

# REPORT



## 운영체제

Operating System

Assignment 4

김태석 교수님

컴퓨터정보공학부

2017202037 오민혁

2020/11/29 (일)



## < Introduction >

이번 과제는 하나의 프로세스에 대한 가상 메모리 주소, 프로세스의 데이터 영역, 코드 영역, 힙 영역 주소, 그리고 원본 파일의 전체 경로를 출력하는 것이다.

이를 하기 위해서는 프로세스의 가상 메모리 관리를 위한 자료 구조인 `struct task_struct`에 대해서 이해하고 다룰 줄 알아야 한다. 또한 원하는 정보를 출력하기 위해 `task_struct`안의 `mm_struct`안의 `vm_area_struct`의 변수들에 대해 파악해야 한다.

## < Conclusion >

<https://stackoverflow.com/questions/33249643/how-to-use-dentry-path-raw>

/ `dentry_path_raw` 함수

## < Conclusion >

### - Analysis

System Call wrapping 을 위해 `file_varea.c` 를 5주차 실습 강의자료를 참고하여 작성하였다. 기존의 `add` 시스템 콜을 wrapping 하였다. 기존의 `add` 시스템 콜 주소를 저장할 포인터인 `syscall_table`을 전역 변수로 설정하고, 기존의 `add` 시스템 콜을 대체 할 `file_varea` 시스템 콜을 정의하는 `_SYSCALL_DEFINEx`를 작성하였다. 이 안에서 프로세스에 대한 정보들을 출력하는 코드를 작성 하였는데 가상 메모리 주소를 출력하기 위해서 우선 `pid_task` 함수를 이용해 현재 프로세스의 정보를 담고 있는 `task_struct` 구조체를 가리키는 포인터를 설정하였다.

그 다음 이 프로세스에 대한 가상 메모리 영역의 정보를 담고있는 구조체인 `vm_area_struct` 를 가리키는 포인터를 설정하였다. 이는 `task_struct` 안에 있는 `mm_struct *mm` 안의 `vm_area_struct * mmap` 을 가리키는 순서로 진행했다.

정보가 위치하는 가상 메모리 주소는 `vm_area_struct` 의 `vm_start`, `vm_end` 를 이용해 출력하였고, 프로세스의 코드 영역 주소는 `mm_struct`의 `start_code`, `end_code`, 데이터 영역 주소는 `mm_struct`의 `start_data`, `end_data`, 그 다음 heap 영역의 주소는 `mm_struct`의 `start_brk`와 `brk`를 이용하여 출력했다. 마지막으로 원본 파일의 전체 경로는 `dentry_path_raw` 함수를 이용하여 출력하였다.

## - 고찰

이번 과제를 하면서 가장 어려웠던 점은 `task_struct`, `mm_struct`, `vm_area_struct`의 자료 구조를 파악하는 점이였다. 실습 참고자료를 몇번이고 다시 보며 손으로 직접 그려보기도 하며 이해 할 수 있었다. 또한 원본 파일의 전체 경로를 출력하는 점에서 굉장히 난감했다. 구글 서치를 해도 정보가 잘 검색되지 않아 힘들었다. 하지만 경로에 대한 정보를 출력해주는 함수가 무조건 있을 거란 의지를 가지고 영문으로도 검색한 결과 `dentry_path_raw` 함수를 찾을 수 있었다.