상명대학교 컴퓨터과학과 유닉스 프로그래밍 실습 지침서

- 실습 번호: lab-06

- 실습 디렉토리: ~/unix/lab-06 이름: \_\_\_\_\_\_\_\_\_정경재\_\_\_\_\_\_\_\_\_\_\_

- 실습 날짜: 2021년 11월 22일

- 실습 제목: Process and Signal

- 실습 내용: 학번: \_\_\_\_\_\_\_\_201810993\_\_\_\_\_\_\_\_\_

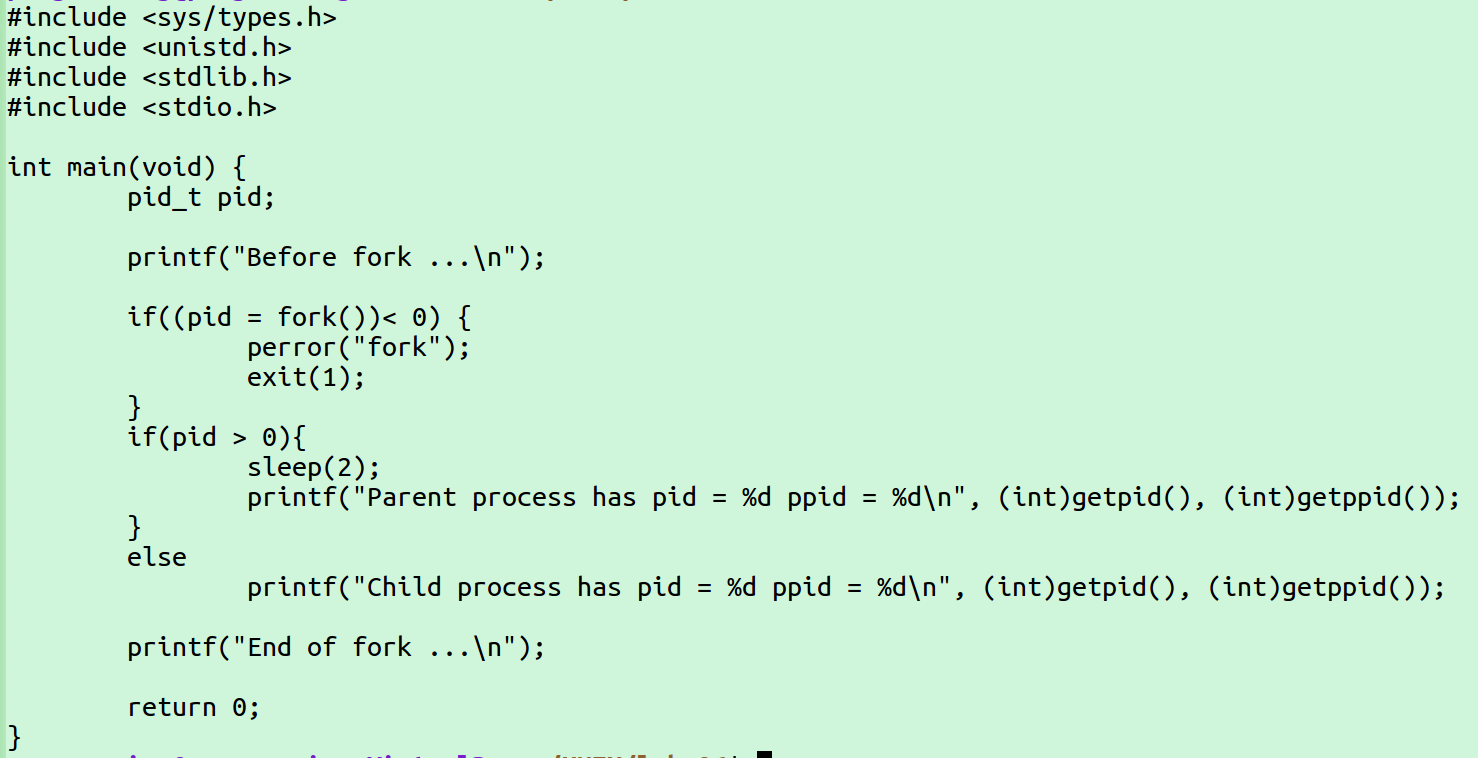
(모든 프로그램 소스에는 comment를 추가하시오

필요한 hearder file은 찾아 넣으시오.

소스와 수행 과정, 결과를 캡처해 붙이시오.)

이번 실습은 우분투를 통해 진행하였습니다.

1. fork()를 이용하여 새로운 프로세스를 만드는 아래 프로그램을 6-1.c로 작성하고 이를 컴파일하여 실행파일 6-1을 만들고 수행하여 결과를 보이고, 결과에 대해 설명하시오.



텍스트이(가) 표시된 사진

자동 생성된 설명

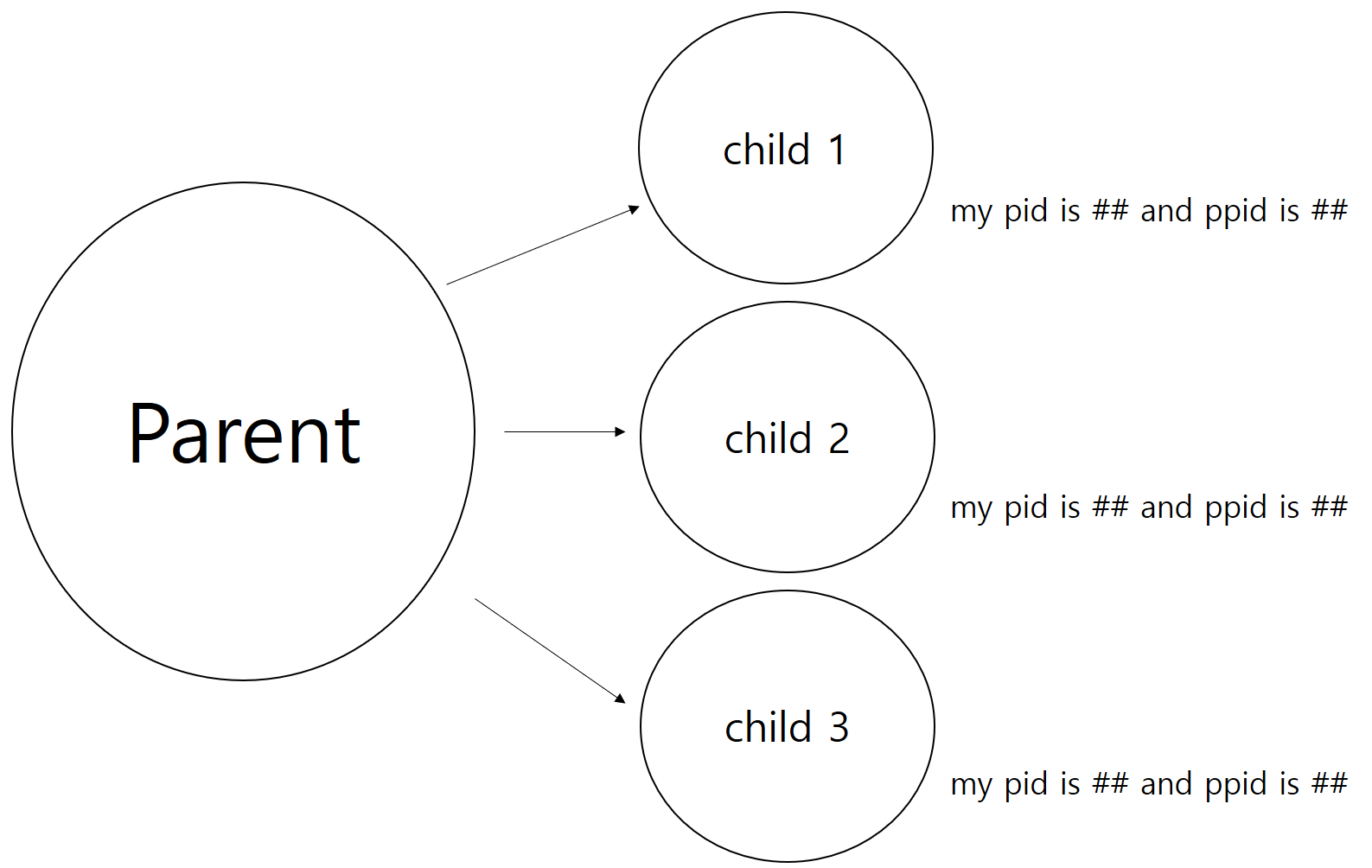
텍스트이(가) 표시된 사진

자동 생성된 설명

해당 프로그램은 fork를 통해 자식프로세스가 생성되고 난 후에 이경우 자식이 먼저 수행되어 자식에 해당 하는 부분이 출력되고 난 후에 “End of fork”를 출력하고 자식은 마무리된다.

그리고 다시 부모 프로세스로 돌아와 2초동안 잠시 멈췄다가 출력문을 작성하고 “End of fork”를 출력하게 되어 해당 결과가 나옵니다..

2. 다음 그림과 같이



fork()를 통해 세 개의 child process를 만들고 각 child에서는 자신의 pid와 parent pid를 출력하는 프로그램을 6-2.c로 만들고 컴파일하고 수행하여 결과를 확인하시오. (6-2.c 제출)

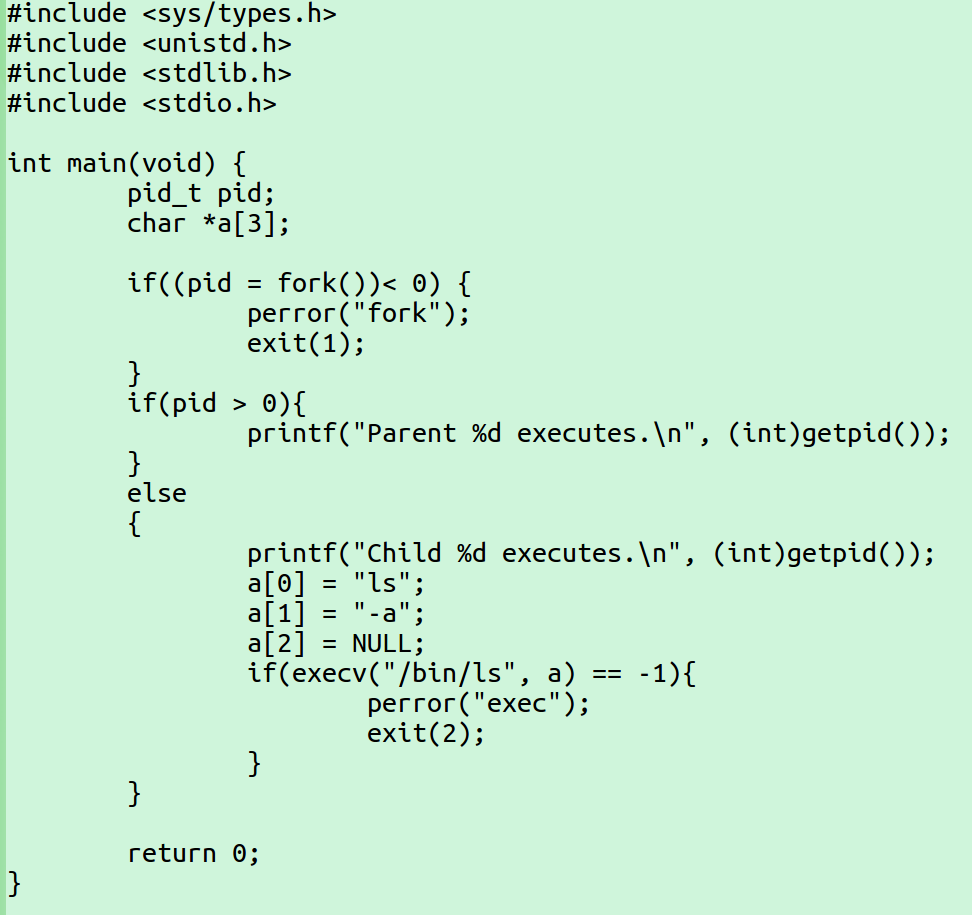
텍스트이(가) 표시된 사진

자동 생성된 설명

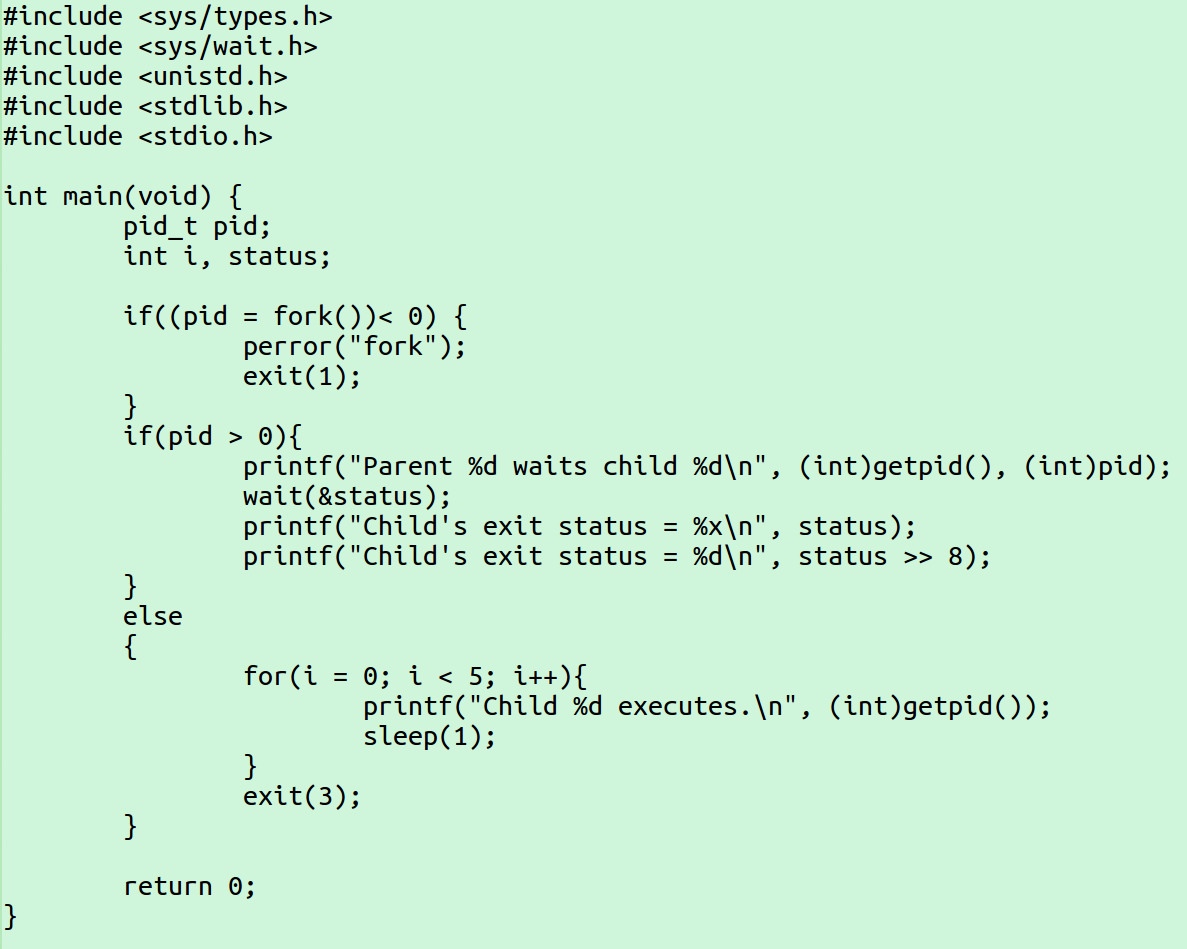
텍스트이(가) 표시된 사진

자동 생성된 설명

3. 아래와 같이 fork()와 execv()를 테스트하는 프로그램 6-3.c를 작성하고 이를 컴파일하여 6-3를 만들고 실행하여 결과를 확인하고 결과에 대해 설명하시오.



4. 아래와 같이 wait()를 테스트하는 프로그램 6-4.c를 작성하고 이를 컴파일하여 6-4를 만들고 실행하여 결과를 확인하고 그에 대해 설명하시오.



텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

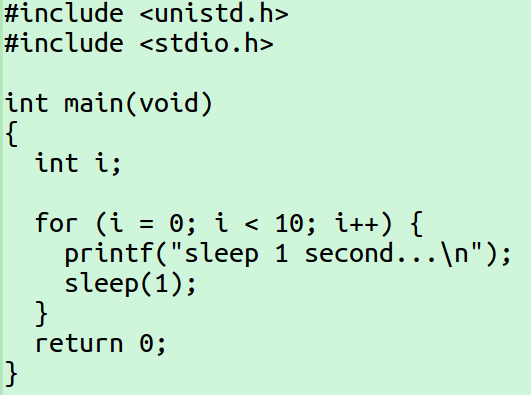
자동 생성된 설명

해당 프로그램은 fork를 통해 자식 프로세스를 생성하고 wait()를 통해 부모 프로세스가 자식 프로세스가 끝나기를 기다리고 있습니다.

따라서 wait를 만난 부모 프로세스는 자식이 끝날 때까지 대기를 하고 있다가 자식 프로세스가 5번의 반복문을 찍고 exit(3)을 통해 status의 값을 3으로 변경하고 돌아오게 되면 그때 부모 프로세스의 나머지 부분이 동작해 status의값에 대해 출력한다.

첫번째는 300000000000을 16진수로 표현하니 300이고 아래는 8칸 쉬프트하여 출력하니 3이 나온다.

5. 아래 프로그램을 파일 6-5.c에 입력하고 컴파일하여 6-5를 만드시오.



텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램을 실행하여 결과를 캡처해 넣으시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

또 프로그램이 종료하기 전에 키 보드에서 Interrupt를 발생하는 'Ctrl-C'를 입력해 보시오. 이때 어떤 현상이 일어나는 지를 확인하고, 캡처하여 넣고, 그 이유를 설명하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램이 실행하는 도중에 인터럽트인 (SIGINT)시그널을 통해 기본 처리방법인 종료가 실행 되어 프로그램이 멈춘다.

6. ‘6-5.c’를 수정하여 이 프로그램이 수행되는 도중 키보드에서 'Ctrl-C'를 입력하더라도 이를 무시하고 계속 수행되게 하는 프로그램 6-6.c를 만드시오. 프로그램을 컴파일 후 실행하여 결과를 캡처하여 넣고, 작동 원리를 설명하시오.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

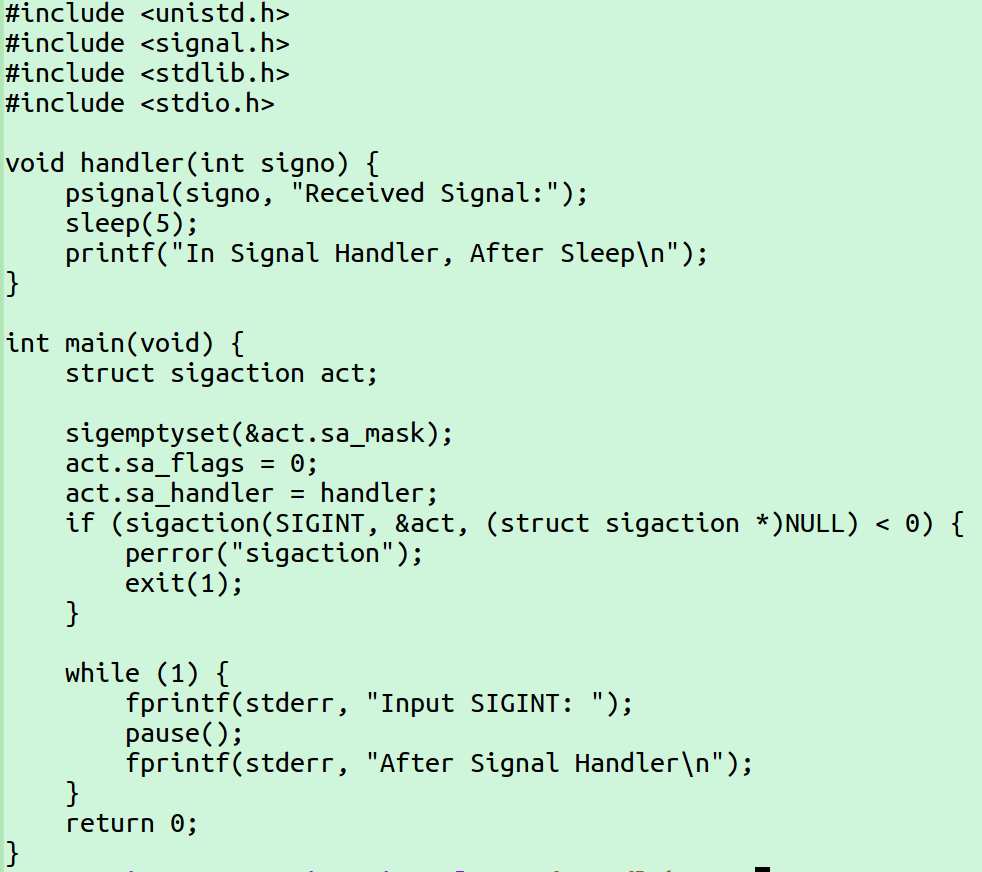
자동 생성된 설명

해당 프로그램은 우리가 signal이라는인터럽트가 발생하면 이를 처리하는 방법으로 가 이를 처리하게 된다.

Ctrl + C의 경우는 종료가 기본적인 처리 방법이므로 이를 sigset을 통해 SIG\_IGN즉 신호를 무시하는 것으로 처리방법을 변경하여 이를 처리합니다.

따라서 Ctrl + C가 아무리 들어오더라도 이 신호는 무시됩니다.

7. 다음과 같은 프로그램 6-7.c를 만들고 실행하여 'Ctrl-C'를 입력하더라도 이 프로그램을 종료할 수 없는 것을 보이시오. 또 왜 이렇게 나오는지 결과를 설명하시오.



텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

기존의 SIGINT의 기본적인 처리는 종료였었습니다.

하지만 sigset을 sigsetempty를 통해 signal을 처리하는 동안 block될 signal집합을 비웠고 SIGINT에 대한 처리방법을 signalset인 action을 통해 처리하도록 변경되어 아무리 ctrl+C를 눌러도 프로그램이 죽지 않는다.

끝.