**Multicore Programming Project 1**

담당 교수 : 최재승

이름 : 황해준

학번 : 20191671

1. **개발 목표**

-Phase 1

단일 명령어를 입력했을 때 올바른 수행 결과가 나올 수 있도록 한다. Fork 함수를 통해 자식 프로세스를 생성, 다양한 명령어를 수행하도록 한다. 일부 명령어의 경우 자식 프로세스가 아닌 built-in으로 구현한다.

-Phase 2

pipeline을 통해 여러 개의 순차적인 command를 입력 받았을 때 순차적으로 이전 명령의 출력을 다음 명령의 입력으로 받을 수 있게끔 구현한다.

-Phase 3

앞서서 구현한 모든 명령어들이 foreground뿐만 아니라 background에서도 돌아갈 수 있도록 구현한다.

1. **개발 범위 및 내용**
   1. **개발 범위**
2. Phase 1

mkdir, rmdir, cd, ls, history 등 기본적인 명령어들을 정상적으로 수행할 수 있었다.

1. Phase 2

pipeline을 통해 다수의 연속적인 명령을 입력 받았을 때 프로그램이 정상적으로 수행되었다.

1. Phase 3
   1. **개발 내용**

* **Phase1 (fork & signal)**
  + fork를 통해서 child process를 생성하는 부분에 대해서 설명
  + connection을 종료할 때 parent process에게 signal을 보내는 signal handling하는 방법 & flow
* Fork 함수를 통해서는 built-in 명령어가 아닌 명령어들을 수행한다. 먼저 입력한 명령어를 분할한 다음, 해당 명령어가 built-in이라면 그대로 실행, 아니라면 fork 함수를 통해 child process를 생성하여 execve를 통해 생성한 child process로 외부 명령을 불러와서 실행한다. 종료 명령을 입력할 경우 history를 별도의 파일에 저장하고 exit syscall을 사용하여 부모 프로세스에 종료를 알리는 signal을 전달한다.
* **Phase2 (pipelining)**
  + Pipeline( ‘|’ )을 구현한 부분에 대해서 간략히 설명 (design & implementation)
  + Pipeline 개수에 따라 어떻게 handling했는지에 대한 설명
* 만약 명령어에 pipeline이 존재한다면 pipeline이 존재하는 명령어를 처리하기 위한 함수, evel\_pipe를 수행한다. 해당 함수는 반복문을 돌면서 파이프가 존재하는지 확인, 파이프가 존재한다면 그 부분을 널문자로 대체한 다음 반복문을 이어 나간다. 또한 Dup2 함수를 통해 매 반복마다 이번 반복문에서 실행하는 명령의 출력이 다음 반복문의 입력으로 들어갈 수 있게끔 했다. 따라서 pipeline의 개수와 무관하게 위의 동일한 방법으로 일괄적으로 처리할 수 있었다.
* **Phase3 (background process)**
  + Background (’&’) process를 구현한 부분에 대해서 간략히 설명
  1. **개발 방법**
* **B.의 개발 내용을 구현하기 위해 어느 소스코드에 어떤 요소를 추가 또는 수정할 것인지 설명. (함수, 구조체 등의 구현이나 수정을 서술)**

**Phase1**

-myshell의 실행을 종료했다가 재개하더라도 기존에 입력했던 명령어들을 기억하기 위해 history 파일을 생성하거나 기존의 history 파일을 읽어들인다.

-history 명령어를 통해 기존에 입력했던 내용들을 출력하기 위해 파일에서 읽어들인 내용을 바탕으로 history 배열을 동적할당한다.

-echo 명령어를 썼을 때 따옴표와 함께 뒤에 입력이 들어오는 경우를 대비하여 parseline 함수에 따옴표를 제거하는 내용을 추가한다.

-cd, history, exit은 built-in 명령어로 처리하고 나머지 명령어들은 child 프로세스를 생성하여 execve syscall을 통해 처리한다.

Phase2

1. **구현 결과**
   1. **Flow Chart**

* **2.B.개발 내용에 대한 Flow Chart를 작성.**
* **(각각의 방법들에서 추가된 내용(fork, pipeline, background)만 특성이 잘 드러나게 그리면 됨.)**

1. **Phase 1 (fork)**
2. **Phase 2 (pipeline)**
3. **Phase 3 (background)**