[socket编程（2）—— 一对多通信](http://blog.csdn.net/robot__man/article/details/52370138)

标签： [socket](http://www.csdn.net/tag/socket)

2016-08-30 15:59 272人阅读 [评论](http://blog.csdn.net/Robot__Man/article/details/52370138#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/Robot__Man/article/details/52370138#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Linux下C应用编程（32） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，遵循GPL协议精神，转载请注明出处。

目录[(?)[+]](http://blog.csdn.net/Robot__Man/article/details/52370138)

1 一对多模型，TCP的编程步骤

服务端：   
1、socket（）获得一个sockfd。注意第二个参数必须SOCK\_STREAM.   
2、准备通信地址（必须服务器的）   
3、bind（）绑定。（开放了端口，允许客户端连接）   
4、监听客户端 listen()函数   
5、等待客户端的连接 accept(),返回用于交互的socket描述符   
6、使用第5步返回sockt描述符，进行读写通信。   
7、关闭sockfd。

客户端：   
客户端的代码与一对一的一样。注意第二个参数必须SOCK\_STREAM.

TCP一对多模型，有两类描述符：   
第一步的描述符不再参与信息交互，只是等待客户端的连接（accept），accept()在客户端连接上来后，会返回一个新的描述符，用于读写通信。

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<unistd.h>

int main(){

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

if (sockfd == -1){

perror("socket"),exit(-1);

}

struct sockaddr\_in addr;

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2222);

addr.sin\_addr.s\_addr = inet\_addr("192.168.66.11");

int res = bind(sockfd,(struct sockaddr\*)&addr,

sizeof(addr));

if (res == -1){

perror("bind"),exit(-1);

}

printf("bind ok\n");

listen(sockfd,100);//监听

struct sockaddr\_in client;

socklen\_t len = sizeof(client);

int fd = accept(sockfd,(struct sockaddr\*)&client,&len);

char \*from = inet\_ntoa(client.sin\_addr);//十六进制转点分十进制

printf("%s连接成功\n",from);

char buf[100] = {};

res = read(fd,buf,sizeof(buf));

printf("接受了%d字节，内容:%s",res,buf);

write(fd,"welcome",7);

close(fd);

close(sockfd);

return 0;

}

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

int main(){

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

if (sockfd == -1){

perror("socket"),exit(-1);

}

struct sockaddr\_in addr;

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2222);//连接端口

addr.sin\_addr.s\_addr = inet\_addr("192.168.66.11");//都是服务器的，改成连接IP

int res = connect(sockfd,(struct sockaddr\*)&addr,

sizeof(addr));

if (res == -1){

perror("connect"),exit(-1);

}

printf("连接成功\n");

write(sockfd,"hello",5);

char buf[100] = {};

res = read(sockfd,buf,sizeof(buf));

printf("读到了%d字节，内容：%s\n",res,buf);

close(sockfd);

return 0;

}

上面的服务端还只是实现了一对一的通信，一对多通信加一个while循环即可。

2 练习

1、在客户端加上输入功能，允许发送不同的信息，并且客户端和服务器端代码要支持多次输入和输出（读写上加循环），输入bye退出。客户端发送的内容改为输入，服务器发回给客户端的内容改成客户端的输入。

2、可以考虑在服务器端启动多进程fork（），支持多个客户端的并行。

server.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<unistd.h>

#include<signal.h>

int sockfd;

void fa(int signo){

printf("服务器正在关闭\n");

sleep(1);

close(sockfd);

exit(0);

}

int main(){

printf("ctrl+c退出服务器\n");

signal(SIGINT,fa);

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

if (sockfd == -1){

perror("socket"),exit(-1);

}

struct sockaddr\_in addr;

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2222);

addr.sin\_addr.s\_addr = inet\_addr("192.168.66.11");

int reuseaddr = 1;//解决地址已被占用问题

setsockopt(sockfd,SOL\_SOCKET,SO\_REUSEADDR,

&reuseaddr,sizeof(reuseaddr));

int res = bind(sockfd,(struct sockaddr\*)&addr,

sizeof(addr));

if (res == -1){

perror("bind"),exit(-1);

}

printf("bind ok\n");

listen(sockfd,100);//监听

while (1){

struct sockaddr\_in client;

socklen\_t len = sizeof(client);

int fd = accept(sockfd,(struct sockaddr\*)&client,

&len);//阻塞函数

char \*from = inet\_ntoa(client.sin\_addr);

printf("%s连接成功\n",from);

pid\_t pid = fork();

if (pid == 0){

char buf[100] = {};

while (1){

res = read(fd,buf,sizeof(buf));

printf("接受了%d字节，内容:%s\n",res,buf);

if (res <= 0){//包括0和-1

break;

}

if (strcmp(buf,"bye") == 0){

break;

}

write(fd,buf,strlen(buf));

memset(buf,0,sizeof(buf));

}

close(fd);

exit(0);

}

close(fd);

}

}

client.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

int main(){

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

if (sockfd == -1){

perror("socket"),exit(-1);

}

struct sockaddr\_in addr;

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2222);//连接端口

addr.sin\_addr.s\_addr = inet\_addr("192.168.66.11");//都是服务器的，改成连接IP

int res = connect(sockfd,(struct sockaddr\*)&addr,sizeof(addr));

if (res == -1){

perror("connect"),exit(-1);

}

printf("连接成功\n");

char buf[100] = {};

while (1){

memset(buf,0,sizeof(buf));//buf清0

printf("请输入要说的话\n");

scanf("%s",buf);

write(sockfd,buf,strlen(buf));

if (strcmp(buf,"bye") == 0){//退出的合适位置

break;

}

memset(buf,0,sizeof(buf));//buf清0

res = read(sockfd,buf,sizeof(buf));

printf("读到了%d字节，内容：%s\n",res,buf);

}

close(sockfd);

return 0;

}