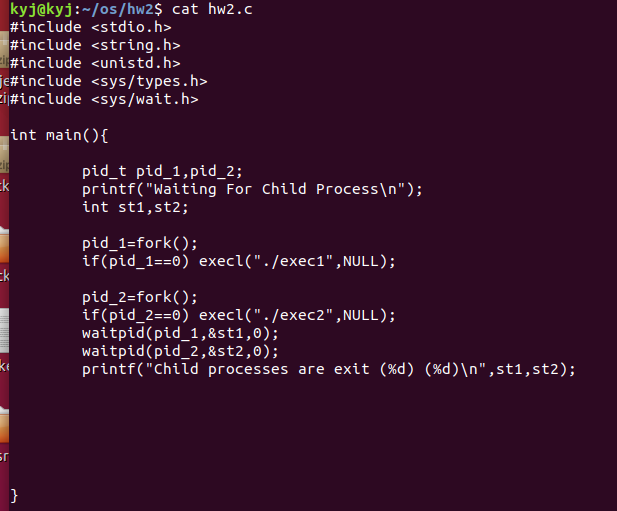
**2019062833** **컴퓨터 소프트웨어학부 김유진**

**운영 체제 HW#2**

**제출 일자: 2021/03/27**

1. **과제 A**
2. **부모 프로세스, 자식 프로세스1, 자식 프로세스2 코드 설명**
   1. **부모**

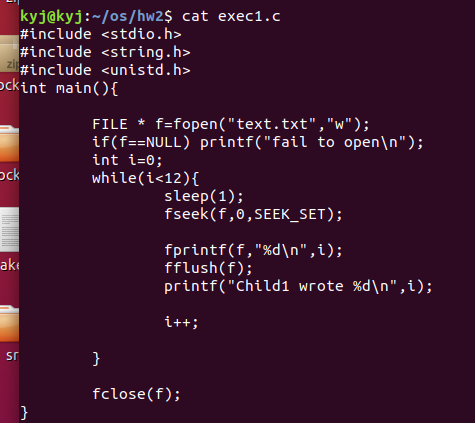


첫번째 자식을 생산한 후 그 분기점이후에 본인이 자식이면 exec1 파일로 새로운 프로그램을 입음

두번째 자식을 생산한 후 그 분기점 이후 본인이 자식인 경우 exec2 파일로 새로운 프로그램을 입음

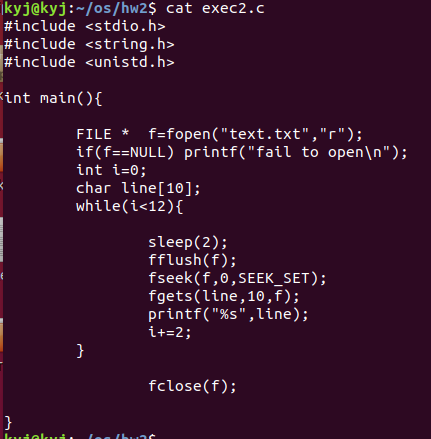
부모는 자식이 다 종료될 때까지 기다림

* 1. **자식1**



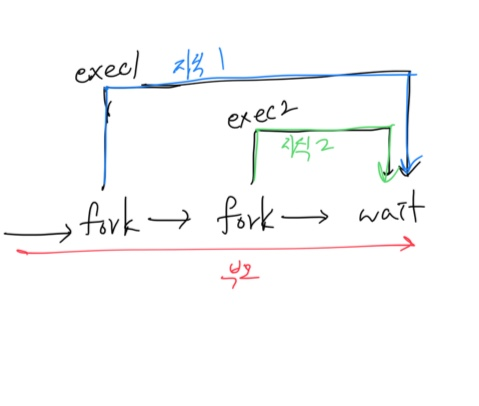
내용을 쓸 파일을 연 후, 파일의 첫 부분에 반복적으로 숫자를 1씩 증가시키며 적는다. fflush를 통해 write를 실제로 적용 시킨다.

* 1. **자식2**

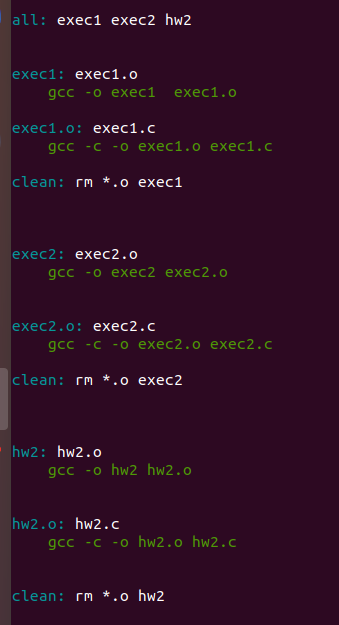


내용을 읽을 파일을 연 후, 파일의 첫 부분에서 반복적으로 숫자를 읽어온다. fflush를 해주지 않으면 write으로 수정한 파일 내용이 갱신되어 있지 않기 때문에 따로 적어준다.

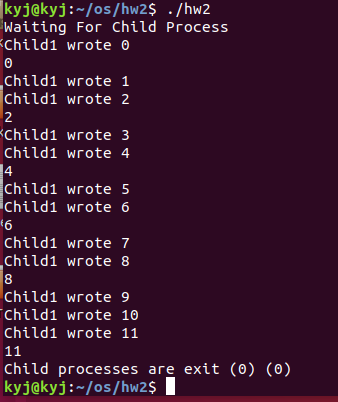
1. **구현한 fork-exec 구조 설명**



1. **컴파일 방법 설명 (Makefile 등)**



1. **실행 결과**



1. **과제 B**
2. **IPC 내용**
   1. **IPC란 프로세스 사이의 데이터 통신**
   2. **대표적인 IPC**
      1. **Pipe**
         1. **단방향 통신으로 부모 자식 간의 프로세스 통신**
         2. **한 쪽에서는 읽기만 하고 다른 쪽에서는 쓰기만 하는 구조에 좋음**
      2. **Named Pipe**
         1. **pipe에 이름이 있는 것.**
         2. **FIFO라는 특수 파일을 사용해서 pipe 역할 대신**
         3. **관련 없는 프로세스들 사이에서 pipe를 이용해 통신하기 위해 만들어짐**
         4. **서버/클라이언트 모델을 따르는 프로그램의 작성이 가능**
      3. **Message Queue**
         1. **메시지를 주고받으며 프로세스간 통신이 이루어짐**
         2. **커널은 메시지를 저장할 장소인 메시지 큐 자료구조를 제공**
         3. **한 프로세스에서 메시지를 생성하여 큐에 집어넣으면 다른 프로세스에서 이를 꺼내가는 구조**
         4. **큐 객체를 식별할 수 있는 고유 번호가 존재.**
         5. **메시지는 사용자가 응용 프로그램에서 구조체를 사용하여 정의하면 됨**
      4. **Shared Memory**
         1. **두 개 이상의 프로세스가 특정한 메모리 영역을 고유하는 것을 말함**
         2. **공유 메모리 영역에 대한 쓰기와 읽기 작업을 통해 데이터 통신이 가능**