실습 과제3

사이버보안학과

201620641

유 상 정

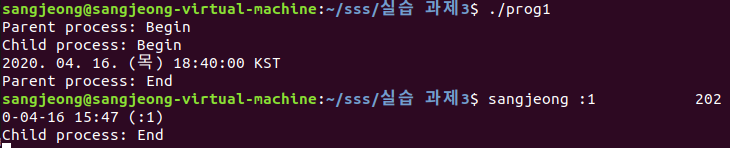
1. 과제 소개

* Fork 를 사용하여 parent 프로세스에서는 현재 시간을 출력하고, child 프로세스는 로그인한 사용자의 정보를 출력하는 프로그램을 작성
* 프로세스 동기화가 되지 않은 프로그램, 동기화를 한 프로그램 2가지 작성

1. Parent process와 child process가 process 동기화가 필요한 이유

* Fork 함수로 자식 프로세스를 생성하면 parent process와 child process는 순서에 관계없이 실행되고, 먼저 실행을 마친 process는 종료한다. 그러나 parent process와 child process 사이의 종료 절차가 제대로 진행되지 않을 때가 있는데, 이때 zombie process나 Orphan process 같은 불안정 상태의 process를 방지하려면 parent process와 child process를 동기화 해야 한다.

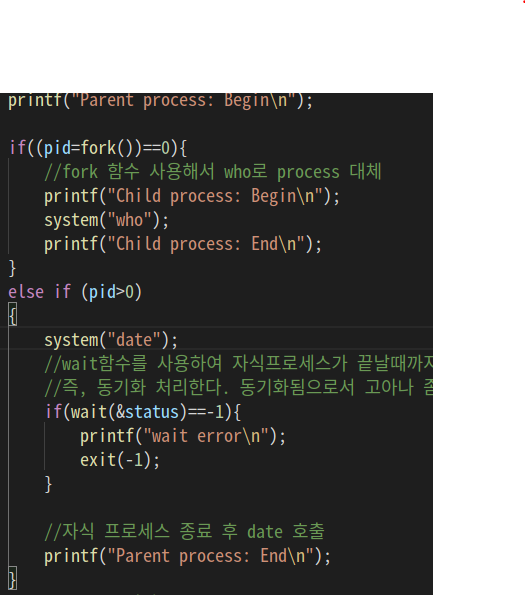
1. 코드 설명
2. 동기화되지않은 프로그램



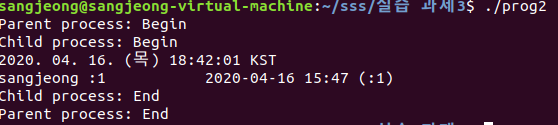
(동기화되지 않은 코드 실행)

이 코드는 parent process가 fork 호출 후 wait함수를 사용하지 않은 code이다. 그래서 parent process와 child process 사이의 순서가 불분명 하다. 위 실행 결과를 보면, 부모 process가 먼저 종료되어서 child process가 할 일을 마쳤지만, 종료되지 않는 경우가 발생하였다. 이와 같이 zombie process가 생기는 경우를 막기위해서는 wait함수를 사용하여 parent process가 child process의 종료 상태를 받아야한다.

1. 동기화된 프로그램



(동기화 된 코드)



(동기화 된 코드 실행화면)

우선 code를 보면 pid > 0 (부모 프로세스)일 때, wait함수를 호출하였다. 그래서 실행 화면을 보면, Parent process와 child process가 시작된 후 parent의 역할인 date가 호출되었다. 그 후에 child의 역할인 who가 나타난다. 그러나 종료 순서는 child 프로세스가 먼저 종료되었다. 즉, wait함수를 사용하여, 자식 프로세스가 종료되면 wait함수가 종료신호를 받고, 자식 프로세스를 process table에서 제거되게 한 후, parent process가 종료되었다. 따라서 자식, 부모 프로세스가 모두 정상 종료되었다. 즉, 프로세스간 동기화가 잘 이루어 졌다고 할 수 있다.