**시스템프로그래밍 lab05 보고서**

학번: 1815060 이름: 문정현

**1. 쉘 인터프리터 작성**

간단한 쉘 인터프리터를 적성한다. 이 인터프리터는 Linux 명령어를 읽고 해석하여 실행시킨다. 이 인터프리터의 기능은 다음과 같다.

(1) 명령어 실행(Command execution)

[shell] cmd

단어들을 명령어 단위로 저장하기 위해 args1 배열에 저장하여 단어가 끝난 경우 args1배열 마지막에 (char\*) 0을 입력하여 매개변수 끝을 입력한 뒤 fork를 통해 자식 프로세스를 생성한 뒤 자식 프로세스에서 execvp(args1[0],args1)를 실행하도록 하고 부모 프로세스는 자식 프로세스가 끝날 때까지 기다리도록 작성했습니다.

(2) 후면 실행(Background execution)

[shell] cmd&

입력된 단어들 중 &가 있는 경우 wait변수를 0으로 한 뒤 단어가 끝난 경우 일반 명령어 와 다르게 fork를 통해 자식 프로세스를 생성한 뒤 자식 프로세스에서 args1 배열에 &가 저장된 위치에 (char \*) 0으로 저장하여 명령어의 매개변수에서 제외시킨 다음 execvp(args1[0],args1)를 실행하도록 하고 wait이 0이기 때문에 부모 프로세스는 자식 프로세스가 끝날 때까지 기다리지 않고 다음 명령어를 입력받을 수 있도록 작성했습니다.

(3) 명령어 순차 실행

[shell] cmd1; cmd2; cmd



입력한 단어들을 args1배열에 계속 추가하다 입력된 단어의 끝이 ;인 경우 해당하는 단어의 ;를 빼고 args1배열에 넣은 뒤 배열 마지막에 (char\*) 0을 입력하여 매개변수 끝을 입력한 뒤 fork를 통해 자식 프로세스를 생성한 뒤 자식 프로세스에서 execvp(args1[0],args1)를 실행하도록 하고 부모 프로세스는 자식 프로세스가 끝날 때까지 기다린 후 다음 명령어를 처리하여 실행하도록 작성했습니다.

(4) 입/출력 리디렉션(Input/Output redirection)

[shell] cmd > outfile

[shell] cmd < infile



들어온 단어가 <, >인 경우 fork로 자식 프로세스를 생성한 뒤 자식 프로세스에서 마지막으로 들어온 단어에 해당하는 파일을 읽기, 쓰기 모드로 open하여 fd변수에 파일디스크립터를 저장합니다. <인 경우에는 dup2(fp,0), >인 경우 dup2(fp,1)를 통해 입출력 리디렉션을 한 후 입출력 재지정 연산자와 파일 이름값은 args1배열에서 (char \*) 0으로 변경합니다. 자식 프로세스에서 execvp(args1[0],args1)를 실행하도록 하고 부모 프로세스는 자식 프로세스가 끝날 때까지 기다리도록 작성했습니다.