­­

과목명 : IOT시스템응용

담당교수 : 봉진숙

학번 : 20170660

학과 : 컴퓨터공학과

이름 : 김동현

**REPORT**

**1. echo\_server.c**

코드)

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <unistd.h>

#define PORT 9000

int main(void){

    int s\_socket, c\_socket;

    struct sockaddr\_in s\_addr, c\_addr;

    int n;

    int len;

    char rcvBuffer[BUFSIZ];

    s\_socket = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

    memset(&s\_addr, 0, sizeof(s\_addr));

    s\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

    s\_addr.sin\_family = AF\_INET;

    s\_addr.sin\_port = htons(PORT);

    if(bind(s\_socket, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1){

        printf("Can not Bind!!!\n");

        return -1;

    }

    if(listen(s\_socket, 5) == -1){

        printf("Listen Fail!!!\n");

        return -1;

    }

    while(1){

        printf("Echo Server started...\n");

        len = sizeof(c\_addr);

        c\_socket = accept(s\_socket, (struct sockaddr\*)&c\_addr, &len);

        printf("Connected IP : %s\n", inet\_ntoa(c\_addr.sin\_addr));

        while((n = read(c\_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){

            rcvBuffer[n] = '\0';

            printf("%s", rcvBuffer);

            write(c\_socket, rcvBuffer, n);

        }

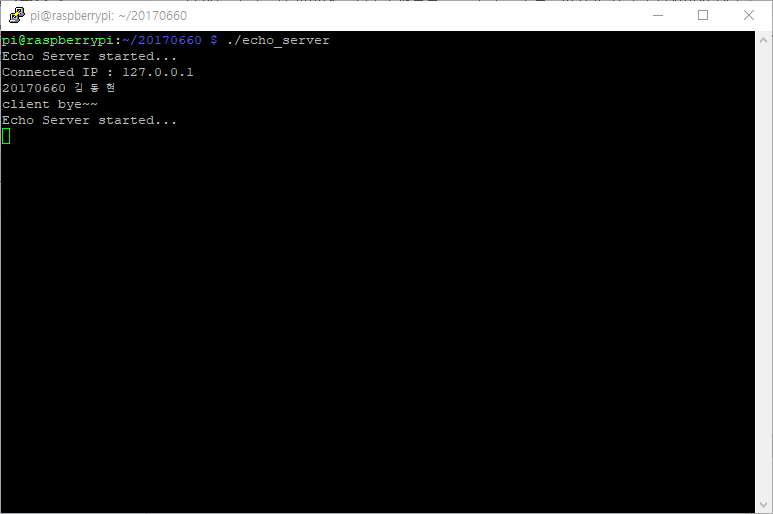
        printf("client bye~~\n");

        close(c\_socket);

    }

}

실행 결과)



**2. echo\_client.c**

코드)

#include <stdio.h>

#include <string.h>

#include <arpa/inet.h>

#include <sys/socket.h>

#include <unistd.h>

#define PORT 9000

#define IPADDR "127.0.0.1"

int main(void){

    int c\_socket;

    struct sockaddr\_in s\_addr;

    char sndBuffer[BUFSIZ], rcvBuffer[BUFSIZ];

    int n;

    c\_socket = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

    memset(&s\_addr, 0, sizeof(s\_addr));

    s\_addr.sin\_addr.s\_addr = inet\_addr(IPADDR);

    s\_addr.sin\_family = AF\_INET;

    s\_addr.sin\_port = htons(PORT);

    if(connect(c\_socket, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1){

        printf("Can not connect!!!\n");

        close(c\_socket);

        return -1;

    }

    while(1){

        memset(sndBuffer, 0, BUFSIZ);

        printf("Input message to send to server.\n");

        printf("if you want to quit, type quit\n");

        if((n = read(0, sndBuffer, BUFSIZ)) > 0){

            sndBuffer[n] = '\0';

            if(!strcmp(sndBuffer, "quit\n"))

                break;

            printf("original Data : %s", sndBuffer);

            if((n = write(c\_socket, sndBuffer, strlen(sndBuffer))) < 0){

                return -1;

            }

            memset(rcvBuffer, 0, BUFSIZ);

            if((n = read(c\_socket, rcvBuffer, BUFSIZ)) < 0){

                return -1;

            }

            printf("echoed Data : %s", rcvBuffer);

        }

    }

    close(c\_socket);

    return 0;

}

실행 결과)



**3. ls\_server.c**

코드)

#include <stdio.h>

#include <string.h>

#include <arpa/inet.h>

#include <sys/socket.h>

#include <unistd.h>

#include <dirent.h>

#define PORT 9001

int main(void){

    int s\_socket, c\_socket;

    struct sockaddr\_in s\_addr, c\_addr;

    int len, n;

    char rcvBuffer[BUFSIZ];

    char err[] = "Directory Error";

    DIR\* dp;

    struct dirent \* dir;

    s\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

    memset(&s\_addr, 0, sizeof(s\_addr));

    s\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

    s\_addr.sin\_family = AF\_INET;

    s\_addr.sin\_port = htons(PORT);

    if(bind(s\_socket, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1){

        printf("Cannot Bind\n");

        perror("Error Message");

        return -1;

    }

    if(listen(s\_socket, 5) == -1){

        printf("Listen Fail!!!\n");

        perror("Error Message");

        return -1;

    }

    while(1){

        printf("List Server Started...\n");

        len = sizeof(c\_addr);

        c\_socket = accept(s\_socket, (struct sockaddr\*)&c\_addr, &len);

        printf("Connected IP : %s\n", inet\_ntoa(c\_addr.sin\_addr));

        if((n == read(c\_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){

            rcvBuffer[n] = '\0';

            printf("%s\n", rcvBuffer);

        }

        if(!strcmp(rcvBuffer, "ls")){

            if((dp = opendir(".")) == NULL){

                write(c\_socket, err, strlen(err));

            }else{

                while((dir = readdir(dp)) != NULL){

                    if(dir->d\_ino == 0)

                        continue;

                    write(c\_socket, dir->d\_name, strlen(dir->d\_name));

                    write(c\_socket, " ", 1);

                }

                closedir(dp);

            }

        }

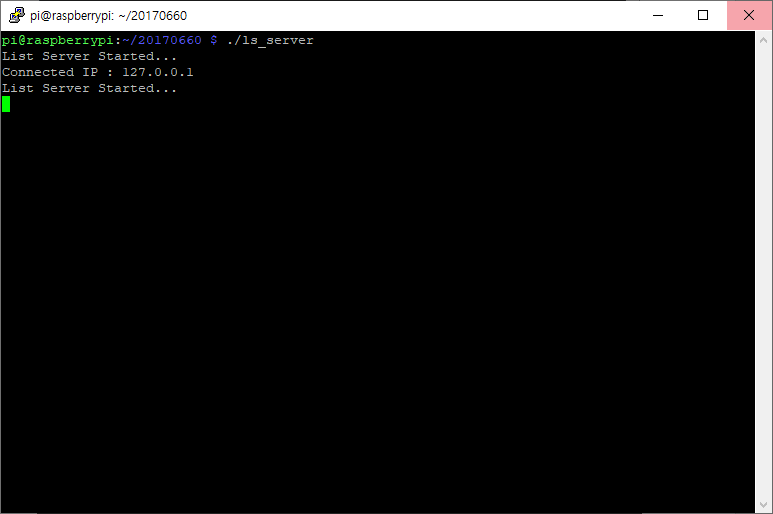
        close(c\_socket);

    }

    close(s\_socket);

}

실행 결과)



**4. ls\_client.c**

코드)

#include <stdio.h>

#include <string.h>

#include <arpa/inet.h>

#include <sys/socket.h>

#include <unistd.h>

#include <stdlib.h>

#define PORT 9001

#define IPADDR "127.0.0.1"

char buffer[BUFSIZ];

int main(void){

    int c\_socket;

    struct sockaddr\_in s\_addr;

    int n;

    char temp;

    c\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

    memset(&s\_addr, 0, sizeof(s\_addr));

    s\_addr.sin\_addr.s\_addr = inet\_addr(IPADDR);

    s\_addr.sin\_family = AF\_INET;

    s\_addr.sin\_port = htons(PORT);

    if(connect(c\_socket, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1){

        printf("Cannot connect\n");

        perror("Error Message");

        return -1;

    }

    printf("Input command..\n");

    scanf("%s", buffer);

    buffer[strlen(buffer)] = '\0';

    if((n = write(c\_socket, buffer, strlen(buffer))) < 0){

        printf("Write error\n");

        exit(-1);

    }

    printf("Received Data : \n");

    while((n = read(c\_socket, &temp, 1)) > 0){

        printf("%c", temp);

        if(temp == ' ')

            printf("\n");

    }

    close(c\_socket);

    return 0;

}

실행 결과)

