lec

4번은 fork쓰는게 아니라 (채팅이 아니니까)

5번 – multi cli 처리하도록.

5번 서버 – loop를 돌면서 accept를 기다림

accept 통과 – 새 cli가 들어온 것임

child마다 sokcet 번호가 달라지게 됨

 !!!

hw

4) Implement simple ftp server and client.

If the client doesn't follow the protocol, the server should stop the communication.

Simple ftp protocol

client => server: hello

server => client: what file do you want?

client => server: file path

server => client: file contents

‘’hello” 들어오면 파일 경로 받고 그 파일 열기

=> servf.c clif.c 로 하기

- fopen( )사용



우선, 제가 ftp통신으로 출력하고 싶은 파일의 내용은 위와 같습니다.

f1파일의 절대경로는 /home/sp/12163291/f1 으로,

./f1입니다.

우선, O\_RDWR 등의 옵션을 위해 10번의 헤더파일(fcntl.h)을 추가합니다.

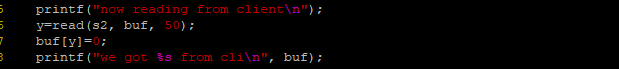
open() write() 등의 함수를 사용하기 위해 11번의 헤더파일(unistd.h)을 추가합니다.



1. client 에서 server로 첫 마디를 보냅니다



2. server에서 client 의 1에서 보낸 첫마디를 읽어들입니다.

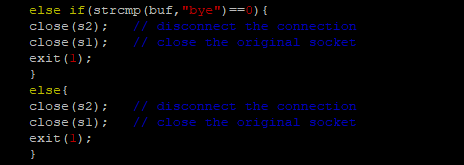


3. server에서, 1에서 받은 첫마디가 hello인지 판별합니다.

만약, hello 이면, 어떤 파일을 열기를 원하냐는 문장을 보냅니다.



만약, hello가 아니면, 연결을 close 합니다.

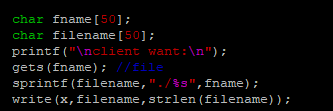


4. client가 1.에서 만약 hello를 보냈을 경우, 서버에서는 어떤 파일을 원하냐는 답을 보낼 것이고, client는 그를 받아 출력합니다.



그리고, 파일의 이름을 입력 받습니다.

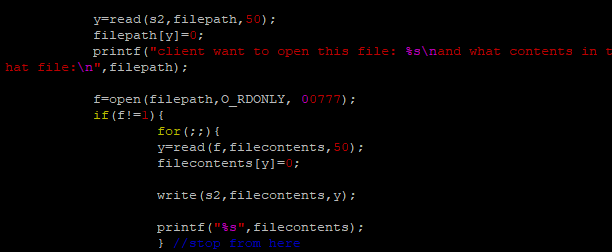
sprintf를 통해, 파일 이름 앞에 ./를 붙여 절대경로화 시켜줍니다.



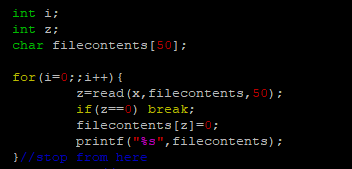
5. server는 client가 4.에서 보낸 파일의 절대경로 값을 받아 f=open에 적용시켜

파일의 내용을 읽은 후,

client로 보내줍니다.

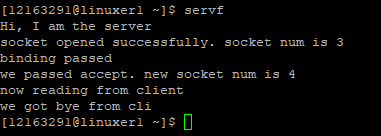


6. client는 server 가 5.에서 파일을 열어서 읽은 후, 보내준 file 내용을 받아 출력합니다.

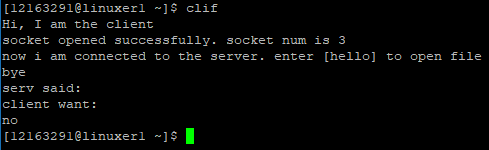


결과를 확인하여 보면 다음과 같습니다.

- bye를 보냈을 경우



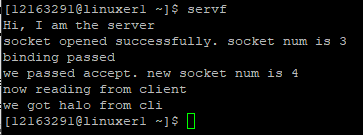
server



client

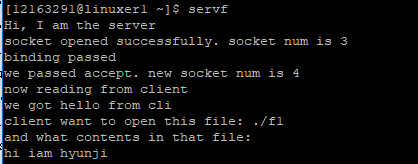
그냥 종료되는 것을 확인할 수 있습니다.

- halo (hello가 아닌 어떤 값이라도 ) 보낼경우

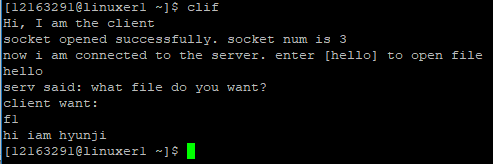


위와 같은 결과를 출력하며, 그냥 종료됩니다.

-hello를 보낼경우, -파일의 내용을 읽어올 경우



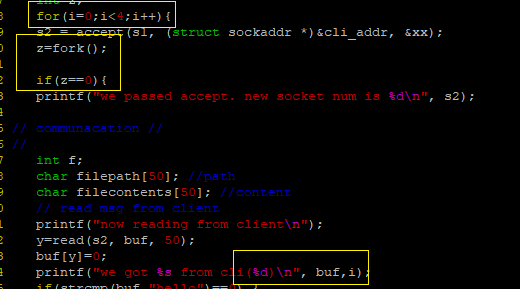
server



client

앞서 보여드렸던 f1파일의 내용을 잘 출력하는 것을 확인할 수 있었습니다.

5) Modify your ftp server( 4번문제) such that it can handle multiple clients at the same time.



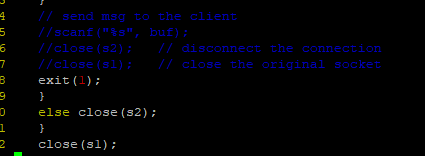
servf.c 파일입니다.

isten상태를 반복해야하므로, s1 accept 직후 fork()로 자식을 만들어줍니다.

이를 for문안에 넣어, 4개의 client를 받을 준비가 되어잇는 listen server 를 4개 만들어 줍니다.

이후 코드는 4)와 거의 동일하지만,

we got from cli 뒤에 i를 넣어줌으로, 현재 serv와 통신하고 있는 client의 구분을 해줍니다.



for문에서 accept로 client와 연결한 후, fork하여 child인 경우에 파일을 주고 받은 후,

자식이 또 자식을 생성하지 않도록 exit(1) 을 해줍니다.

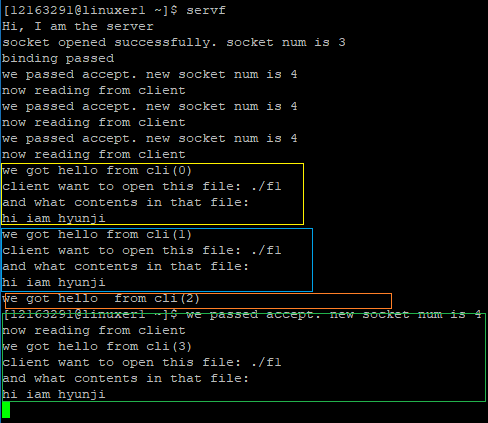
자식 과정 이후, close(s2) 해줍니다

s1(listen 소켓)은 다양한 clnt들의 요청을 받기 위해서 계속 열려있어야 하므로, for문이 끝난 이후 마지막에 close 해줍니다.



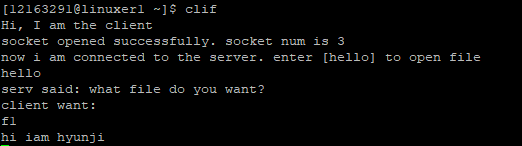
clif.c 파일의 경우,

파일이 없으면 서버에서 보낸 nofile을 인식하여, 프로그램을 종료해줄 수 있도록 하였습니다.

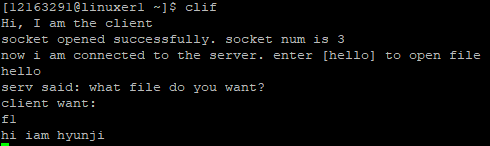


결과는 위와 같습니다.

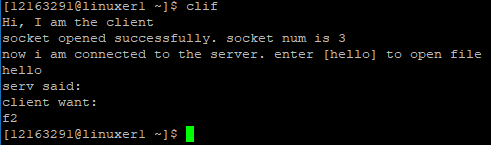
1. 노란 부분



2. 파란 부분

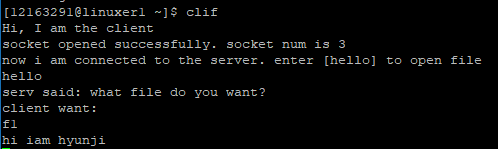


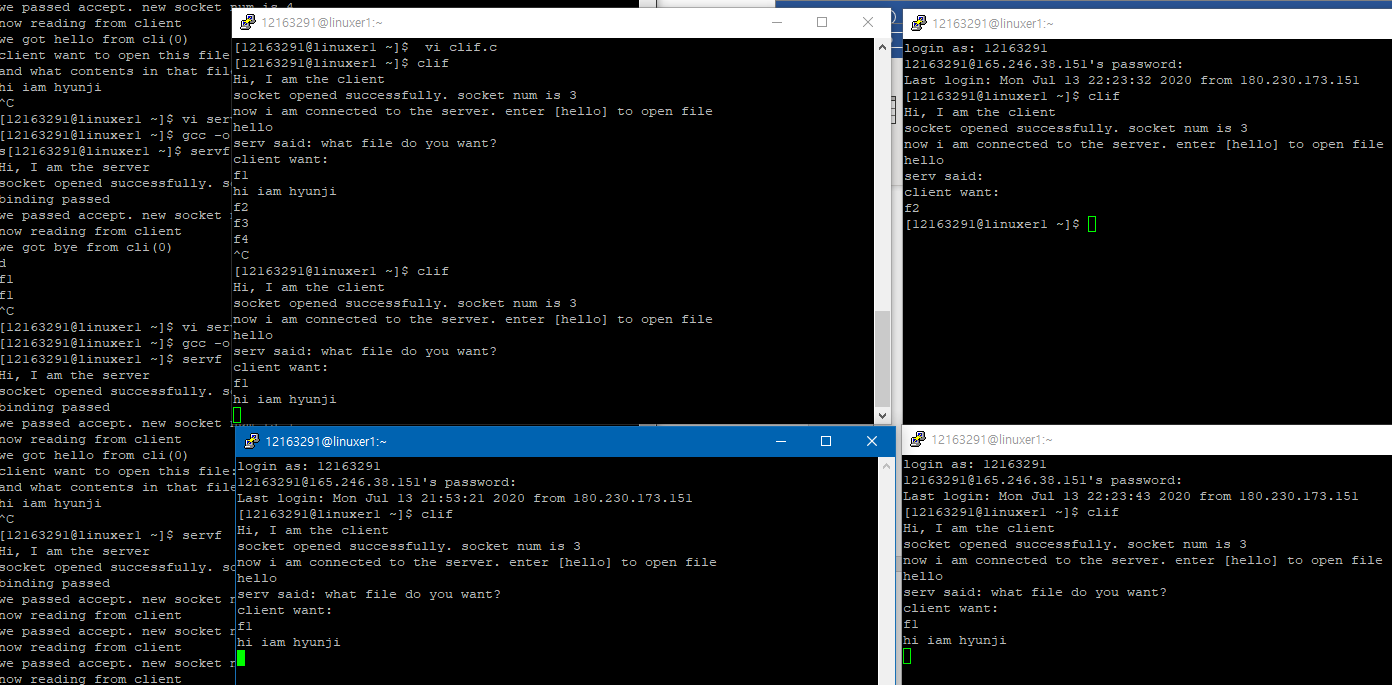
3. 주황부분



f2 파일이 없어, 내용을 읽어오지 못했습니다.

4. 초록부분





하나의 serv와, client 4개를 동시에 실행시키고 있는 모습입니다.

6) Write a client in your PC as follows and let it talk to the server program in the lab server. To compile the client program:

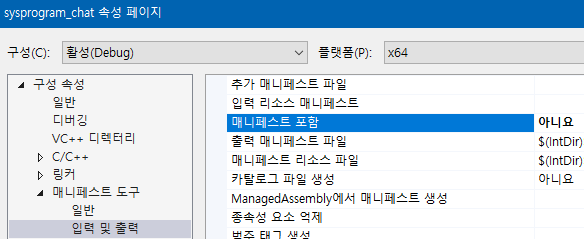
- Make an empty c++ project and copy the code given below.

- Adjust the server IP and port number

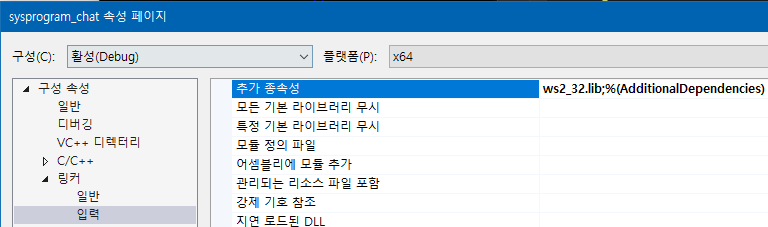
- Select

“project->project property->manifest tools->input and output->include manifest”

and set “No”



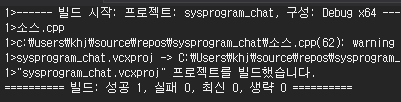
- add ws2\_32.lib in project>project property>link>input>additional dependencies



- Select build->Solution Build

 x64 중요!

- You should see “Success 1” at the bottom of the compile window.



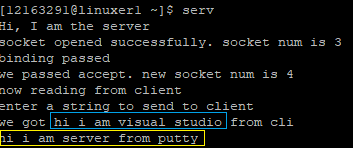
Visual studio 환경 설정을 위와 같이 해준 후,



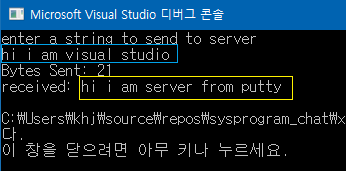
포트번호와 IP주소값을 알맞게 작성해줍니다

serv파일은

4,5) 에서 사용한 servf.c 가 아닌, serv.c 파일을 그대로- 사용하였습니다.



server putty 결과



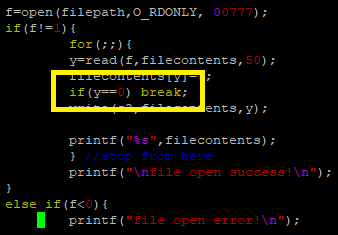
client visual studio 결과

결과들이 잘 출력된 것을 확인할 수 있었습니다.

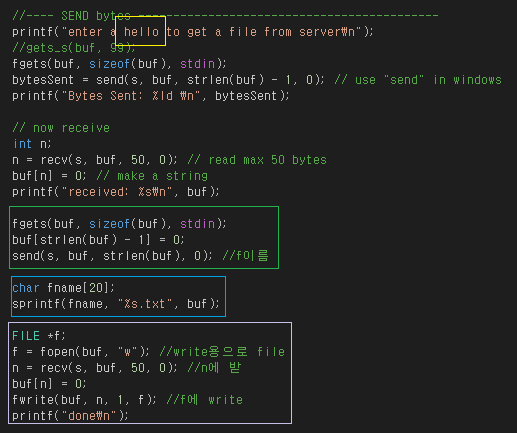
7) Write an ftp client in your PC and let it talk to the ftp server you made in problem 5). Use this client to download a file from the lab server.

FILE fopen 함수를 다뤄야 합니다.

servfcp.c 파일을 사용하였습니다



+ 실수로 넣지 않았던 if break문을 넣어 주었습니다



vs 코드입니다.

hello를 입력하라는 문구를 출력합니다.

hello를 bytesent – send로 server에 보내줍니다.

recv를 통해 내용을 받습니다.

초록부분 – 받고싶은 파일의 이름을 입력받아

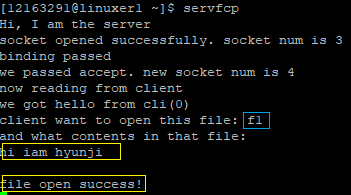
send를 통해 server에 보내줍니다.

파란부분 – sprintf를 통해 file의 이름 뒤에 .txt를 붙여줍니다.

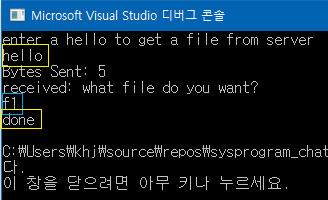
보라부분 – 파일내용을 받아 FILE, fopen(), fwrite()를 통해 파일로 만들어

텍스트 파일로 만들어 저장해줍니다.

완료되면, done을 출력합니다.



server putty 결과입니다.

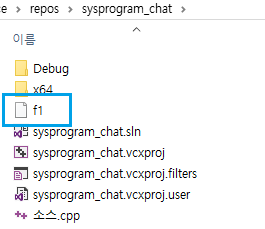
****

client visual studio결과 입니다

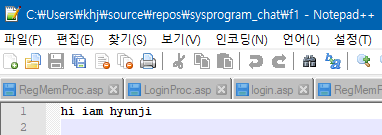
hello를 보내고, 파일이름을 보내,

파일이름에 해당하는 파일의 내용을 읽어 받아와,

.txt.파일로 저장합니다



텍스트 파일이 생성돼, 저장되어 있는 것을 확인할 수 있었습니다.



note pad를 이용해 내용을 확인해보니, f1의 내용이 잘 들어가 있는 것을 확인할 수 있었습니다.

\*\*server binding 실패시,

ps -ef로 PID 찾든 해서 kill해주면됨