#include <stdio.h>

#include <sys/socket.h>

#include <unistd.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <time.h>

#define SERVER\_PORT 20000 // define the defualt connect port id

#define LENGTH\_OF\_LISTEN\_QUEUE 10 //length of listen queue in server#define BUFFER\_SIZE 255

#define WELCOME\_MESSAGE "welcome to connect the server. "

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

int servfd,clifd;

struct sockaddr\_in servaddr,cliaddr;

if ((servfd = socket(AF\_INET,SOCK\_STREAM,0)) < 0)

{

printf("create socket error!/n");

exit(1);

}

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(SERVER\_PORT);

servaddr.sin\_addr.s\_addr = htons(INADDR\_ANY);

if (bind(servfd,(struct sockaddr\*)&servaddr,sizeof(servaddr))<0)

{

printf("bind to port %d failure!/n",SERVER\_PORT);

exit(1);

}

if (listen(servfd,LENGTH\_OF\_LISTEN\_QUEUE) < 0)

{

printf("call listen failure!/n");

exit(1);

}

while (1)

{

//server loop will nerver exit unless any body kill the process

char buf[BUFFER\_SIZE];

long timestamp;

socklen\_t length = sizeof(cliaddr);

clifd = accept(servfd,(struct sockaddr\*)&cliaddr,&length);

if (clifd < 0)

{

printf("error comes when call accept!/n");

break;

}

strcpy(buf,WELCOME\_MESSAGE); //inet\_ntop(INET\_ADDRSTRLEN,cliaddr.sin\_addr,buf,BUFFER\_SIZE);

printf("from client,IP:%s,Port:%d/n",inet\_ntoa(cliaddr.sin\_addr),ntohs(cliaddr.sin\_port));

timestamp = time(NULL);

strcat(buf,"timestamp in server:");

strcat(buf,ctime(&timestamp));

send(clifd,buf,BUFFER\_SIZE,0);

close(clifd);

}//exit

close(servfd);

return 0;

}

#include <stdio.h>

#include <sys/socket.h>

#include <unistd.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <stdlib.h>

#define SERVER\_PORT 20000 // define the defualt connect port id

#define CLIENT\_PORT ((20001+rand())%65536) // define the defualt client port as a random port

#define BUFFER\_SIZE 255#define REUQEST\_MESSAGE "welcome to connect the server./n"

void usage(char \*name){

printf("usage: %s IpAddr/n",name);

}

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

int servfd,clifd,length = 0;

struct sockaddr\_in servaddr,cliaddr;

socklen\_t socklen = sizeof(servaddr);

char buf[BUFFER\_SIZE];

if (argc < 2)

{

usage(argv[0]);

exit(1);

}

if ((clifd = socket(AF\_INET,SOCK\_STREAM,0)) < 0)

{

printf("create socket error!/n");

exit(1);

}

srand(time(NULL));//initialize random generator

bzero(&cliaddr,sizeof(cliaddr));

cliaddr.sin\_family = AF\_INET;

cliaddr.sin\_port = htons(CLIENT\_PORT);

cliaddr.sin\_addr.s\_addr = htons(INADDR\_ANY);

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

inet\_aton(argv[1],&servaddr.sin\_addr);

servaddr.sin\_port = htons(SERVER\_PORT);

//servaddr.sin\_addr.s\_addr = htons(INADDR\_ANY);

if (bind(clifd,(struct sockaddr\*)&cliaddr,sizeof(cliaddr))<0)

{

printf("bind to port %d failure!/n",CLIENT\_PORT);

exit(1);

}

if (connect(clifd,(struct sockaddr\*)&servaddr, socklen) < 0)

{

printf("can't connect to %s!/n",argv[1]);

exit(1);

}

length = recv(clifd,buf,BUFFER\_SIZE,0);

if (length < 0)

{

printf("error comes when recieve data from server %s!",argv[1]);

exit(1);

}

printf("from server %s :/n/t%s ",argv[1],buf);

close(clifd);

return 0;

}