**유닉스 프로그래밍 3차 보고서**

*2019 11 24 일요일*

*12171619*

*박윤수*

1. 요구사항

텍스트이(가) 표시된 사진

자동 생성된 설명

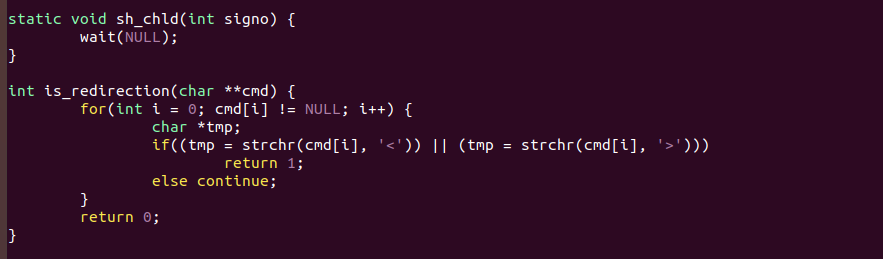
모든 테스트 사항에 대한 결과이다.

리다이렉션과 파이프 테스트를 백그라운드로 실행하면 프롬프트가 먼저 출력된 후 리다이렉션과 파이프 테스트의 결과가 출력된다.

1. 구현 방법



우선 pipevector라는 이차원 문자배열을 선언하였다. 입력받은 명령어 문자열을 ‘|’ 기준으로 자른 후 pipevector에 저장한다. 이 과정에 대한 코드는 이후 메인함수에서 설명하려고 한다.



위의 함수는 인수로 받은 명령어 문자열에 ‘<’나 ‘>’가 있는지 확인하고 있으면 0, 없으면 1을 출력한다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

리다이렉션을 하는 함수이다. 우선 위의 is\_redirection 함수를 통해 리다이렉션이 있는지 확인하고 있을 경우 인수로 입력받은 이차원 문자배열 cmd에서 ‘<’나 ‘>’ 가 존재하는 인덱스를 i에 저장하고, ‘<’ 일 때는 type = 0으로, ‘>’ 일 때는 type = 1으로 저장한다. 이후 각 type에 맞게 stdin과 stdout을 리다이렉션 하고, cmd에서 리다이렉션 표시와 open된 파일명을 지운다.

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

위의 함수는 join 함수이다. 기본 틀은 실습 ppt에서 진행했던 방식과 동일하고, 다중 파이프일 경우를 위해 for문을 이용하는 것만 다르다. ‘A | B | C’ 의 명령어가 입력이 되면 join함수에서 먼저 파이프를 생성하고 손주 프로세스가 A를 실행하는데 A의 결과를 파이프에 출력하고 자식 프로세스에서 그 결과를 파이프에서 읽어온다. 이후 같은 방식으로 B를 실행한다. 마지막 C의 경우 파이프를 생성하지 않았으므로 결과를 stdout으로 출력한다. (각 명령어의 실행은 pipevector에 있는 명령어를 makelist를 통해 exec을 진행할 수 있는 형태로 변경 후 진행한다.)

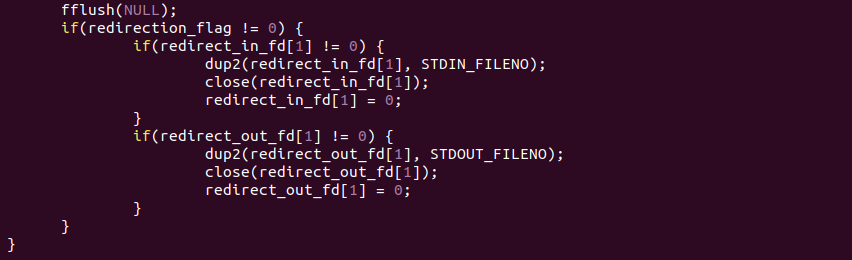
텍스트이(가) 표시된 사진

자동 생성된 설명

메인함수에서 cmdline에 ‘|’나 ‘<’, ‘>’가 있는지 확인하고 있을 경우 pipe\_flag = 1, redirection\_flag = 1로 한 후 위의 코드를 실행하는데, 우선 pipe\_flag가 1일 경우 맨 위의 코드에서 언급했던 명령어를 ‘|’ 기준으로 자르는 작업을 진행한다. cmdvector[i]에서 ‘|’ 이 아니면 tmp라는 문자열에 cmdvector[i]를 이어 붙이고, ‘|’이면 tmp 문자열을 pipevector[j]에 복사한다. (strcpy의 경우 입력받는 문자열이 포인터형식이 아니여야 하므로 동적할당을 이용한다.) 이후 join함수를 진행하는데 backflag가 1이면 백그라운드로 실행, 아니면 그냥 실행한다.

pipe\_flag가 0인 경우 원래의 코드를 실행하는데 리다이렉션을 해야하는지를 redirection\_flag를 통해 확인하고 redirection을 진행한다.

(마지막 부분에선 redirection\_flag로 리다이렉션을 해야하는지를 확인해야 하는데 오타가 났다. 하지만 is\_redirection으로 다시 확인하므로 크게 문제가 되지는 않는다. 사실, redirection\_flag는 is\_redirection으로 리다이렉션을 확인하기 때문에 필요가 없는 변수이다. 위의 오타는 실제 코드에선 고쳐서 작성하였다.)



마지막에 리다이렉션 과정에서 변화가 생긴 stdin과 stdout을 원 상태로 복구하는 과정이다. 각 redirect\_in\_fd와 redirect\_out\_fd int형 배열에서 2번째 변수에 STDIN\_FILENO와 STDOUT\_FILENO를 복사해 두었고, 이를 이용해 stdin과 stdout을 다시 원 상태로 복구하면 된다.

1. 문제점

고아 프로세스 생성 원리를 이용하여 백그라운드를 구현하면 좀비가 생겨야 하는데 init 프로세스가 너무 빨리 입양하고 wait을 실행해서 좀비가 생기지 않던 문제는 여전히 남아있다. 그래도 입양 전에 myshell 프로세스가 sigchld발생시 wait을 해주는 코드를 삽입해서 좀비가 아에 생기지 않도록 진행하였다.