exit
exit
healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/ptree$ make clean
make -C /lib/modules/5.13.0-44-generic/build M=/home/healthykim/Desktop/kernellab-handout/ptree clean;
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-44-generic'
  CLEAN /home/healthykim/Desktop/kernellab-handout/ptree/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-44-generic'
sudo rmmod dbfs_ptree.ko

```

2) part2

```
healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/paddr$ make
make -C /lib/modules/5.13.0-44-generic/build M=/home/healthykim/Desktop/kernellab-handout/paddr modules;
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-44-generic'
CC [M] /home/healthykim/Desktop/kernellab-handout/paddr/dbfs_paddr.o
MODPOST /home/healthykim/Desktop/kernellab-handout/paddr/Module.symvers
CC [M] /home/healthykim/Desktop/kernellab-handout/paddr/dbfs_paddr.mod.o
LD [M] /home/healthykim/Desktop/kernellab-handout/paddr/dbfs_paddr.ko
BTF [M] /home/healthykim/Desktop/kernellab-handout/paddr/dbfs_paddr.ko
Skipping BTF generation for /home/healthykim/Desktop/kernellab-handout/paddr/dbfs_paddr.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-44-generic'
gcc -o app app.c;
sudo insmod dbfs_paddr.ko
[sudo] password for healthykim:
healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/paddr$ sudo ./app
[TEST CASE] PASS
healthykim@healthykim-VirtualBox:~/Desktop/kernellab-handout/paddr$ make clean
make -C /lib/modules/5.13.0-44-generic/build M=/home/healthykim/Desktop/kernellab-handout/paddr clean;
make[1]: Entering directory '/usr/src/linux-headers-5.13.0-44-generic'
CLEAN /home/healthykim/Desktop/kernellab-handout/paddr/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-5.13.0-44-generic'
rm app;
sudo rmmod dbfs_paddr.ko
```

2. 구현 방법

1) Part 1

Part 1에서는 pid tree를 출력하는 기능을 구현했다. 이를 위해 먼저 지금의 pid를 string에 담은 다음 해당 pid의 프로세스를 받아오고, 그로부터 프로세스의 parent를 따라 올라가며 각 node process의 pid를 string의 앞에 붙이는 방식으로 구현했다. string에 pid들을 담는 과정이 끝나면, 미리 정의해 둔 debugfs_blob_wrapper에 string 정보를 담고, 이를 ptree blob에 출력하는 방식으로 구현했다.

2) Part 2

Part 2에서는 virtual address를 이용하여 page table에서 physical address를 찾아가는 것을 구현했다. 먼저 page offset를 저장해 두고, pid를 활용하여 프로세스를 찾아서 page directory를 찾아가는 과정이다. page directory에서 virtual address의 9비트씩을 활용해서 다음 page table을 계속 찾아가는 것이다. 그러다 페이지 테이블 엔트리를 찾으면, 해당 엔트리로 page number를 찾고, page offset을 붙여서 physical address를 구한다. physical address를 구하고 나면 패킷에 physical address를 기록한다.

3. 어려웠던 점

먼저 실습을 시작하면서 환경 문제가 있었다. 현재 듀얼부팅으로 우분투 20.04 LTS, 커널 버전 5.13.0-44-generic을 사용하고 있는데 이유를 모르겠어서 에러가 발생하여 동일한 버전의 virtual box로 옮겨서 과제를 진행해야 했다.

다음으로 part 2에서 문제가 있었다. asm/pgtable.h에 정의되어 있다고 하는 'pgd_offset()'함수를 호출하려고 했는데, 계속 이 함수가 없다는 에러가 났다. 모든 'xxx_offset~'함수에서 같은 에러가 발생했기 때문에 이 부분은 함수를 활용하지 못했고 일일이 address를 읽어 가며 다음 페이지를 찾아 가도록 구현했다.

4. 새롭게 배운 점

아무래도 커널 프로그래밍에 사용되는 자료구조와 여러 API들을 경험해 볼 수 있었던 것 같다. 그리고 Part 1을 하는 도중 vim으로 텍스트 에디터를 열어 봤을 때 처음 보는 문자 '^@'가 떠서 당황했는데, 이것이 null 문자라는 것을 알게 되었다. 처음에 size을 크게 잡아 data wrapper에 저장했더니 이런 문제가 발생했고 이를 방지하기 위해 필요한 만큼만 size로 저장하는 방식으로 바꾸었더니 문제를 해결할 수 있었다.