Socket Programming #1

Name: Jang Ju Man

Student Number: 21700633

1. **TCP가 byte stream service를 제공한다는 의미를 파악하기 위함입니다.**
   1. Requirements
      1. Server에서는 단지 client에서 전송한 데이터를 화면에 출력시킨다.
      2. Client에서는 Server와 connection establish 한 다음, 즉시 10 Byte 데이터 3개를 3번의 send() 함수를 이용하여 server에 전송한다.  
         (ex. 0123456789, ABCDEFGHIJ, KLMNOPQRST)
      3. 그 다음부터는 터미널로부터 입력된 데이터를 server에 전송함(한 라인의 데이터를 받아서 전송함). 그리고 동영상 예에서처럼 사용자가 ‘q’를 입력하면 client 프로그램을 종료 시킴.
      4. Server는 client와 connection establish 한 다음, *5초간 sleep한 다음*, 최대 buffer 크기가 100 Byte인 버퍼를 이용하여 읽은 다음, 읽은 데이터를 화면에 출력시킴(recv() 이용하여 데이터 읽음).   
         동일한 형태로 (최대 buffer 크기가 100 Byte인 buffer를 이용하여 읽음) 수신한 데이터를 읽어서 화면에 출력하는 동작을 반복함.
      5. *Client가 종료될 때, server도 종료 시키려면, client와 server에서는 어떻게 프로그램 하여야 하는가?*
   2. Result
      1. Screen Capture :
         1. Buffer size : 100
         2. Buffer size : 5
      2. Answer :

TCP는 connection oriented protocol이기 때문에 client의 연결이 종료되면 서버에서의 데이터 수신 값이 0이 된다. 이것을 사용하여 조건을 걸면 connection이 끊기면(client의 연결이 종료되면) while문을 탈출하여 다음 시퀀스(return 0)를 진행할 수 있게 된다.

while((str\_len=recv(clnt\_sock, message, BUFSIZE, 0)) != 0)

1. **1번과 동일한 형태의 프로그램에서 단지 Server에서 최대 buffer 크기가 5인 버퍼를 이용하여 client에서 보내 준 데이터를 읽은 다음 화면에 출력시킴.**
   1. Requirements
      1. 1, 2의 실험을 통하여 TCP가 어떻게 동작하는 가를 설명하시오.   
         (보고서에 실행 결과도 캡처해서 포함시킴)
   2. Result
      1. Screen Capture :
         1. 스크린샷이(가) 표시된 사진

            자동 생성된 설명Buffer size : 100
         2. Buffer size : 5

스크린샷이(가) 표시된 사진

자동 생성된 설명

* + 1. Answer :   
        TCP는 byte stream service이기 때문에 데이터 경계가 없어서 서버에서 데이터를 잘 모아서 packet으로 만드는 과정이 필요하고 데이터가 버퍼사이즈에 따라 한번에 올 수도, 여러 패킷에 나눠서 올 수도 있다. 그래서 버퍼사이즈가 5인 경우에도 데이터의 손실이 없이 잘 도착하는 것을 볼 수 있다.

1. **이 실험은 UDP가 message oriented service를 제공한다는 의미를 파악하기 위함입니다. 1번과 동일한 형태로 동작하는 UDP server와 client 프로그램을 작성하여 동작을 시킴으로써, UDP의 특성을 파악하여 UDP 특성을 기술하십시오**
   1. Requirements
      1. Server에서 읽을 때에는 recvfrom() 함수를 이용해야 함.
      2. UDP는 connectionless service이다. *Client가 종료되면 server도 종료시키고 싶을 경우 어떻게 해야 하는가? (이 실험 환경에서는 패킷 손실이 거의 없겠지만, 있는 경우에도 논리적으로 동작하도록 하여야 함.)*
   2. Result
      1. Answer  
          UDP는 loss가 발생해도 회복 할 수 없는 프로토콜이다. 따라서 client가 종료되었다는 메시지를 서버 쪽에서 받지 못한 경우를 생각해야 한다. client에서 q 메시지를 통해 종료를 서버에게 알려준다. 이 과정에서 패킷이 손실 될 수 있으므로, 패킷을 보내는데 실패하면 얻는 리턴 값 -1을 활용한다. 패킷이 잘 도착했고 이로 인해 서버쪽에서 종료가 완료되면, 서버가 종료되었기 때문에 클라이언트는 -1이라는 리턴 값을 받게 된다. 이것을 확인하면 클라이언트는 종료한다.  
          요점은 서버가 종료될 때까지 클라이언트는 서버에게 종료 메시지를 보내고, 서버가 비로소 종료되면 클라이언트도 close()를 실행하고 프로그램을 종료한다는 것이다. 패킷 손실률이 높을지라도 서버가 종료되지 않는다면 무한대로 패킷을 서버로 보내기때문에 패킷의 손실이 있더라도 실행가능하다.
2. **2번과 동일한 형태로 동작하는 UDP server와 client 프로그램을 작성하여 동작을 시킴으로써, UDP의 특성을 파악하여 UDP 특성을 기술하십시오.**
   1. Requirements
      1. Server에서 읽을 때에는 recvfrom() 함수를 이용해야 함.
      2. 3번과 4번 실험을 통하여 UDP의 동작을 설명하시오.   
         (보고서에 실행 결과도 캡처하여 포함시킴/ TCP의 경우와 비교하여 차이점을 설명하여야 함.)
   2. Result
      1. Screen Capture
         1. Buffer size : 100  
            스크린샷이(가) 표시된 사진

            자동 생성된 설명
         2. 스크린샷이(가) 표시된 사진

            자동 생성된 설명Buffer size : 5
      2. Answer

Buffer size를 100으로 했을 때는 100 이하의 사이즈의 입력 값은 잘 작동한다. 단 buffer size가 5가 되었을 때 처음 출력되는 10byte 3개의 입력 값부터 손실이 발생하고 이후에 5byte이상의 입력들은 앞의 5byte까지만 입력 받아지고 이후에는 데이터가 손실된다. TCP와 다르게 UDP는 데이터가 아예 손실되어 서버에 도착하지 않는 경우도 발생했다. 즉, UDP에는 데이터의 경계가 존재하고 loss의 가능성이 있다. 따라서 UDP는 TCP와 다르게 unreliable하게 작동하고 packet loss가 있을 때 수복하지 못하는 것을 실험결과를 통해 알 수 있다.