|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **과 목** | 2020년 1학기 네트워크프로그래밍 | | | **주 차** | 12-2 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 수요일 2020년 6월 10일 자정까지 입니다. | | | | |

1. 온라인에서 설명했던 HTTP Request Method를 사용하여 스마트시스템 연구실 홈페이지에 접속하여 메인 페이지 내용을 화면에 출력하는 리눅스 기반 소켓 프로그램을 작성하세요.

파일명 : webclient.c

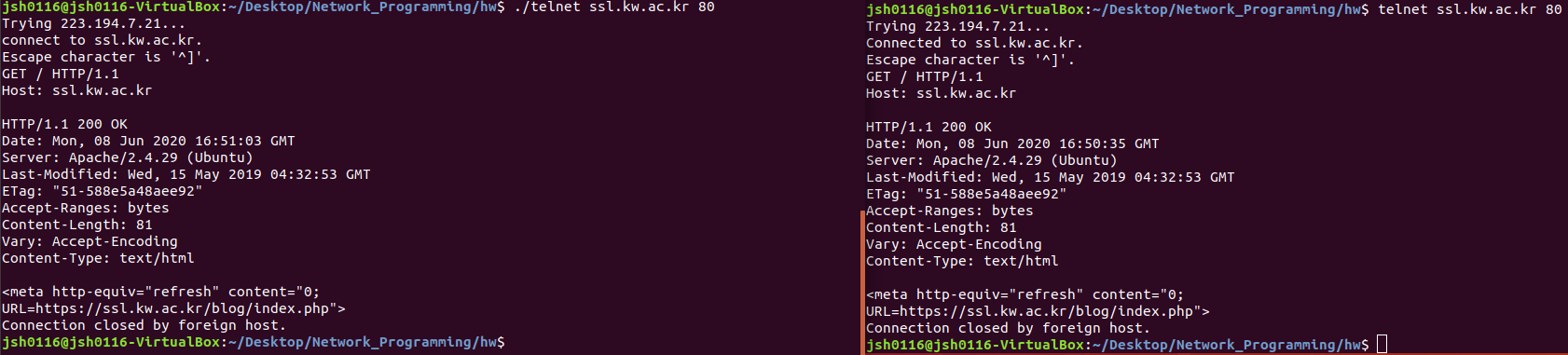
접속포트 : 80

접속서버 : ssl.kw.ac.kr

<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  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142 | /////////////////////////////////////////////////////////////////////////  // File Name       : webclient.c                                       //  // Date            : 2020/06/04 ~ 2020/06/10                           //  // OS              : Ubuntu 18.04.4 LTS                                //  // Student Name    : Seung Hoon Jeong                                  //  // Student ID      : 2015707003                                        //  // ------------------------------------------------------------------- //  // Title : Network Programming Assignment #12-2                        //  // Description : Simple HTTP Web Browser                               //  /////////////////////////////////////////////////////////////////////////    /\* 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 webclient.c ./[telnet] [ssl.kw.ac.kr] [80]                                //  // Output: 1. success HTTP/1.1 200 OK ...                                                         //  //         2. fail : HTTP/1.1 400 Bad Request ...                                                 //  // ============================================================================================== //  // Purpose: HTTP Command Test using telnet command                                                //  ////////////////////////////////////////////////////////////////////////////////////////////////////  int main(int argc, char \*\*argv)  {      int sockfd, n;      struct sockaddr\_in servaddr;        char first\_buff[BUF\_SIZE] = {0};      char second\_buff[BUF\_SIZE] = {0};      char third\_buff[BUF\_SIZE] = {0};      char buff[BUF\_SIZE] = {0};      char \*url = "ssl.kw.ac.kr";      struct hostent \*host;        /\* 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);      }        /\* get hostname \*/      if((host = gethostbyname(url)) < 0) {          perror("gethostbyname");          exit(0);      }        memset((char \*)&servaddr,'\0',sizeof(servaddr)); // initialize server socket information struct to zero      servaddr.sin\_family = AF\_INET;      servaddr.sin\_addr.s\_addr = \*(u\_long \*)host->h\_addr\_list[0]; // 4 byte address casting      servaddr.sin\_port = htons(atoi(argv[2])); // short data(port number) to network byte order        printf("Trying %s...\n",inet\_ntoa(servaddr.sin\_addr));      /\* connect socket \*/      if(connect(sockfd,(struct sockaddr\*)&servaddr, sizeof(servaddr)) < 0) {          perror("connect");          close(sockfd);          exit(1);      }      else printf("connect to %s.\n",host->h\_name);      printf("Escape character is '^]'.\n");        while(1)      {          ///////////////////text input///////////////////          read(STDIN\_FILENO, first\_buff, BUF\_SIZE);          /\* first line exception handling \*/          if(strcmp(first\_buff,"GET / HTTP/1.1\n")) {              write(sockfd,first\_buff,sizeof(first\_buff)); // 입력 틀리면 bad request 응답받기 위함              break;          }          else {              strcat(buff,first\_buff);              bzero(first\_buff,sizeof(first\_buff));          }          read(STDIN\_FILENO, second\_buff, BUF\_SIZE);          /\* second line exception handling \*/          if(strcmp(second\_buff,"Host: ssl.kw.ac.kr\n")) {              write(sockfd,second\_buff,sizeof(second\_buff));            }          else {              strcat(buff,second\_buff);              bzero(second\_buff,sizeof(second\_buff));            }          read(STDIN\_FILENO, third\_buff, BUF\_SIZE);          /\* third line exception handling \*/          if(strcmp(third\_buff,"\n")) {              write(sockfd,third\_buff,sizeof(third\_buff));              bzero(third\_buff,sizeof(third\_buff));          }          else {              strcat(buff,third\_buff);              bzero(third\_buff,sizeof(third\_buff));              break;          }      }        if(!strcmp(buff,"GET / HTTP/1.1\nHost: ssl.kw.ac.kr\n\n")) {          /\* Sending HTTP Request \*/          if(write(sockfd,"GET / HTTP/1.1\r\nHost:ssl.kw.ac.kr\r\n\r\n",sizeof("GET / HTTP/1.1\r\nHost: ssl.kw.ac.kr\r\n\r\n")) < 0) {              write(STDERR\_FILENO, "write error!!\n", sizeof("write error!!\n"));              exit(1);          }            /\* Receiving HTTP Reply (OK Request) \*/          if(read(sockfd,buff,BUF\_SIZE-1) < 0) {              write(STDERR\_FILENO, "read error!!\n", sizeof("read error!!\n"));              exit(0);          }          else printf("%s",buff);          sleep(4);      }        /\* Receiving HTTP Reply (Bad Request) \*/      if(read(sockfd,buff,BUF\_SIZE-1) < 0) {          write(STDERR\_FILENO, "read error!!\n", sizeof("read error!!\n"));          exit(0);      }      else printf("%s",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) |

<성공 결과 화면>



Webclient.c 소스를 이용한 request/reply와 실제 telnet command 명령어 request/reply비교

<예외 처리 결과 화면>

1. 첫번째 줄 잘못 입력했을 때



1. 두번째 줄 잘못 입력했을 때

