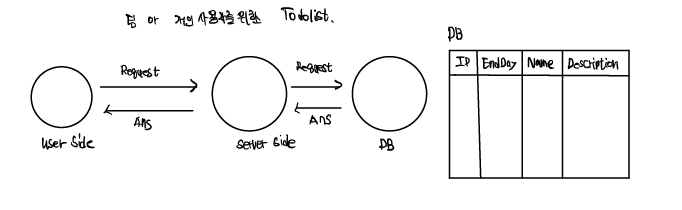
**2021 UNIX 프로그래밍  
프로젝트 2:   
소켓 프로그래밍을 활용한 팀 또는 개인 일정관리 프로그램**

201810993 정경재

1. **주제 설명**  
   이번 프로젝트 2는 팀 또는 개인의 해야 하는 작업들을 효율적으로 관리하기 위한 Todolist를 socket programming과 SQLITE를 활용해서 제작해봤습니다.
2. **시스템 설계도**시스템에 대한 설계도는 다음과 같습니다.  
   ****  
   해당 사진과 같은 구조를 가지고 있습니다.  
   사용자는 6가지의 작업을 수행 할 수 있습니다.  
   첫번째로는 특정날짜에 해야 하는 일에 대한 데이터를 작성해서 서버에 전달할 수 있습니다.  
   두번째로는 특정 ID에 대한 해야 하는 일을 지우는 작업을 할 수 있습니다.  
   세번째로는 모든 해야 하는 일을 제공 받을 수 있습니다.  
   네번째로는 특정 ID에 대한 해야 하는 일을 제공 받을 수 있습니다.  
   다섯 번째로는 오늘 날짜에 대한 해야 하는 일을 제공받을 수 있습니다.  
   여섯 번째로는 원하는 ID에 해당하는 해야 하는 일의 마감날짜를 변경 할 수 있습니다.

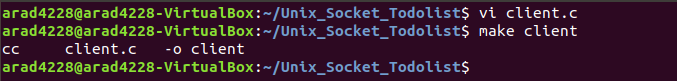
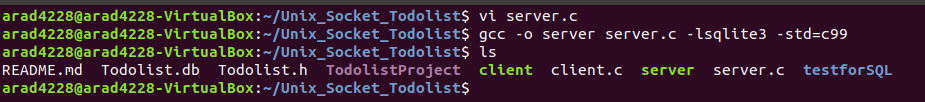
서버는 다음과 같은 작업을 할 수 있습니다.  
첫번째로는 서버는 사용자로부터 받은 데이터를 DB에 저장 할 수 있습니다.  
두번째로는 서버는 사용자로부터 받은 데이터를 DB에서 삭제할 수 있습니다.  
세번째로는 서버는 사용자에게 모든 DB의 데이터를 제공할 수 있습니다.  
네번째로는 서버는 사용자로부터 특정 ID에 대한 데이터를 제공 할 수 있습니다.  
다섯 번째로는 서버는 사용자에게 오늘 날짜에 대한 해야 하는 일을 제공 할 수 있습니다.  
여섯 번째로는 사용자가 수정하길 원하는 데이터를 수정할 수 있습니다.

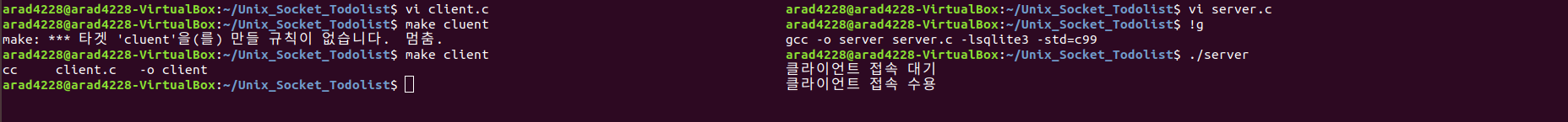
다음은 DB(SQLITE)입니다.  
DB는 위의 사진에서 보시다 싶은 테이블의 구조를 지니고 있습니다.  
첫번째 ID는 DB의 primary 값으로 검색이나 삭제에 사용됩니다.  
두번째로 EndDay는 해야 하는 일에서 마감일에 해당합니다. 해당 column을 primary key로 사용하지 못하는 이유는 같은 마감일에 해야 하는 일이 다수 일 수 있기 때문입니다.  
세번째로 Name는 해야 하는 일의 이름을 의미합니다.  
네번째로 Description은 해야 하는 일의 세부사항입니다.

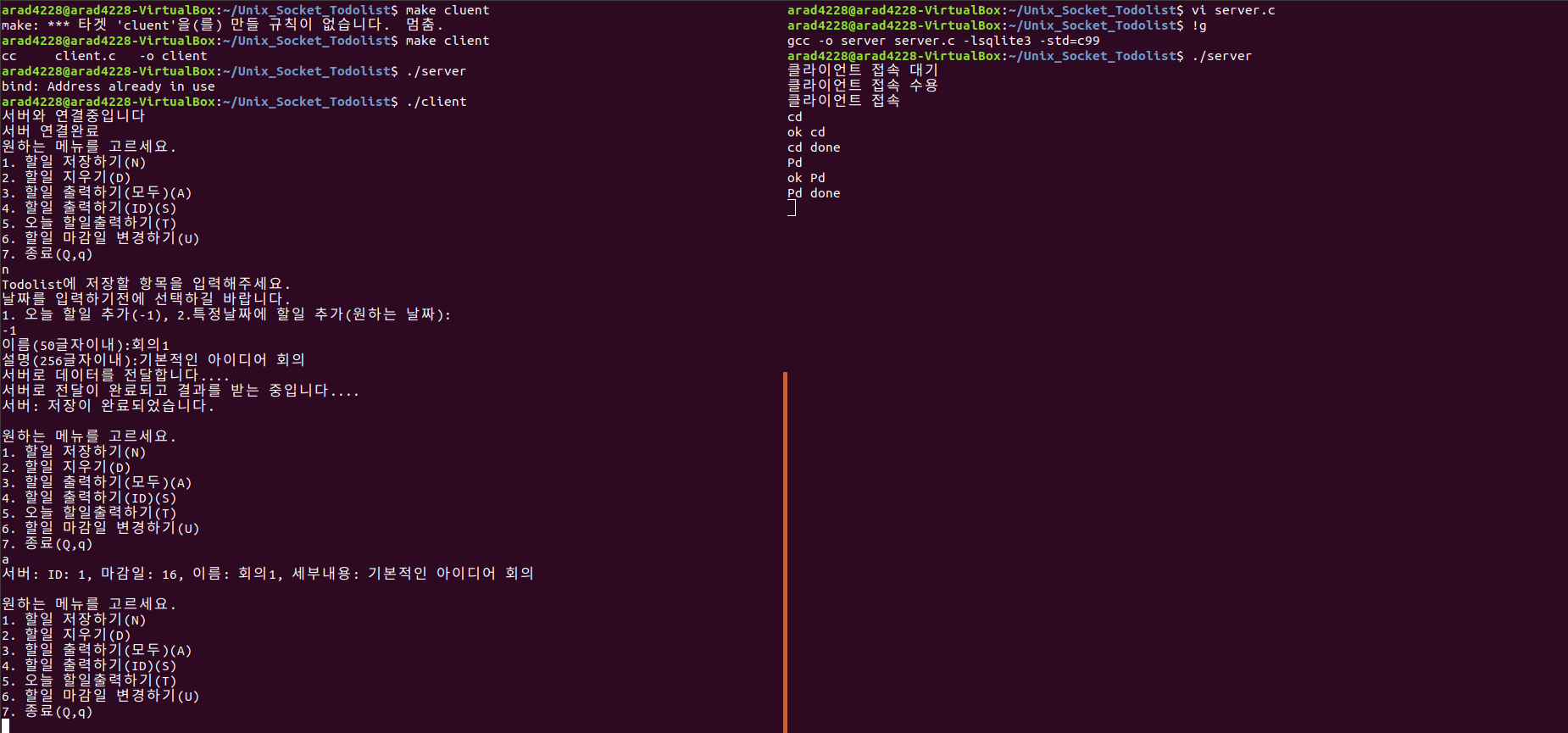
1. **사용된 기술**첫번째로는 소켓 프로그래밍에서 TCP/IP통신을 통한 소켓프로그래밍을 통해 반복서버를 구현하였습니다.  
   왜냐하면은 사용자-서버 구조에서 서버는 사용자에 붙어있지 않기에 서버는 따로 동작하는 상황을 만들고자 TCP/IP 통신하며 로컬 host의 9000번 포트를 통해 통신을 합니다.  
   두번째로는 SQLITE 즉 DB를 사용했습니다.  
   사용자의 데이터를 효율적으로 관리하기 위해 DB를 통해 관리하도록 하였습니다.  
   세번째 Unix의 시스템정보 중 시간정보를 사용해서 작성되었습니다.  
   사용자의 옵션 중 오늘 해야 하는 일을 작성하기 위해 time 정보를 받아와서 사용했습니다.
2. **사전 준비 및 컴파일 방법  
   - 사전 준비**1. 먼저 SQLITE3를 설치 해 주셔야 합니다.  
   첫번째 방법: 직접 파일을 통해 설치합니다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   두번째 방법: sudo apt install sqlite3를 통해 설치가 가능합니다.  
   텍스트이(가) 표시된 사진

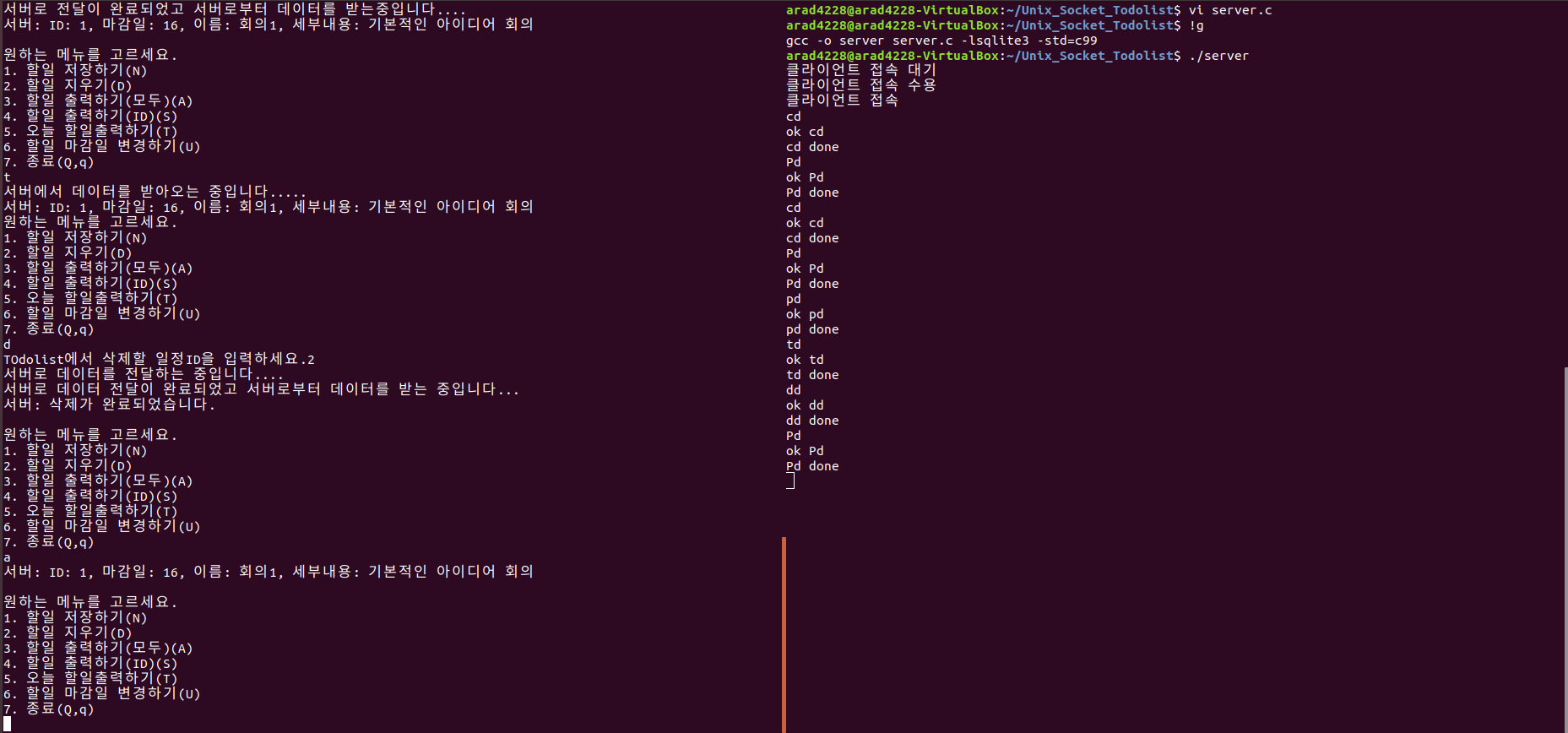
   자동 생성된 설명  
   2. 두번째로 sqlite3의 library를 다운 받습니다.  
   Sudo apt-get install libsqlite3-dev을 통해 다운을 받을 수 있습니다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   **- 컴파일 방법**먼저 client입니다. ****  
   다음은 server입니다.  
     
   gcc -o server server.c -lsqlite3 (-std=c99)  
   를 통해 서버를 컴파일 할 수 있습니다.
3. **사용 매뉴얼 및 사용 예시**1. 모든 파일을 컴파일을 해줍니다.  
   2. 컴파일이 정상적으로 끝났다면 서버를 작동시킵니다.

  
3. 클라이언트를 접속시킵니다.  
텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명  
4. 사용자는 자신이 원하는 메뉴를 괄호안의 영어를 통해 제어합니다.  
  
텍스트이(가) 표시된 사진

자동 생성된 설명

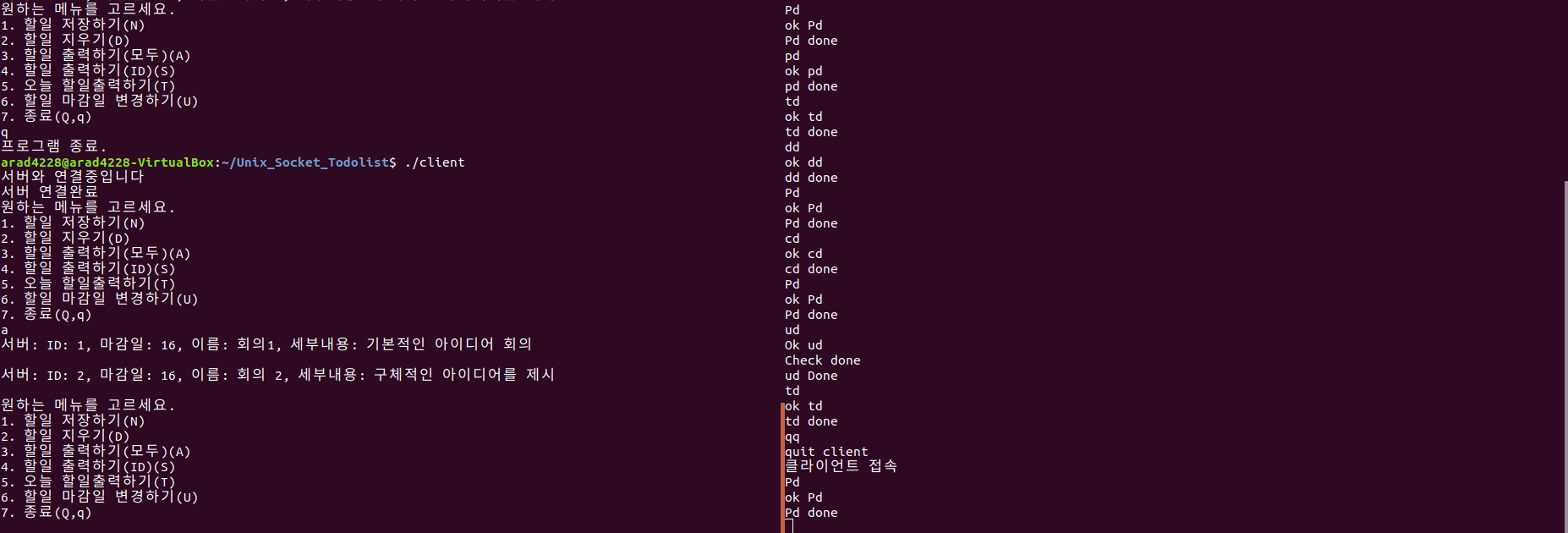


텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

해당 과정을 통해 사용자는 원하는 날짜에 해야 하는 일을 추가하고  
해당 일이 마무리 되었다면 삭제를 아니면 기한을 뒤로 수정하는 작업을 통해 해야 하는 일을 효율적으로 관리 할 수 있습니다.  
또한 서버는 반복서버고 항상 on이 되어있으므로 사용자 작업을 끝내고 나가도 서버는 매번 새로운 사용자를 받아줍니다.  


그리고 서버에는 log형식으로 사용자가 작업한 목록이 나와있어 사용자가 들어왔을 때 어떠한 작업을 진행했는지 확인이 가능합니다.

또한 서버를 꺼야 하는 경우는 ctrl+ q 를 통해 off를 할 수 있습니다.  
🡪 이유는 서버는 항상 동작하며 사용자에게 서비스를 제공한다고 가정하였습니다.