

**计算机学院实验报告**

班 级 网络工程1801 姓 名 吴斌 学 号 18408020129

时 间 2020.11.1 评 分 教师签名

课程名称  **网络程序设计**

1、实验名称； 流式套接字编程

2、实验目的；

通过在linux操作系统平台下,用C语言进行流式套接字网络编程,熟练掌握linux下的流式套接字的基本编程方法和客户端/服务器程序的结构.

3、实验原理；

1.流式套接字的通信特点

2.流式套接字api

3.创建套接字函数socket

4.连接请求函数connect

5.绑定本地地址函数bind

6.监听函数listen

7.接受请求函数accept

8.套接字i/o操作

9.流式套接字的通信过程

4、实验内容与结果；

场景一server.c

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<errno.h>

#include<string.h>

#include<netdb.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<arpa/inet.h>

#define BUFSIZE 512

static void bail(const char \*on\_what)

{

fputs(strerror(errno),stderr);

fputs(":",stderr);

fputs(on\_what,stderr);

fputc('\n',stderr);

exit(1);

}

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

{

int sockfd;

int new\_fd;

struct sockaddr\_in server\_addr;

struct sockaddr\_in client\_addr;

socklen\_t size;

int portnumber;

char reqBuf[BUFSIZE];

int z;

if(argc!=2){

fprintf(stderr,"错误\n");

exit(1);

}

if((portnumber = atoi(argv[1]))<0){

fprintf(stderr,"错误\n");

exit(1);

}

if((sockfd = socket(PF\_INET,SOCK\_STREAM,0))==-1){

fprintf(stderr,"Socket error:%s\a\n",strerror(errno));

exit(1);

}

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(portnumber);

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

fprintf(stderr,"Bind error:%s\a\n",strerror(errno));

exit(1);

}

if(listen(sockfd,128)==-1){

fprintf(stderr,"Listen error:%s\a\n",strerror(errno));

exit(1);

}

printf("waiting for the client's request...\n");

while(1){

size = sizeof(struct sockaddr\_in);

if((new\_fd = accept(sockfd,(struct sockaddr\*)(&client\_addr),&size))==-1){

fprintf(stderr,"Accept error:%s\a\n",strerror(errno));

exit(1);

}

fprintf(stdout,"Server got connection from %s\n",inet\_ntoa(client\_addr.sin\_addr));

for( ; ;){

z=read(new\_fd,reqBuf,sizeof(reqBuf));

printf("the length of the String is %d\n",z);

if(z<0)

bail("read()");

if(z=0){

close(new\_fd);

break;

}

reqBuf[z]=0;

int num=0;

int j;

for(j=0;j<z;j++){

if(reqBuf[j]<='z'&&reqBuf[j]>='a' || reqBuf[j]<='Z'&&reqBuf[j]>='A')

num++;

}

z = write(new\_fd,&num,sizeof(int));

printf("the number of the english letters is:%d\n",num);

if(z<0)

bail("write()");

}

}

}

场景二client.c

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<errno.h>

#include<string.h>

#include<netdb.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<arpa/inet.h>

#define BUFSIZE 512

static void bail(const char \*on\_what)

{

fputs(strerror(errno),stderr);

fputs(":",stderr);

fputs(on\_what,stderr);

fputc('\n',stderr);

exit(1);

}

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

{

int sockfd;

char buf[BUFSIZE];

struct sockaddr\_in server\_addr;

struct hostent \*host;

int portnumber;

int nbytes;

int z;

char reqBuf[BUFSIZE];

if(argc!=3){

fprintf(stderr,"usage:%s hostname portnumber\a\n",argv[0]);

exit(1);

}

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

fprintf(stderr,"gethostname error\n");

exit(1);

}

if((portnumber = atoi(argv[2]))<0){

fprintf(stderr,"usage:%s hostname portnumber\a\n",argv[0]);

exit(1);

}

if((sockfd = socket(PF\_INEF,SOCK\_STREAM,0))==-1){

fprintf(stderr,"Socket error:%s\a\n",strerror(errno));

exit(1);

}

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

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

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

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

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

fprintf(stderr,"connect error:%s\a\n",strerror(errno));

exit(1);

}

printf("connected to server %s\n",inet\_ntoa(server\_addr.sin\_addr));

for( ; ; ){

fputs("\nEnter a string(^D or 'quit' to exit):",stdout);

if(!fgets(reqBuf,sizeof(reqBuf),stdin)){

printf("\n");

break;

}

z = strlen(reqBuf);

if(z>0&&reqBuf[--z]=='\n')

reqBuf[z] = 0;

if(z==0)

continue;

if(!strcasecmp(reqBuf,"QUIT")){

printf("press any key to end client.\n");

getchar();

break;

}

z = write(sockfd,reqBuf,strlen(reqBuf));

printf("client has sent '%s' to the server\n",reqBuf);

if(z<0)

bail("write()");

if((nbytes = read(sockfd,buf,sizeof(buf)))==-1){

fprintf(stderr,"read error:%s\n",strerror(errno));

exit(1);

}

if(nbytes==0){

printf("server has close the socket.\n");

printf("press any key to exit...\n");

getchar();

break;

}

buf[nbytes]='\0';

printf("result from %s port %u: the number of english letters is %d\n",inet\_ntoa(server\_addr.sin\_addr),(unsigned)ntohs(server\_addr.sin\_port),\*(int \*)(buf));

}

close(sockfd);

return 0;

}

5、总结与讨论；

通过在linux操作系统平台下,用C语言进行了流式套接字网络编程,熟练掌握了linux下的流式套接字的基本编程方法和客户端/服务器程序的结构.