|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **과 목** | 2020년 1학기 네트워크프로그래밍 | | | **주 차** | 13-1 주차 |
| **분 류** | 과제 |
| **학 과** | 전자통신공학과 | **학 번** | 2015707003 | **성 명** | 정승훈 |
| **제 출**  **방 법** | 1. 아래 과제를 잘 읽고 소켓 라이브러리를 사용하여 프로그램을 작성하세요.  2. Ubuntu OS, gcc compiler, vi 편집기를 사용하여 소스 코드를 작성하고 디버깅하세요.  3. 동작 화면의 스크린 캡처를 첨부하고,  4. 문제 아래에다 본인이 작성한 소스 코드를 복사하여 붙여넣으면 됩니다.  5. 소스 코드에 주석처리를 잘하시면 더 좋은 점수를 받을 수 있습니다.  6. 제출기한은 다음 주 월요일 2020년 6월 15일 자정까지 입니다. | | | | |

1. 온라인 동영상에 설명했던 FTP Command를 사용하여 FTP Test Server에 접속하여 서버에 있는 welcome.txt 텍스트 파일을 다운로드하여 화면에 표시하는 프로그램을 작성하세요.

파일명 : ftpclient.c

접속포트 : 21 외 Passive Listen 소켓

접속서버 : 223.194.7.95

접속아이디 : elec

접속패스워드 : comm

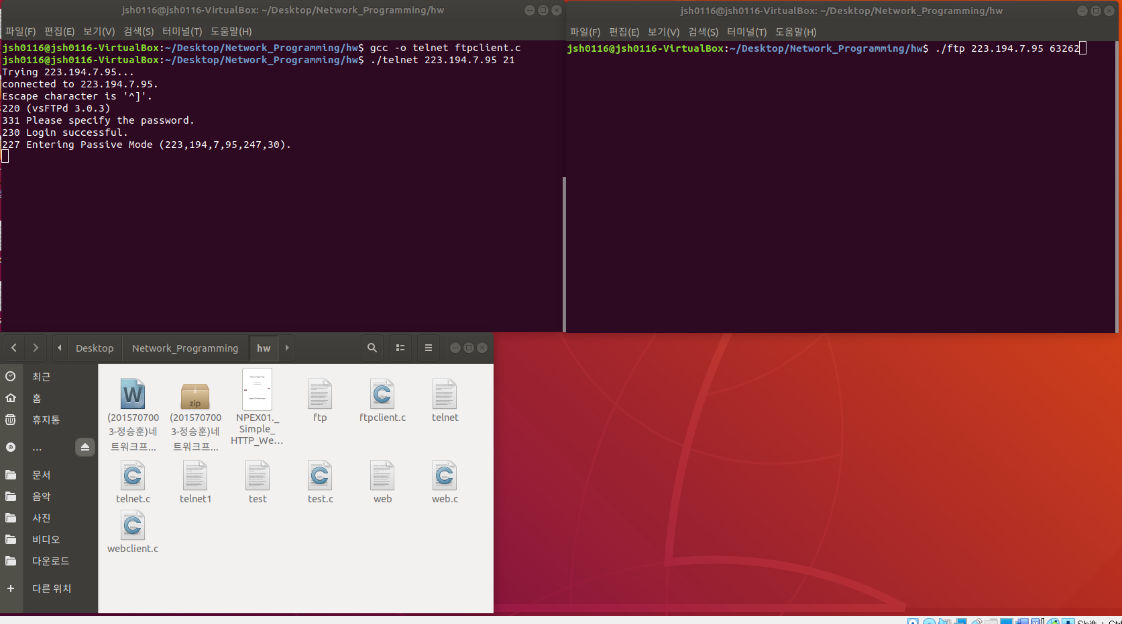
서버파일명 : welcome.txt

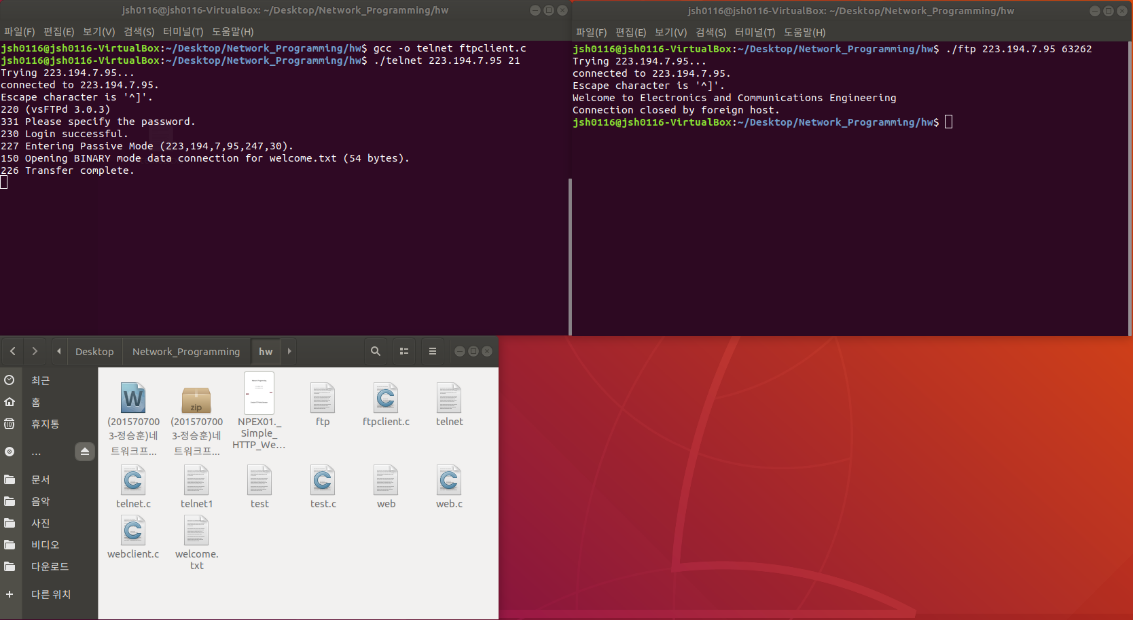
<Source Code>

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102 | /////////////////////////////////////////////////////////////////////////  // File Name       : ftpclient.c                                       //  // Date            : 2020/06/09 ~ 2020/06/15                           //  // OS              : Ubuntu 18.04.4 LTS                                //  // Student Name    : Seung Hoon Jeong                                  //  // Student ID      : 2015707003                                        //  // ------------------------------------------------------------------- //  // Title : Network Programming Assignment #13-1                        //  // Description : Simple FTP Client                                     //  /////////////////////////////////////////////////////////////////////////    /\* 필요한 헤더 파일 선언 \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <unistd.h>  #include <sys/socket.h>  #include <arpa/inet.h>  #include <netdb.h>  #include <netinet/in.h>  #define BUF\_SIZE 1024    ////////////////////////////////////////////////////////////////////////////////////////////////////  // Function : int main(int argc, char \*\*argv)                                                     //  // ============================================================================================== //  // Input: gcc -o telnet ftpclient.c,  ./[telnet] [223.194.7.95] [21]                              //  // Output: 331 Please specify the password., 230 Login successful.                                //  //         227 Entering Passive Mode (223,194,7,95,247,30).                                       //  //         150 Opening BINARY mode data connection for welcome.txt (54 bytes).                    //  //         226 Transfer complete.                                                                 //  // ============================================================================================== //  // Purpose: FTP Command Test using telnet command                                                 //  ////////////////////////////////////////////////////////////////////////////////////////////////////  int main(int argc, char \*\*argv)  {      int sockfd, n;      struct sockaddr\_in servaddr;        char buff[BUF\_SIZE] = {0};        /\* argument count exception handling \*/      if(argc != 3) {          printf("Usage : %s [URL] [PORT\_NUMBER]\n",argv[0]);          exit(0);      }        /\* open socket \*/      if((sockfd = socket(AF\_INET,SOCK\_STREAM, 0)) < 0) {          perror("sock");          exit(0);      }        memset(&servaddr,'\0',sizeof(servaddr)); // initialize server socket information struct to zero      servaddr.sin\_family = AF\_INET;      servaddr.sin\_addr.s\_addr = inet\_addr(argv[1]);      servaddr.sin\_port = htons(atoi(argv[2])); // short data(port number) to network byte order        printf("Trying %s...\n",argv[1]); // trying to enter ftp server      /\* connect socket \*/      if(connect(sockfd,(struct sockaddr\*)&servaddr, sizeof(servaddr)) < 0) {          perror("connect");          close(sockfd);          exit(1);      }      else printf("connected to %s.\n",argv[1]);      printf("Escape character is '^]'.\n");        bzero(buff,sizeof(buff));      n = read(sockfd,buff,BUF\_SIZE-1);      printf("%s",buff); // print 220 (vsFTPd 3.0.3) or welcome.txt file contents        /\* .txt 파일 내용 출력 될 때 \*/      if(!strcmp(buff,"Welcome to Electronics and Communications Engineering\n"))      {          FILE\* fp = fopen("welcome.txt", "w");          if (fp == NULL)              perror("file");          fwrite(buff, 1, n, fp); // write buffer contents          memset(buff, '\0', BUF\_SIZE);          fclose(fp);          close(sockfd);          printf("Connection closed by foreign host.\n");          return 0;      }      bzero(buff,sizeof(buff));      char msg[BUF\_SIZE] = "USER elec\r\nPASS comm\r\nPASV\r\nRETR welcome.txt\r\n";      write(sockfd, msg, sizeof(msg)); // send message        while(1)      {          read(sockfd, buff, BUF\_SIZE-1);          printf("%s", buff); // print result          bzero(buff,sizeof(buff));        }      /\* close socket \*/      close(sockfd);      printf("Connection closed by foreign host.\n");      return 0;  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

<Result>

1. ID,PASSWORD로그인, PASV입력, (247\*256) + 30 =63262 포트 접속



1. 파일 다운로드 완료
2. 다운로드 파일 내용 확인

