#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

int main(int argc, char \*argv[]){

// define server socket

int server\_socket;

server\_socket = socket(AF\_INET, SOCK\_STREAM, 0);

//define server address

struct sockaddr\_in server\_address;

server\_address.sin\_family = AF\_INET;

server\_address.sin\_addr.s\_addr = INADDR\_ANY;

server\_address.sin\_port = htons(64000);

//bind socket

bind(server\_socket, (struct sockaddr\*)&server\_address, sizeof(server\_address));

listen(server\_socket, 5);

printf("\nserver started listening\n");

while(1){

int client\_socket;

client\_socket = accept(server\_socket, NULL, NULL);

char \*msg = "Welcome, this is server Mohit";

write(client\_socket, msg, strlen(msg));

printf("\nmessage sent...");

char buff[256];

read(client\_socket, buff, sizeof(buff));

printf("\nmessage received from client: %s\n", buff);

char \*p = strstr(buff, "END");

if(p != NULL){

printf("\nserver is going to off\n");

break;

}

}

close(server\_socket);

}

// read and write v function

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <fcntl.h>

int main(int argc, char \*argv[])

{

int BUFFER\_SIZE=80;

//define server socket

int client\_socket;

client\_socket=socket(AF\_INET,SOCK\_STREAM,0);

//define server address

struct sockaddr\_in server\_address;

server\_address.sin\_family = AF\_INET;

server\_address.sin\_addr.s\_addr = INADDR\_ANY;

server\_address.sin\_port = htons(64000);

int connection\_status=connect(client\_socket,(struct sockaddr\*)&server\_address,sizeof(server\_address));

if(connection\_status==-1)

{

printf("\nError Connecting to Server\n");

exit(1);

}

char buff[256];

read(client\_socket, buff, sizeof(buff));

printf("Message received from Mohit's server : %s\n",buff);

char mybuf1[BUFFER\_SIZE], mybuf2[BUFFER\_SIZE];

struct iovec iov[2];

iov[0].iov\_base = mybuf1;

iov[0].iov\_len = BUFFER\_SIZE;

iov[1].iov\_base = mybuf2;

iov[1].iov\_len = BUFFER\_SIZE;

while (1) {

ssize\_t n = readv(STDIN\_FILENO, iov, 2);

if (n < 0) {

fprintf(stderr, "Failed to read input from stdin: %s\n", strerror(errno));

return 1;

}

if (n == 0) {

break;

}

if (writev(client\_socket, iov, 2) < 0) {

fprintf(stderr, "Failed to write to socket: %s\n", strerror(errno));

return 1;

}

}

close(client\_socket); }

