[socket编程（3）—— UDP协议通信](http://blog.csdn.net/robot__man/article/details/52369794)

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

2016-08-30 15:32 23人阅读 [评论](http://blog.csdn.net/robot__man/article/details/52369794#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/robot__man/article/details/52369794#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/52369794)

1 TCP和UDP的区别

　　TCP传输数据时，采用的是**数据流**方式。UDP传输数据时，采用的是**数据报**方式。   
　　TCP是有连接协议（基于连接），在数据传输的整个过程，保持服务器和客户端的连接。会重发一切错误数据，因此可以保证数据的正确性和完整性。   
　　UDP是无连接协议，在数据发送和接收的时候连一下即可，发送/接收完毕就不保持连接。不会重发错误数据，因此UDP有可能丢失数据。

2 UDP编程

**UDP开发没有listen()和accept()，发送和接收函数使用：sendto()和recvfrom()，个别时候可以使用read()，write()基本不能使用。**   
　　sendto()相当于write()+connect()效果，参数是前3个参数和write()一样，第4个参数置0即可，后两个参数和connect()一样。区别：sendto()只是发送时连一下，不会保持连接。而connect()保持连接。   
　　recvfrom()相当于read()+accept()效果，参数前3个和read()一样，第4个参数置0即可，后两个参数和accept()一样。   
　　（注意：accept()和connect()的最后一个参数，前者是取地址长度的地址，是个指针，后者就是通信地址的长度）

下面是一个UDP编程实现的时间服务器：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<time.h>

#include<signal.h>

#include<arpa/inet.h>//加的函数的头文件

int sockfd;

void fa(int signo){

printf("服务器正在关闭!\n");

sleep(1);

close(sockfd);

exit(0);

}

int main(){

printf("按Ctrl+C退出服务器！\n");

signal(SIGINT,fa);

int sockfd = socket(AF\_INET,SOCK\_DGRAM,0);//网络通信AF\_INET

if (sockfd == -1){

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

}

struct sockaddr\_in addr;//网络通信的地址

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2223);//字节顺序本机转网络

addr.sin\_addr.s\_addr = inet\_addr("192.168.66.11");//点分十进制转十六进制

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

sizeof(addr));

if (res == -1){

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

}

printf("绑定成功\n");

char buf[100] = {};

while(1){

struct sockaddr\_in from;

socklen\_t len = sizeof(from);

res = recvfrom(sockfd,buf,100,0,

(struct sockaddr\*)&from,&len);

//res = read(sockfd,buf,100);

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

time\_t cur = time(0);

struct tm \*cu= localtime(&cur);

sprintf(buf,"%4d-%02d-%02d %02d:%02d:%02d",

cu->tm\_year + 1900,cu->tm\_mon + 1,cu->tm\_mday,

cu->tm\_hour,cu->tm\_min,cu->tm\_sec);

printf("客户端%s连接成功\n",inet\_ntoa(from.sin\_addr));

sendto(sockfd,buf,strlen(buf),0,

(struct sockaddr\*)&from,len);

}

//close(sockfd);

}

#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\_DGRAM,0);

if (sockfd == -1){

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

}

struct sockaddr\_in addr;

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(2223);//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 = sendto(sockfd,"hello",5,0,

(struct sockaddr\*)&addr,sizeof(addr));

printf("res = %d\n",res);

char buf[100] = {};

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

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

close(sockfd);

return 0;

}