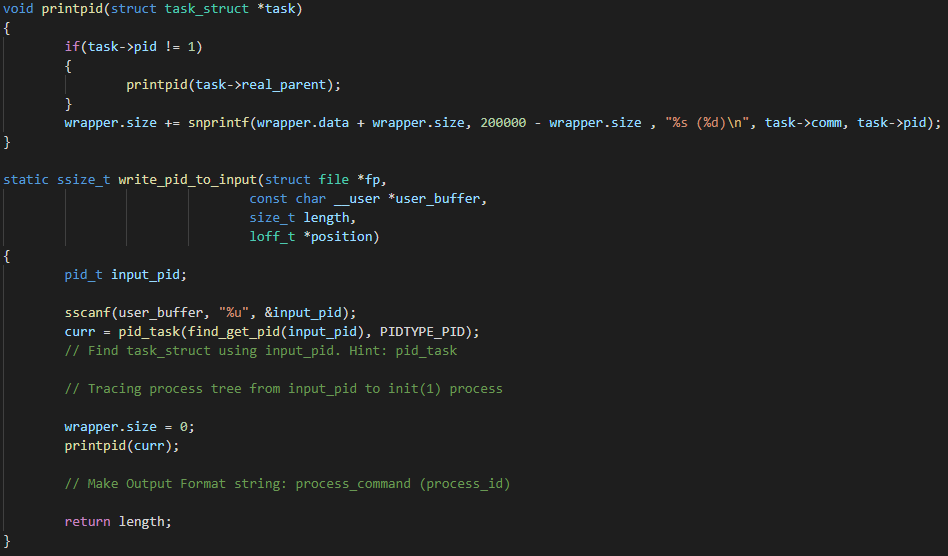
**시스템 프로그래밍 kernel lab 보고서**

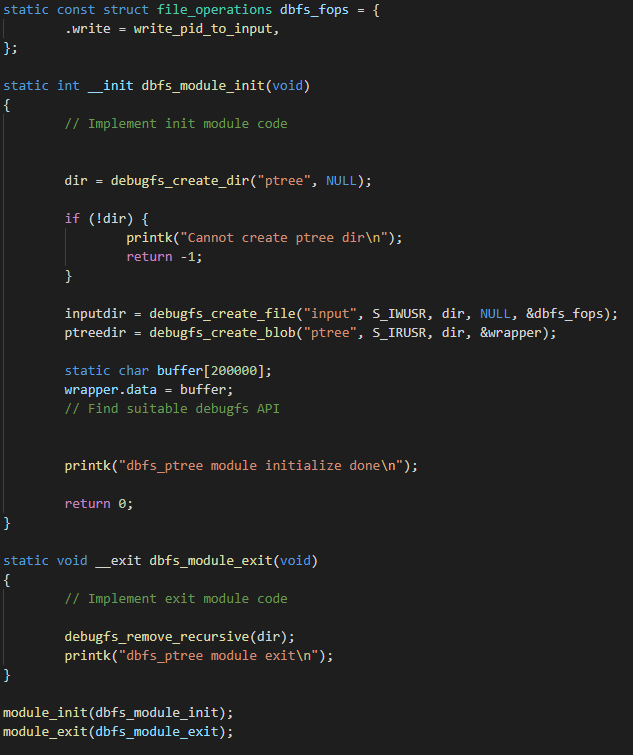
2017-12751 컴퓨터공학부 이동학

이번 lab에서는 process id로부터 process tree를 tracing 하고 virtual address를 이용하여 physical address를 찾는 코드를 구현하면서 debug file system interface를 이용한 linux kernel module programming을 이해하는 것을 목표로 하였습니다

1. ptree.c



process의 pid를 출력하는 함수로, real\_parent와 comm을 이용해 pid가 1인 프로세스까지 올라가는 재귀함수를 구현하였습니다. 재귀가 끝나면 역으로 추적하면서 출력 버퍼에 포맷 스트링을 담아주었고, 이를 출력하기 위해 wrapper를 사용하였습니다.



file write를 하면 write\_pid\_to\_input을 실행하게 하고 input과 ptree 파일 생성을 위해 debugfs\_create\_dir을 사용하였습니다. input의 경우 pid만 써야하므로 S\_IWUSR모드를 사용하고 dbfs\_fops 구조체를 넣어주었습니다. blob의 경우 output 파일을 작성할 버퍼를 만들어 이를 넣어주었습니다. 마지막으로 모듈을 제거할 때 만든 디렉토리를 삭제합니다.

2. paddr.c



mmap fail이 발생해서 paddr을 0x252345000으로 바꿔주었고, app.c에서 packet 구조체를 넘겨받으므로 같은 구조체를 정의하여 사용하였습니다. x86-64 환경이므로 48bit의 virtual address를 physical address로 전환하였고, 4kb page table을 사용하므로 하위 12bit를 page offset으로 사용하였습니다. 위 함수에서는 bit mask를 사용하여 각 level의 page table마다 offset을 계산해주고 mapping을 할 때 page offset만큼 전환해주었습니다.

후기: 커널을 직접 다루는 것은 신기하고 재미있는 경험이었지만, low level programming인 만큼 참고할 자료가 많지 않고 debug file system interface나 kernel module의 task\_struct등이 직관적으로 다가오지 않아서 많이 어려웠습니다. 하지만 그 덕분에 kernel level programming과 user level programming의 차이를 배우고 kernel data structure의 구조와 메커니즘을 익힐 수 있었습니다.