

OS - Assignment 1

201720715 박주현

1. 무엇을 했고, 어디까지 진행했는지

Mysh2, 운영 체제의 두 번째 과제는 shell의 기본적인 기능을 가지는 shell을 구현하는 것이다. 아래에 자세히 설명하겠다. System call은 구글링으로 많이 검색해 보았다.

1. void parse_command(const char* input, int *argc, char*** argv)

: input값에 처음에 "이 있는지 보고 "이 없으면 띄어쓰기로 strtok를 사용하였다. 만약 "이 있으면 먼저 "로 strtok를 사용한 후, ""로 묶여 있지 않은 부분을 다시 띄어쓰기로 strtok를 사용하여 파싱하였다. 또한 마지막 원소를 NULL로 초기화 해주었다. malloc을 사용해 처음 input값을 strcpy를 이용해 사용하였고, 개행 문자 처리를 위해 wt, wn으로도 파싱하였다.

2. int does_exefile_exitst(const char* path)

: path에 있는 파일이 실행 가능한지 아닌지 확인하기 위해 access함수를 사용해 access함수의 return값이 0이면 실행 가능하다고 판단하였다.

3. struct command_entry* fetch_command(const char* command_name)

: command name에 해당하는 struct commands를 반환해준다. Command name이 pwd나 cd라면 주어진 구조체에서 아니라면 return NULL로 반환해주었다.

4. int do_pwd(int argc, char** argv)

: system call getcwd()를 사용하여 현재의 working directory를 출력하였다. 또한 fflush를 사용해서 test5번에서 순서가 바뀌는 것을 해결하였다. 이 역시, 구글링을 통해 알게 되었다.

5. int do_cd(int argc, char** argv)

: 앞서 구현했던 does_exefile_exists 함수로 argv[1]에 있는 파일이 실행 가능한지 먼저 판단하였고, argc가 2일때, chdir system call을 이용해서 구현하였다.

6. void err_cd(int err_code)

: 파라미터의 값에 따라 err를 처리해주었다.

7. main.c의 TODO

: 수업시간에 배운 fork를 이용하여 자식 프로세스를 사용하고 exec함수중 excve를 사용하여 command를 처리해주었다.

2. 이번 과제를 진행하면서 배우게 된 것

: fork를 수업시간에 배웠는데, 과제를 하면서 어떤 식으로 돌아가고, 왜 기다려야 하며 wait을 해주지 않았을 때 꼬이는 것, exit이 없으면 무한루프가 돈다는 등 여러가지를 몸소 느꼈다. 또 6번 보너스 문제를 할 때 스스로 부족함을 많이 느꼈다. 6번을 해결하기위해 상대경로를 절대경로로 바꾸기 위해 구글링도 많이 해보고, 여러 함수를 알게 되었지만 제대로 사용하는 못하는 탓인지 계속 에러가 났고 원하는 대로 되지 않았다. 그래서 rm에 /bin/을 strcat하여 6번을 해결하였다.

3. 이번 과제에 대한 피드백

과제가 정말 처음엔 너무 막막하고 어려웠다.저번 과제가 없이 이번 과제가 정말로 첫 과제였다면, 정말 힘들었을 것 같다. 물론 저번 과제가 있었지만 여전히 너무 어려웠고 fork와 exec를 처음 사용해서 많이 어려웠다. 또 과제를 해결하는 내내 모든 메일을 읽으면서 생각이 정리되고 모르는 점들을 찾을 수 있었다. 아쉬운 점은 처음에 파싱을 할 때 테스트 케이스를 보고 하면 된다고 하였는데, 나는 ""표시가 있어야 한다고 생각해서 그렇게 파싱을 하였다가, 이번엔 /bin을 bin으로 저장하였다가, 여러 실패를 걸치고 오류가 나서 다시 새로 짰다. 조금 더 구체적으로 파싱에 대한 설명이 나와있으면 좋았을 것 같다.