

## PA2 보고서

시스템프로그래밍실습

2018312280 이상수

Executable:

Ls, man, grep, sort, awk, bc 는 `execvp` 로 구현하였습니다.

Head, tail, cat, cp, mv, rm, pwd 는 `realpath`로 (Make를 실행 후 각 바이너리 파일이 원래 경로에 그대로 존재한다는 가정하에 `./pa2` 를 실행하면 `pa2`에서 `main`함수에서 위 바이너리파일들의 경로를 가져와서 전역변수로 저장을 합니다. 만약 `make`를 실행 후 바이너리 파일들의 경로를 바꿔 버리시면 제 코드가 의도대로 실행이 안될 확률이 높습니다.) 각 파일 실행파일의 경로를 구해서 저장해두고, 필요시 호출하여 `execv` 로 실행시켰습니다.

`mv`는 제대로 구현되지 못했습니다.

각 파일은 `main`하나안에서 `argv`로 인수를 받는 방식으로 다 구성하였습니다.

`Builtin_cmd`는 `minishell.c` 의 `main` 내에 구현하여 거기서 처리하도록 만들었습니다.

`Main`을 지나서 일어나는 대부분의 동작은 `builtin`을 제외하고 첫 `fork`를 `eval`에서 한 다음 실행됩니다. 최초의 `fork` 후 `groupgid`를 최초 `fork` 된 자식의 `pid`에 맞춰 주어 맨 첫 `minishell`의 `pid`와 다르게 하였습니다.

`Pipe` 와 `redirection`은 저희가 배운대로 `dup2`를 활용하였고, 특정 문자(<,>,>>)를

기준으로 각 문자를 나눠 명령을 수행하거나 파일을 읽었습니다.

특히 pipe는 일단 받은 문자에 | 이 포함되어 있는지 판단하고, 문자를 |을 기준으로 여러 string으로 나눴습니다. 처음 string에는 < 가 있는지 판단하여 redirection을 해주었고, 그 다음 for 문으로 나눈 문자중 첫,마지막을 제외한 나머지 개수만큼 fork 를 해주어 문장들을 처리해주고, 마지막 문장은 > 나 >>가 있는지 판단후 redirection을 해주었습니다.

Sigint, sigtstp는 int는 sit\_ign를 하였고, stp는 main에서 받아서 받으면 wait을 실행해 child가 나오면 그 child의 그룹에 sigkill을 날렸습니다.

Background는 구현하지 못했습니다.