# **网络编程第二次作业**

**姓名：杨庆 学号：201822090316**

# 1.作业题目

（1）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。

（2）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；

接收客户传来的字符串，反转后传递给客户。

# 2.实验步骤及结果

2.1 编译

在程序所在目录下，打开终端程序，输入指令如下。

gcc client.c -o client

gcc server.c -o server

2.2运行

运行服务器程序，在刚才编译的窗口内输入指令：./server。

运行客户端程序。新打开终端，输入以下命令：./client 127.0.0.1

3.3.结果

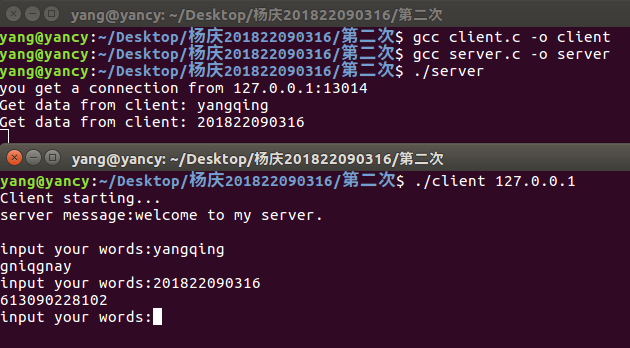
客户端显示服务器已连接，并提示输入字符串。

服务器端显示所连接的客户端的IP地址和端口号。

客户端：输入字符串，并传递给服务器，将服务器送来的反转后字符串输出。

服务器：显示从客户端得到的字符串，并进行反转，反转后返回客户端。

如下图所示。



# 3.客户端源代码

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#define PORT 6789

#define MAXDATASIZE 1000

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

int connfd,numbytes,scan\_state;

char buffer[MAXDATASIZE];

struct hostent \*host;

struct sockaddr\_in server;

printf("Client starting...\n");

if(argc!=2){

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

exit(-1);

}

if((host=gethostbyname(argv[1]))==NULL){

perror("Get host by name failed.\n");

exit(1);

}

if((connfd=socket(AF\_INET,SOCK\_STREAM,0))==-1){

perror("Create socket failed.\n");

exit(1);

}

bzero(&server,sizeof(server));

server.sin\_family=AF\_INET;

server.sin\_port=htons(PORT);

server.sin\_addr=\*((struct in\_addr \*)host->h\_addr);

if(connect(connfd,(struct sockaddr \*)&server,sizeof(struct sockaddr))==-1){

perror("Connect failed.\n");

exit(1);

}

if(((numbytes=recv(connfd,buffer,MAXDATASIZE,0))==-1)){

perror("Receive data failed.\n");

exit(1);

}

buffer[numbytes]='\0';

printf("server message:%s\n",buffer);

while(1){

printf("input your words:");

if((scan\_state=scanf("%s",buffer))==-1){

printf("end\n");

break;

}

numbytes=strlen(buffer);

send(connfd,buffer,numbytes,0);

if(((numbytes=recv(connfd,buffer,MAXDATASIZE,0))==-1)){

perror("Receive data failed.\n");

exit(1);

}

buffer[numbytes]='\0';

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

}

close(connfd);

}

# 4.服务器源代码

#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdlib.h>

#define PORT 6789

#define BACKLOG 1

#define MAXDATASIZE 1000

void inverse\_str(char\* str,int len){

char temp;

char \*strLeft = str;

char \*strRight = str + len - 1;

while(strRight > strLeft)

{

char temp = \*strLeft;

\*strLeft = \*strRight;

\*strRight = temp;

--strRight;

++strLeft;

}

}

int main(void){

int listenfd,connectfd,numbytes;

struct sockaddr\_in server,client;

socklen\_t sin\_size;

char buffer[MAXDATASIZE];

char name[20];

pid\_t pid;

if((listenfd=socket(AF\_INET,SOCK\_STREAM,0))==-1){

perror("Create socket failed.\n");

exit(-1);

}

int opt=SO\_REUSEADDR;

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

bzero(&server,sizeof(server));

server.sin\_family=AF\_INET;

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

server.sin\_port=htons(PORT);

if(bind(listenfd,(struct sockaddr \*)&server,sizeof(struct sockaddr))==-1){

perror("Bind socket failed.\n");

exit(-1);

}

if(listen(listenfd,BACKLOG)==-1){

perror("Listen socket failed.\n");

exit(-1);

}

sin\_size=sizeof(struct sockaddr\_in);

while(1){

if((connectfd=accept(listenfd,(struct sockaddr \*)&client,&sin\_size))==-1){

perror("Accept failed.\n");

exit(-1);

}

printf("you get a connection from %s:%d\n",inet\_ntoa(client.sin\_addr),client.sin\_port);

send(connectfd,"welcome to my server.\n",22,0);

while(1){

if(((numbytes=recv(connectfd,buffer,MAXDATASIZE,0))==-1)){

perror("Receive data of name of client error.\n");

exit(1);

}else if(numbytes==0){

printf("connection end\n");

break;

}

buffer[numbytes]='\0';

printf("Get data from client: %s\n", buffer);

inverse\_str(buffer,numbytes);

send(connectfd,buffer,numbytes,0);

}

close(connectfd);

}

close(listenfd);

}