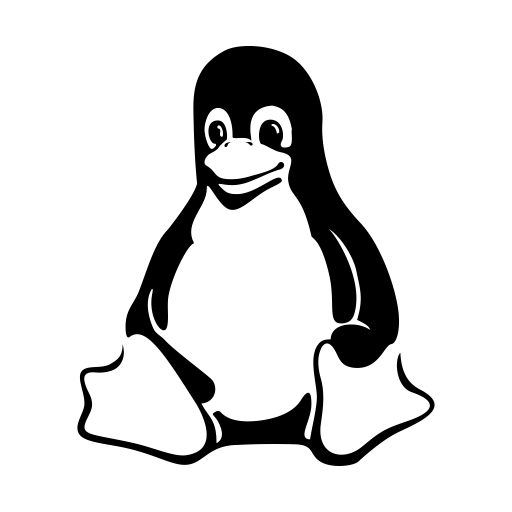
Linux高级程序设计

实验作业8



学号：0x091395

姓名：uknowho

年级：2014

班级：软件工程（3）班

# 思考与实验：

1. 在Linux系统下编写一个socket程序，要求服务端等待客户的连接请求.一旦有客户连接,服务器端打印出客户端的IP地址和端口,并且向服务器端发送欢迎信息和时间。

**echo\_client.c**

#include<stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <string.h>

#include<netdb.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#define SERVPORT 3333

#define MAXDATASIZE 500 /\*每次最大数据传输量 \*/

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

{

int sockfd,recvbytes;

char buf[MAXDATASIZE];

struct hostent \*host;

struct sockaddr\_in serv\_addr;

if(argc<2)

{

fprintf(stderr,"Please enter the sever's hostname!\n");

exit(1);

}

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

perror("gethostbyname error!");

exit(1);

}

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

perror("socket create error");

exit(1);

}

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

serv\_addr.sin\_port=htons(SERVPORT);

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

bzero(&(serv\_addr.sin\_zero),8);

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

{

perror("connet error!");

exit(1);

}

if((recvbytes=recv(sockfd,buf,MAXDATASIZE,0))==-1){

perror("connect出错");

exit(1);

}

buf[recvbytes]='\0';

printf("服务器端:%s\n",buf);

close(sockfd);

}

**echo\_server.c**

#include<stdio.h>

#include<stdlib.h>

#include<errno.h>

#include <string.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <sys/socket.h>

#include <sys/wait.h>

#define SERVPORT 3333 /\*服务器监听端口号 \*/

#define BACKLOG 10 /\* 最大同时连接请求数 \*/

int main()

{

int sockfd,client\_fd;

struct sockaddr\_in my\_addr; /\* 本机地址信息 \*/

struct sockaddr\_in remote\_addr; /\* 客户端地址信息 \*/

int sin\_size;

char buff[516];

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

{

perror("socket创建失败！");

exit(1);

}

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

/\* 把16位值从主机字节序转换成网络字节序\*/

my\_addr.sin\_port=htons(SERVPORT);

my\_addr.sin\_addr.s\_addr = INADDR\_ANY;

bzero(&(my\_addr.sin\_zero),8);

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

{

perror("bind出错");

exit(1);

}

if(listen(sockfd,BACKLOG)==-1)

{

perror("listen出错");

exit(1);

}

while(1)

{

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

if((client\_fd=accept(sockfd,(struct sockaddr\*)&remote\_addr,&sin\_size))

==-1)

{

perror("accept error");

continue;

}

printf("收到一个连接来自: IP:%s",inet\_ntoa(remote\_addr.sin\_addr));

printf("端口号:%d\n",SERVPORT);

/\* 子进程代码段 \*/

if(!fork()){

time\_t ticks;

ticks = time(NULL);

snprintf(buff,sizeof(buff),"%.24s\r%s",ctime(&ticks),"欢迎");

if(send(client\_fd,buff,80,0)==-1)

perror("send出错");

close(client\_fd);

exit(0);

}

close(client\_fd);

}

}

1. 编写一个基于TCP协议的网络通信程序，要求服务器通过socket连接后，并要求输入用户，判断为liu时，才向客户端发送字符串"Hello, you are connected!"。在服务器上显示客户端的IP地址或域名。

**Cu\_client.c**

#include<stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <string.h>

#include<netdb.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#define SERVPORT 3333

#define MAXDATASIZE 100 /\*每次最大数据传输量 \*/

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

{

int sockfd,recvbytes;

char buf[MAXDATASIZE];

struct hostent \*host;

char name[MAXDATASIZE]; /\*定义用户名\*/

struct sockaddr\_in serv\_addr;

if(argc<2)

{

fprintf(stderr,"Please enter the sever's hostname!\n");

exit(1);

}

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

perror("gethostbyname error!");

exit(1);

}

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

perror("socket create error");

exit(1);

}

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

serv\_addr.sin\_port=htons(SERVPORT);

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

bzero(&(serv\_addr.sin\_zero),8);

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

{

perror("connet error!");

exit(1);

}

printf("请输入用户名:");

scanf("%s",name);

name[strlen(name)] = '\0';

fflush(stdout);

send(sockfd, name, strlen(name), 0);

/\*发送用户名到sockfd\*/

if((recvbytes=recv(sockfd,buf,MAXDATASIZE,0))==-1){

perror("connect出错");

exit(1);

}

buf[recvbytes]='\0';

printf("收到:%s",buf);

close(sockfd);

}

**Cu\_server.c**

#include<stdio.h>

#include<stdlib.h>

#include<errno.h>

#include <string.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <sys/socket.h>

#include <sys/wait.h>

#define SERVPORT 3333 /\*服务器监听端口号 \*/

#define BACKLOG 10 /\* 最大同时连接请求数 \*/

int main()

{

int sockfd,client\_fd;

struct sockaddr\_in my\_addr; /\* 本机地址信息 \*/

struct sockaddr\_in remote\_addr; /\* 客户端地址信息 \*/

int sin\_size;

char buff[256];/\*用于输入用户名的缓冲区\*/

int recvbytes;

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

{

perror("socket创建失败！");

exit(1);

}

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

/\* 把16位值从主机字节序转换成网络字节序\*/

my\_addr.sin\_port=htons(SERVPORT);

my\_addr.sin\_addr.s\_addr = INADDR\_ANY;

bzero(&(my\_addr.sin\_zero),8);

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

{

perror("bind出错");

exit(1);

}

if(listen(sockfd,BACKLOG)==-1)

{

perror("listen出错");

exit(1);

}

while(1)

{

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

if((client\_fd=accept(sockfd,(struct sockaddr\*)&remote\_addr,&sin\_size))

==-1)

{

perror("accept error");

continue;

}

if((recvbytes=recv(client\_fd,buff,256,0))==-1){

perror("connect出错");

exit(1);

}

/\*接收从客户端传来的用户名\*/

buff[recvbytes] = '\0';

fflush(stdout);

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

printf("收到一个连接来自: %s\n",inet\_ntoa(remote\_addr.sin\_addr));

/\* 子进程代码段 \*/

if(!fork()){

if(strncmp("chalmers", buff, 4) == 0) {

if(send(client\_fd,"Hello, you are connected!\n",26,0)==-1)

perror("send出错");

exit(0);

}else

send(client\_fd,"please reput username!\n",26,0);

}

close(client\_fd);

}

}

3. 一个以客户机/服务器模式工作，要求在客户端读取系统文件/etc/passwd内容，传送到服务端，服务器端接受字符串，并在显示器显示出来。

客户端：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<netdb.h>

#include<errno.h>

#include<sys/types.h>

int port=8888;

int main()

{

int sockfd;

int i=0;

int z;

char buf[80],str1[80];

struct sockaddr\_in adr\_srvr;

FILE \*fp;

printf("打开文件......\n");

/\*以只读的方式打开liu文件\*/

fp=fopen("/etc/passwd","r");

if(fp==NULL)

{

perror("打开文件失败");

exit(1);

}

printf("连接服务端...\n");

/\* 建立IP地址 \*/

adr\_srvr.sin\_family=AF\_INET;

adr\_srvr.sin\_port=htons(port);

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

bzero(&(adr\_srvr.sin\_zero),8);

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

if(sockfd==-1)

{

perror("socket 出错");

exit(1);

}

printf("发送文件 ....\n");

/\* 读取三行数据，传给udpserver\*/

for(i=0;i<3;i++)

{

fgets(str1,80,fp);

printf("%d:%s",i,str1);

sprintf(buf,"%d:%s",i,str1);

z=sendto(sockfd,buf,sizeof(buf),0,(struct sockaddr \*)&adr\_srvr,

sizeof(adr\_srvr));

if(z<0)

{

perror("recvfrom 出错");

exit(1);

}

}

printf("发送.....\n");

sprintf(buf,"stop\n");

z=sendto(sockfd,buf,sizeof(buf),0,(struct sockaddr \*)&adr\_srvr,

sizeof(adr\_srvr));

if(z<0)

{

perror("sendto 出错");

exit(1);

}

fclose(fp);

close(sockfd);

exit(0);

}

服务器端：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<netdb.h>

#include<errno.h>

#include<sys/types.h>

int port=8888;

int main()

{

int sockfd;

int len;

int z;

char buf[256];

struct sockaddr\_in adr\_inet;

struct sockaddr\_in adr\_clnt;

printf("等待客户端....\n");

/\* 建立IP地址 \*/

adr\_inet.sin\_family=AF\_INET;

adr\_inet.sin\_port=htons(port);

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

bzero(&(adr\_inet.sin\_zero),8);

len=sizeof(adr\_clnt);

/\* 建立socket \*/

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

if(sockfd==-1)

{

perror("socket 出错");

exit(1);

}

/\* 绑定socket \*/

z=bind(sockfd,(struct sockaddr \*)&adr\_inet,sizeof(adr\_inet));

if(z==-1)

{

perror("bind 出错");

exit(1);

}

while(1)

{

/\* 接受传来的信息 \*/

z=recvfrom(sockfd,buf,sizeof(buf),0,(struct sockaddr \*)&adr\_clnt,&len);

if(z<0)

{

perror("recvfrom 出错");

exit(1);

}

buf[z]=0;

printf("接收:%s",buf);

/\* 收到stop字符串，终止连接\*/

if(strncmp(buf,"stop",4)==0)

{

printf("结束....\n");

break;

}

}

close(sockfd);

exit(0);

}

