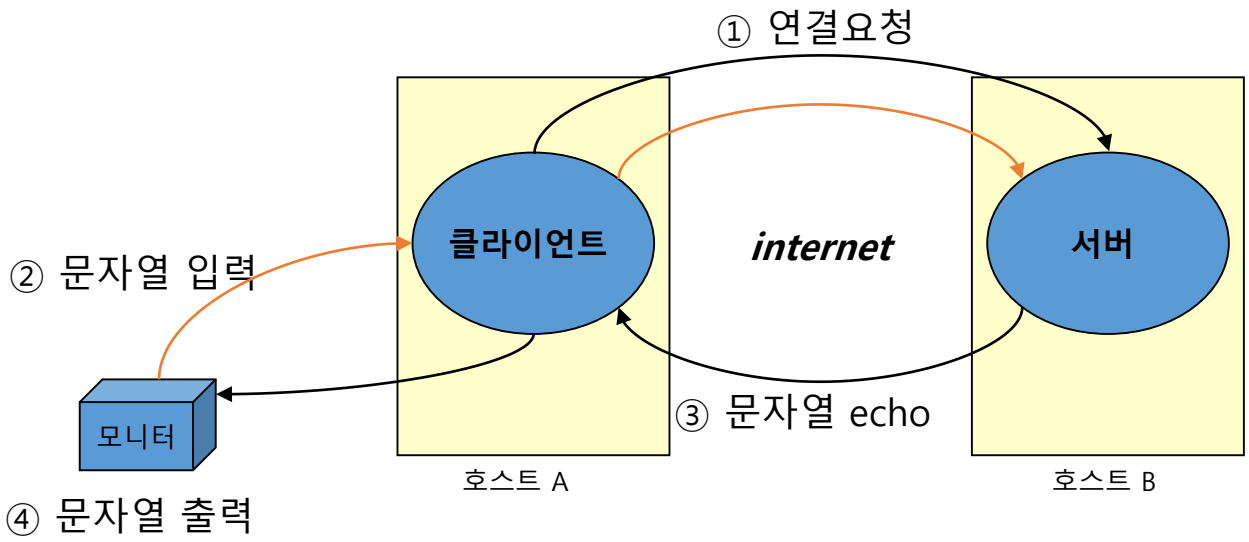
C언어로 구현하는 TCP/IP 소켓 프로그래밍

echo 프로그램



① 연결요청 : 서버 프로그램에 연결 요청 (클라이언트 -> 서버)

② 문자열 전송 : 사용자가 입력한 문자열 전송 (클라이언트 -> 서버)

③ 문자열 echo : 사용자가 보낸 문자열 echo (서버 -> 클라이언트)

④ 문자열 출력 : 문자열 화면 출력 (클라이언트 -> 모니터)

echo 프로그램 / 서버

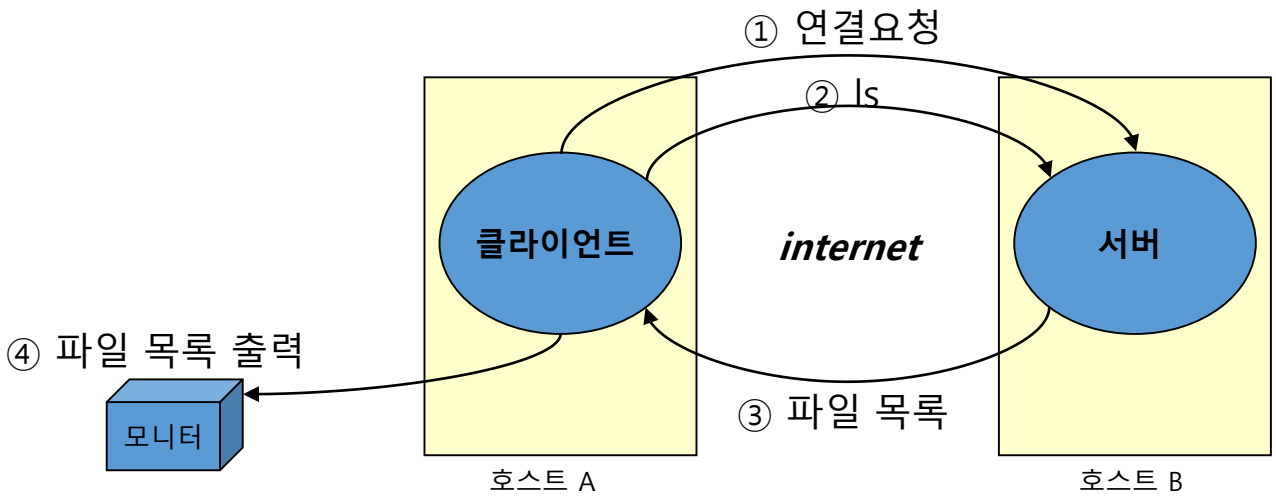
```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <sys/socket.h>
4.  #include <arpa/inet.h>
5.  #include <unistd.h>

6.  #define PORT 9000

7.  int main(void){
8.   int  s_socket, c_socket;
9.   struct sockaddr_in s_addr, c_addr;

10.  int n;
11.  int len;
12.  char rcvBuffer[BUFSIZ];

13.  s_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);   ①

14.  memset(&s_addr, 0, sizeof(s_addr));                    ②
15.  s_addr.sin_addr.s_addr = htonl(INADDR_ANY);
16.  s_addr.sin_family = AF_INET;
17.  s_addr.sin_port = htons(PORT);

17.  if(bind(s_socket, (struct sockaddr*)&s_addr, sizeof(s_addr)) == -1){   ③
18.    printf("Can not Bind!!!₩n");
19.    return -1;
20.  }

21.  if(listen(s_socket, 5) == -1){                         ④
22.    printf("Listen Fail!!!₩n");
23.    return -1;
24.  }

25.  while(1){
26.          printf("Echo Server started...₩n");
27.    len = sizeof(c_addr);                                ⑤
28.    c_socket = accept(s_socket, (struct sockaddr*)&c_addr, &len);
29.    printf("Connected IP : %s₩n", inet_ntoa(c_addr.sin_addr));

30.    while((n = read(c_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){   ⑥
31.        rcvBuffer[n] = '₩0';
32.        printf("%s", rcvBuffer);
33.        write(c_socket, rcvBuffer, n);
34.    }

35.    printf("client bye~~₩n");
36.    close(c_socket);                                     ⑦
37.  }
38.         close(s_socket);                                ⑧
39.  return 0;
40.  }
```

- s_socket : 클라이언트의 연결 요청을 처리하는 듣기소켓
- c_socket : 연결된 클라이언트의 소켓과 직접 통신하는 연결소켓

echo 프로그램 / 클라이언트

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <arpa/inet.h>
4.  #include <sys/socket.h>
5.  #include <unistd.h>

6.  #define PORT 9000
7.  #define IPADDR "127.0.0.1"

8.  int main(void){
9.      int c_socket;
10.     struct sockaddr_in s_addr;

11.     char sndBuffer[BUFSIZ], rcvBuffer[BUFSIZ];

12.     int n;

13.     c_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);   ①

14.     memset(&s_addr, 0, sizeof(s_addr));                     ②
15.     s_addr.sin_addr.s_addr = inet_addr(IPADDR);
16.     s_addr.sin_family = AF_INET;
17.     s_addr.sin_port = htons(PORT);

18.     if(connect(c_socket, (struct sockaddr*)&s_addr, sizeof(s_addr)) == -1){   ③
19.         printf("Can not connect!!!₩n");
20.         close(c_socket);
21.         return -1;
22.     }

23.     while(1){
24.         memset(sndBuffer, 0, BUFSIZ);                       ④
25.         printf("Input message to send to server.₩n");
26.         printf("if you want to quit, type quit.₩n");
27.         if((n = read(0, sndBuffer, BUFSIZ)) > 0){
28.             sndBuffer[n] = '₩0';
29.             if(!strcmp(sndBuffer, "quit₩n"))
30.                 break;

31.             printf("original Data : %s", sndBuffer);
32.             if((n = write(c_socket, sndBuffer, strlen(sndBuffer))) < 0){
33.                 return -1;
34.             }

35.             memset(rcvBuffer, 0, BUFSIZ);                   ⑤
36.             if((n = read(c_socket, rcvBuffer, BUFSIZ)) < 0){
37.                 return -1;
38.             }
39.
40.             printf("echoed Data : %s", rcvBuffer);
41.         }
42.     }
43.     close(c_socket);                                        ⑥
44.
45.     return 0;
46. }
```

서버 디렉토리의 파일 목록 출력



① 연결요청	: 서버 프로그램에 연결 요청	(클라이언트 -> 서버)
② ls 요청	: ls 전송	(클라이언트 -> 서버)
③ 파일 목록 전송	: 파일 목록 전송	(서버 -> 클라이언트)
④ 화면 출력	: 파일 목록 화면 출력	(클라이언트 -> 모니터)

디렉토리의 파일 목록 출력

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <dirent.h>

4. int main(void){
5.  DIR* dp;
6.  struct dirent* dir;

7.  if((dp = opendir(".")) == NULL){
8.    printf("directory open error₩n");
9.    exit(-1);
10.}
11.
12.while((dir = readdir(dp)) != NULL){
13.  if(dir->d_ino == 0)
14.      continue;
15.  printf("%s₩n", dir->d_name);
16.}

17.closedir(dp);

18.return 0;
19.}
```

서버 디렉토리의 파일 목록 출력 / 서버

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <arpa/inet.h>
4.  #include <sys/socket.h>
5.  #include <unistd.h>

6.  #include <dirent.h>

7.  #define PORT 9001

8.  int main(void){
9.   int s_socket, c_socket;
10.  struct sockaddr_in s_addr, c_addr;
11.  int len, n;
12. char rcvBuffer[BUFSIZ];
13. char err[] = "Directory Error";

14.  DIR* dp;
15.  struct dirent * dir;

16. s_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); ①

17. memset(&s_addr, 0, sizeof(s_addr));                   ②
18. s_addr.sin_addr.s_addr = htonl(INADDR_ANY);
19. s_addr.sin_family = AF_INET;
20. s_addr.sin_port = htons(PORT);

                                                          ③
21. if(bind(s_socket, (struct sockaddr*)&s_addr, sizeof(s_addr)) == -1){
22.    printf("Cannot Bind₩n");
23.    perror("Error Message");
24.    return -1;
25.  }
```

```
26.  if(listen(s_socket, 5) == -1){                        ④
27.     printf("Listen Fail!!!₩n");
28.     perror("Error Message");
29.     return -1;
30.  }

31.  while(1){
32.     printf("List Server Started...₩n");
33.     len = sizeof(c_addr);
34.     c_socket = accept(s_socket, (struct sockaddr*)&c_addr, &len);   ⑤

35.      printf("Connected IP : %s₩n", inet_ntoa(c_addr.sin_addr));

36.      if((n = read(c_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){    ⑥
37.                              rcvBuffer[n] = '₩0';
38.                              printf("%s₩n", rcvBuffer);
39.     }

40.     if(!strcmp(rcvBuffer, "ls")){
41.          if((dp = opendir(".")) == NULL){
42.          write(c_socket, err, strlen(err));
43.          }else{
44.               while((dir = readdir(dp)) != NULL){
45.                    if(dir->d_ino == 0)
46.                         continue;
47.                    write(c_socket, dir->d_name, strlen(dir->d_name));
48.                    write(c_socket, " ", 1);
49.               }
50.               closedir(dp);
51.          }
52.               }
53.     close(c_socket);                                     ⑦
54.  }
55.  close(s_socket);                                       ⑧
56.  return 0;
57.  }
```

서버 디렉토리의 파일 목록 출력 / 클라이언트

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <arpa/inet.h>
4.  #include <sys/socket.h>
5.  #include <unistd.h>
6.  #include <stdlib.h>

7.  #define PORT 9001
8.  #define IPADDR "127.0.0.1"

9.  char buffer[BUFSIZ];

10. int main(void){
11.     int c_socket;
12.     struct sockaddr_in s_addr;

13.     int n;
14.     char temp;

15.     c_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);   ①

16.     memset(&s_addr, 0, sizeof(s_addr));                     ②
17.     s_addr.sin_addr.s_addr = inet_addr(IPADDR);
18.     s_addr.sin_family = AF_INET;
19.     s_addr.sin_port = htons(PORT);

19. if(connect(c_socket, (struct sockaddr*)&s_addr, sizeof(s_addr)) == -1){   ③
20.         printf("Cannot connect₩n");
21.         perror("Error Message");
22.         return -1;
23.     }

24.     printf("Input command..₩n");                            ④
25.     scanf("%s", buffer);
26.     buffer[strlen(buffer)] = '₩0';
27.     if((n = write(c_socket, buffer, strlen(buffer)))   < 0){
28.         printf("Write error₩n");
29.         exit(-1);
30.     }

31.     printf("Received Data : ₩n");                           ⑤
32.     while((n = read(c_socket, &temp, 1)) > 0){
33.         printf("%c", temp);
34.         if(temp == ' ')
35.             printf("₩n");
36.     }
37.     close(c_socket);                                        ⑥
38.     return 0;
39. }
```

정리

- 정리
  - echo 프로그램
  - 서버 디렉토리 파일 목록 출력 프로그램

  ※ 구조는 변하지 않고 내용(처리)부분만 변화됨
    - 서버: socket(), bind(), listen(), accept(), read()/write(), close()
    - 클라이언트: socket(), connect(), read()/write(), close()
- 다음주…
  - Java와 이클립스가 설치 되어야 합니다.
    1. Java 설치(환경변수 포함)
    2. 이클립스 설치
    3. java와 이클립스 설치(동영상): https://www.youtube.com/watch?v=GRXhbbs6Go0