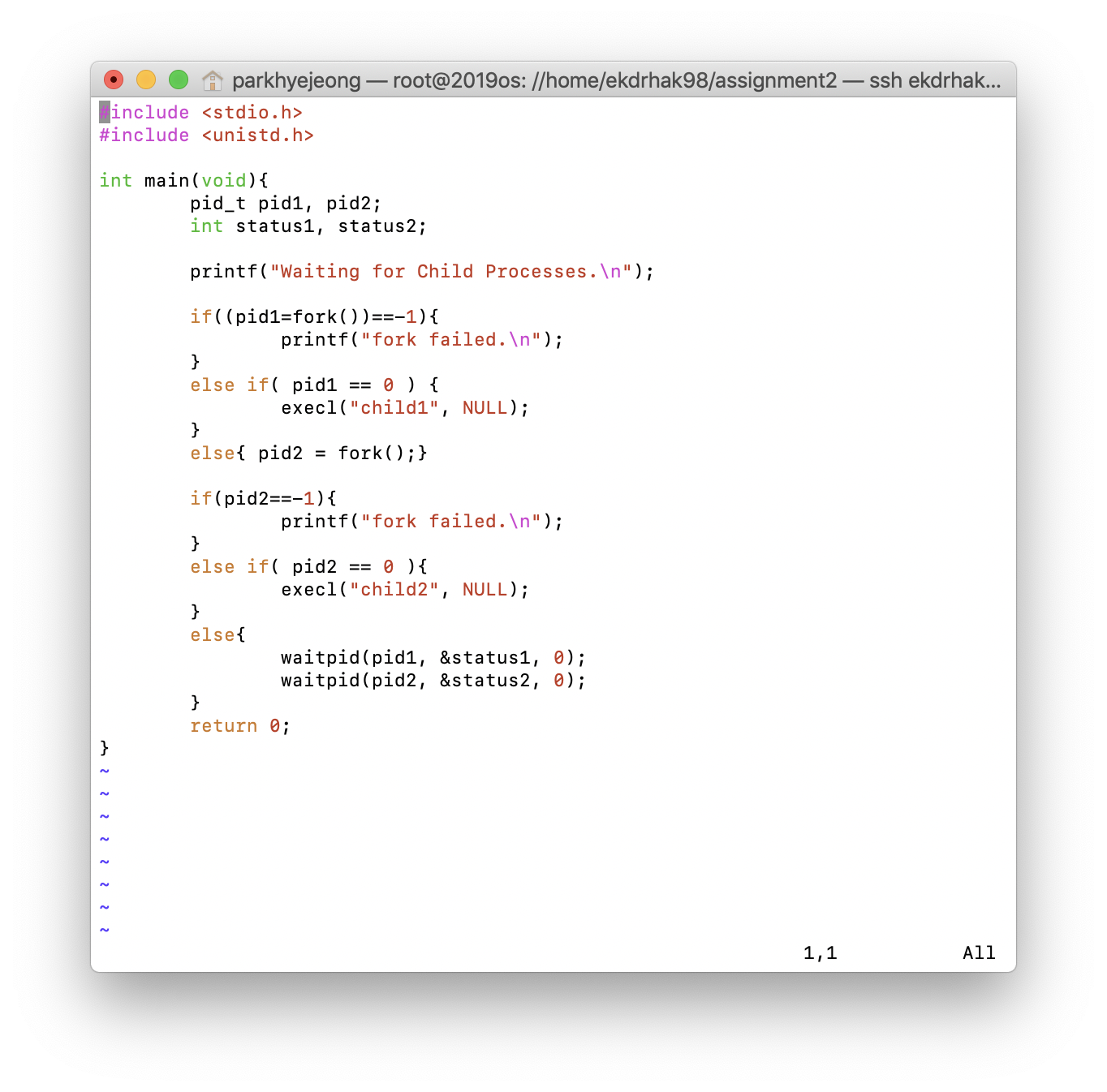
2017029716 박혜정

과제 A



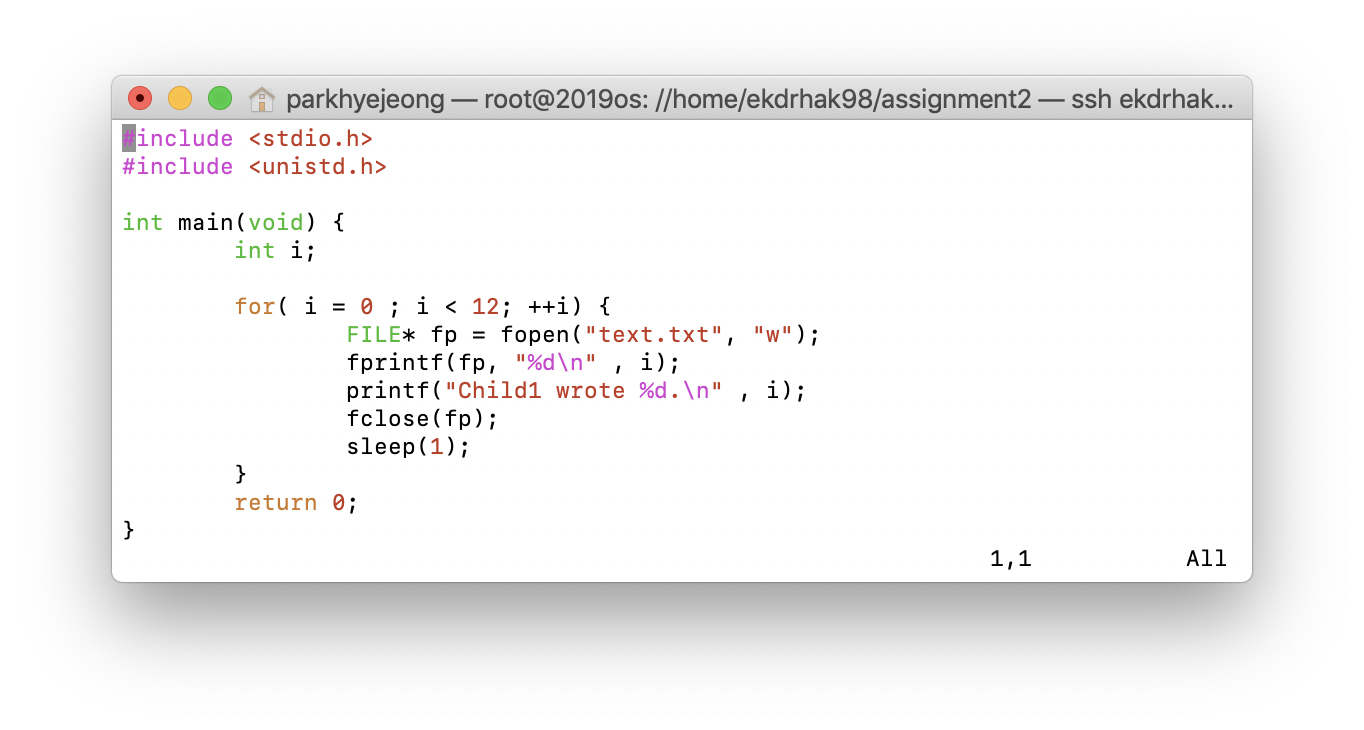
main.c

pid1 = fork()를 통해 프로세스 하나를 복사 후

execl(“child1”,NULL); 로 child1으로 대체

pid2 = fork()로 프로세스 하나 더 복제

execl(“chlid2”, NULL);로 child2로 대체



child1.c

“w”모드로 text.txt파일을 열어서 숫자와 개행문자 입력 후 파일 닫음

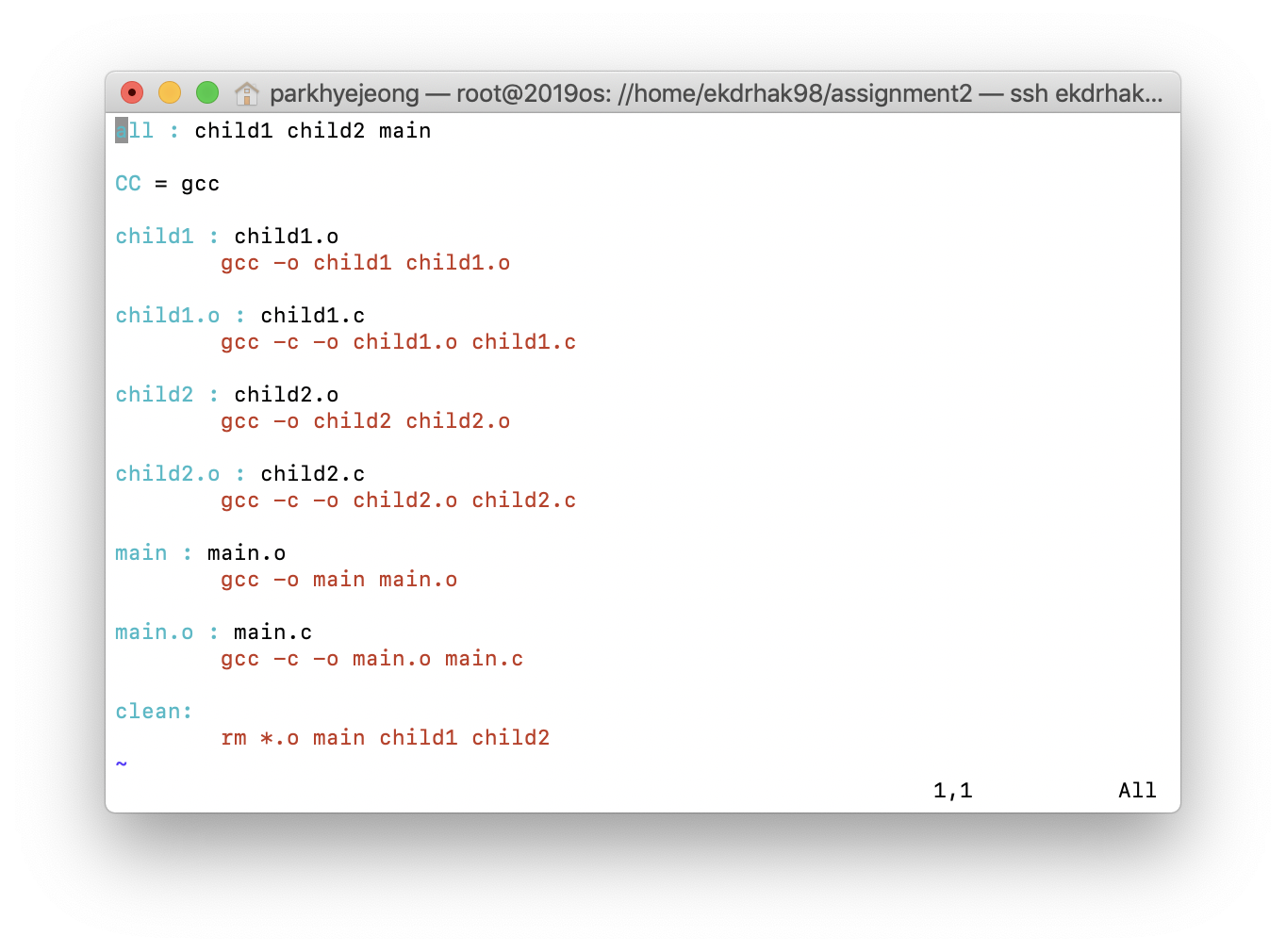
그 후 1초간 sleep



chlid2.c

“r”모드로 text.txt파일을 열어서 한 줄씩 읽고 출력 후 파일 닫음

그 후 2초간 sleep

Makefile

컴파일 방법 : make

이후 실행 : ./main

(왜인지 두번째 실행부터 제대로 돌아갑니다 ㅜㅜ

넓은 아량으로 한 번만 더 돌려주세요 ㅜㅜ)

과제B

ipc(Inter-Process Communication)

: 프로세스들 사이에 서로 데이터를 주고 받는 행위 또는 그에 대한 방법이나 경로

1) PIPE

두 개의 프로세스를 연결하게 되고, 하나의 프로세스는 데이터를 쓰기만, 다른 하나는 데이터를 읽기만 할 수 있다. 한쪽 방향으로만 통신이 가능한 파이프의 특징 때문에 Half-Duplex(반이중) 통신이라고 부르기도 한다.

2) Named PIPE(FIFO)

통신을 할 프로세스가 명확하게 알 수 있는 경우 사용, 예를 들어 자식과 부모 프로세스간 통신의 경우에 사용할 수 있으며, Named PIPE는 전혀 모르는 상태의 프로세스들 사이의 통신의 경우 사용

Named PIPE는 부모 프로세스와 무관하게 전혀 다른 모든 프로세스들 사이에서 통신이 가능한데 그 이유는 프로세스 통신을 위해 이름이 있는 파일을 사용하기 때문

3) Message Queue

Queue(큐)는 선입선출의 자료구조를 가지는 통신설비로 커널에서 관리한다. 입출력 방식으로 보자면 위의 Named PIPE와 동일

Named PIPE와 다른 점이라면 Name PIPE가 데이터의 흐름이라면 메시지 큐는 메모리 공간이라는 점

4) Shared Memory

공유메모리가데이터 자체를 공유하도록 지원하는 설비

프로세스간 메모리 영역을 공유해서 사용할 수 있도록 허용

공유메모리는 중개자가 없이 곧바로 메모리에 접근할 수 있기 때문에 다른 모든 IPC들 중에서 가장 빠르게 작동할 수 있다.

5) Memory Map

Shared Memory와 마찬가지로 메모리를 공유

하지만 Memory Map의 경우 열린파일을 메모리에 맵핑시켜서, 공유

6) Socket

프로세스와 시스템의 기초적인 부분이며, 프로세스 들 사이의 통신을 가능하게 한다.

<sys/socket.h>라는 헤더를 이용하여 사용할 수 있으며, 같은 도메인에서의 경우에서 연결 될 수 있다.

7) Semaphore

Named PIPE, PIPE, Message Queue와 같은 다른 IPC설비들이 대부분 프로세스간 메시지 전송을 목적으로 하는데 반해, Semaphore는 프로세스 간 데이터를 동기화 하고 보호하는데 그 목적을 둔다.

프로세스간 메시지 전송을 하거나, 혹은 Shared Memory를 통해서 특정 데이타를 공유하게 될 경우 발생하는 문제가 공유된 자원에 여러개의 프로세스가 동시에 접근하면 안되며, 단지 한번에 하나의 프로세스만 접근 가능하도록 만들어줘야 함, 이 때 사용되는 것이 Semaphore