学号：202021090107 姓名：熊静 作业名：第二次作业

作业要求：

客户端：

从命令行读入服务器的IP地址；并连接到服务器；

循环从命令行读入一行字符串，并传递给服务器，由服务器对字符串反转，并将结果返回客户程序；客户程序显示反转后的字符串；服务器端：

接收客户的连接请求，并显示客户的IP地址和端口号；

客户端程序为client2.c文件，

服务器程序为server2.c文件。

client2.c：

#include <stdio.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <string.h>

#include <arpa/inet.h>

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

if(argc<2){

printf("usage: %s <server address>\n", argv[0]);

return -1;

}

int sockfd;

struct sockaddr\_in server\_addr;

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

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

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

server\_addr.sin\_port = htons(9999);

if((inet\_pton(AF\_INET, argv[1], &server\_addr.sin\_addr)) < 0)

perror("invalid IP address\n");

if(connect(sockfd, (struct sockaddr\*)&server\_addr, sizeof(server\_addr)) < 0)

perror("can't connect to server\n");

char sends[1024], recvs[1024];

memset(sends, 0, sizeof(sends));

memset(recvs, 0, sizeof(recvs));

while (1)

{

printf("Please enter a string that you want to send to the server: ");

fgets(sends, sizeof(sends), stdin);

if(send(sockfd, sends, strlen(sends)-1, 0) == -1)

perror("send error\n");

if(strcmp(sends, "quit\n") == 0 || strcmp(sends, "exit\n") == 0){

close(sockfd);

break;

}

if(recv(sockfd, recvs, sizeof(recvs), 0) == -1)

perror("recv error\n");

fputs(recvs, stdout);

memset(sends, 0, sizeof(sends));

memset(recvs, 0, sizeof(recvs));

printf("\n");

}

close(sockfd);

return 0;

}

server2.c：

#include <stdio.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <string.h>

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

int listenfd, connfd;

socklen\_t clilen;

struct sockaddr\_in server\_addr, client\_addr;

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

int opt = SO\_REUSEADDR;

setsockopt(listenfd, SOL\_SOCKET, SO\_REUSEADDR, &opt, sizeof(opt));

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

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

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

server\_addr.sin\_port = htons(9999);

if(bind(listenfd, (struct sockaddr\*)&server\_addr, sizeof(server\_addr)) < 0)

perror("bind error\n");

if(listen(listenfd, 10) < 0)

perror("listen error\n");

char buff[1024];

memset(buff, 0, sizeof(buff));

while(1){

clilen = sizeof(client\_addr);

if((connfd = accept(listenfd, (struct sockaddr\*)&client\_addr, &clilen)) < 0)

perror("accept error\n");

printf("client address: %s, client port: %d\n", inet\_ntoa(client\_addr.sin\_addr), client\_addr.sin\_port);

while(1){

if(recv(connfd, buff, sizeof(buff), 0) == -1)

perror("recv error\n");

if(strcmp(buff, "quit") == 0 || strcmp(buff, "exit") == 0){

close(connfd);

break;

}

int len = strlen(buff);

for(int i = 0; i < len/2; i++){

char tmp = buff[i];

buff[i] = buff[len-1 - i];

buff[len-1 - i] = tmp;

}

if(send(connfd, buff, strlen(buff), 0) == -1)

perror("send error\n");

memset(buff, 0, sizeof(buff));

}

close(connfd);

}

close(listenfd);

}

运行截图：