

# 운영체제 Mysh2 보고서

201720720 조민재

이번 과제는 리눅스로 나만의 셸을 구현하는 과제였다.

먼저 커맨드를 파싱하는 함수를 구현했는데, 이게 가장 생각하기 어렵고 시간이 많이 걸렸다. 내가 생각한 방법은 공백과 ""를 모두 null로 바꿔준 후 null단위로 끊어서 argv 배열에 넣어주는 방법이었다. ""안의 공백은 null로 바꿔주면 안되므로, 먼저 flag를 0으로 설정해서 XOR 을 사용하여 ""를 만나면 조건문으로 "" 사이의 공백은 null로 바뀌지 않게 하였다. 이렇게 스트링을 null단위로 끊어 주었으면, 반복문을 사용해 스트링의 포인터값과 인덱스를 옮겨주며 argv에 메모리 할당을 하고 strcpy로 스트링을 복사하였다. 이렇게 해주면 null단위로 끊어진 스트링이 차례대로 argv배열에 들어가게 된다. 그리고 마지막에 할당 해제를 해주어 메모리가 새는 것을 방지하였다.

다음으로 commands.c의 fetch\_command를 작성하였는데, 내가 파싱한 커맨드의 이름과 커맨드 구조체 배열의 이름이 같은 것을 리턴하게 하였다. 그 뒤에 do\_pwd에서는 먼저 버퍼를 설정하고, getcwd로 경로를 받아와 출력했다. 그런데 printf를 쓰면 출력 순서상의 오류가 발생하여 write을 사용하여 출력해주었다.

그리고 do\_cd에서는 현재 디렉토리를 변경하는 chdir을 사용했고, 디렉토리가 없거나 디렉토리가 아니면 1과 2를 리턴하여 err\_cd에서 각각의 경우에 에러 메시지를 출력하였다.

다음으로 does\_exefile\_exists에서 실행가능한 파일인지 확인하는 것을 구현하였는데, 여기서는 파일의 상태를 확인해주는 stat을 사용하였다. 그리고 만약에 상대경로에서 실행 가능하면 절대경로를 확인하기 위해 getenv를 사용하여 PATH를 받아와 파싱하였다. strtok를 사용하여 :로 끊어 파싱한 후에, 파싱한 경로의 파일이 실행가능한지 다시 한번 stat을 사용하여 확인하였다.

마지막으로 mysh.c에서 argv[0]에 있는 프로그램을 실행하는 것을 구현하였다. 먼저 포크를 한 후에, 그것이 child라면 execvp를 사용하여 지정된 경로의 파일을 실행하였다. 만약 parent라면 waitpid를 해주었다.

이번 과제에서는 valgrind를 사용했을 때, memory leak이 나지 않아도 다른 메모리관련 오류가 발생할 수도 있다는 것을 배웠고, 문자열을 파싱하는 것이 조금 더 능숙해진 것 같다. 또한 exec 계열의 함수를 어떻게 사용하는지 알게 되었다.

그리고 이번 과제에서 리눅스에서 shell이 하는 역할과 어떻게 돌아가는 것인지 조금은 알게 되었지만, 파싱하는 것부터가 구현하기 매우 어려웠다. 다음 과제에서는 이런 부수적인 것은 조금 더 간단하게 해주셨으면 하는 작은 소망이 있다. 그리고 테스트케이스에 대한 검토를 더 확실하게 해 주셨으면 좋겠다.