

kikilizhm的专栏

个人资料



kikilizhm

+ 加关注

发私信

访问：63198次

积分：816

等级：BLOG > 3

排名：千里之外

原创：20篇

转载：4篇

译文：0篇

评论：60条

文章搜索

文章分类

WEB (7)

工具应用 (5)

linux驱动 (7)

uboot移植 (6)

Linux_C (4)

windows编程 (5)

文章存档

2015年03月 (1)

2014年11月 (1)

2013年04月 (1)

2013年03月 (4)

2012年11月 (4)

展开

阅读排行

linux下C语言socket网络编程简例

(28207)

ARM的BIN文件反汇编方

(6516)

【win网络编程】socket编程

(4179)

💡 CSDN Android客户端 下载就送50C币

又见人月神话

最流行的语言想学就学

写博文，传代码，送C币

原 linux 下C语言socket网络编程简例

分类：[linux驱动](#) [Linux_C](#)

2012-08-13 01:34

🔍 28243人阅读

💬 评论(47)

🔖 收藏

🚩 举报

socket

网络

linux

编程

c

语言

原创文章，转载请注明出处字样和出处，谢谢！

这里给出在linux下的简单socket网络编程的实例，使用tcp协议进行通信，服务端进行监听，在收到客户端的连接后，发送数据给客户端；客户端在接受到数据后打印出来，然后关闭。程序里有详细的说明，其中对具体的结构体和函数的实现可以参考其他资料。

程序说明： 这里服务器的端口号和ip地址使用固定的设置，移植时可以根据具体情况更改，可以改写为参数传递更好，这里为了方便，使用固定的。

移植时服务端可以不用更改，编译后可直接运行；客户端将ip改为服务器的地址，然后编译运行。可以使用netstat 进行查看相应的运行状态。

[cpp] view plain copy print ?

```
01.  /*****
02.  文件名： server.c
03.  linux 下socket网络编程简例 - 服务端程序
04.  服务器端口设为 0x8888 （端口和地址可根据实际情况更改，或者使用参数传入）
05.  服务器地址设为 192.168.1.104
06.  作者:kikilizhm#163.com (将#换为@)
07.  */
08.
09.  #include <stdlib.h>
10.  #include <sys/types.h>
11.  #include <stdio.h>
12.  #include <sys/socket.h>
13.  #include <linux/in.h>
14.  #include <string.h>
15.
16.  int main()
17.  {
18.  int sfp,nfp; /* 定义两个描述符 */
19.  struct sockaddr_in s_add,c_add;
20.  int sin_size;
21.  unsigned short portnum=0x8888; /* 服务端使用端口 */
22.
23.  printf("Hello,welcome to my server !\r\n");
24.  sfp = socket(AF_INET, SOCK_STREAM, 0);
25.  if(-1 == sfp)
26.  {
27.      printf("socket fail ! \r\n");
28.      return -1;
29.  }
30.  printf("socket ok !\r\n");
31.
32.  /* 填充服务器端口地址信息，以便下面使用此地址和端口监听 */
33.  bzero(&s_add,sizeof(struct sockaddr_in));
```


基于windows IIS的C语言	(4108)
结构体的自引用（或者说	(2772)
基于VB2008的winsocket	(2247)
windows下sqlite的.dll生	(1448)
【ARM基础知识】ARM	(1309)
虚拟机中linux无法访问外	(1173)
使用 Socket 通信实现 F1	(1066)

评论排行	
linux下C语言socket网络:	(47)
基于windows IIS的C语言	(8)
uboot中lowlevel_init运行	(2)
Windows SOCKET网络	(1)
tiny6410上第一个hello m	(1)
结构体的自引用（或者说	(1)
【win网络编程】socket	(0)
【原EDN博文】USB开发	(0)
Fedora上安装并配置sar	(0)
入门级 网络现场直播 的	(0)

推荐文章	
*Android应用程序UI硬件加速渲染的Display List渲染过程分析	
*外部排序，杀鸡焉用牛刀?	
*算法:比较排序之外学习新的线性时间排序	
*Android自定义控件（状态提示图表）	
*iOS开发之可穿戴设备蓝牙4.0 BLE开发	

最新评论	
linux下C语言socket网络编程简例 qq_24782791: 楼主的例子非常棒啊，适合初学者。我是在一个电脑里面的VMware开两个redhat虚拟机尝试的，两台...	
基于windows IIS的C语言CGI WE色郎中: CGI啊？既然是嵌入式，服务器咋不是部署在嵌入式设备里呢？现在还有在用CGI呀，落伍了，应该用更好的...	
linux下C语言socket网络编程简例 Echo_Wei1991: @kikilizhm:不知道哪里出问题了 没改端口 重新运行一下就好了很奇怪 要不就是我小伙伴给我鼓...	
linux下C语言socket网络编程简例 kikilizhm: @yoany:端口修改为0x400以上试试，应该是端口的问题。	
linux下C语言socket网络编程简例 kikilizhm: @u011310054:客户端的地址有改成你自己的服务器的地址吗、	
linux下C语言socket网络编程简例 kikilizhm: @yoany:bind fail 应该是在服务端吧，是不是端口已经被用了，	
linux下C语言socket网络编程简例 Echo_Wei1991: 总是bind fail! T_Tcli中把 s_add.sin_addr.s_addr=htonl...	
基于windows IIS的C语言CGI WE blinkNL: 跟lz握个手我也是想在IIS上搭建一下CGI的环境，网上这方面资料确实少啊，lz这篇文章对我很有帮助...	
tiny6410上第一个hello module驱 wangyao199252: 无法卸载	

34.	s_add.sin_family=AF_INET;
35.	s_add.sin_addr.s_addr=htonl(INADDR_ANY); /* 这里地址使用全0，即所有 */
36.	s_add.sin_port=htons(portnum);
37.	/* 使用bind进行绑定端口 */
38.	if(-1 == bind(sfp,(struct sockaddr *)&s_add), sizeof(struct sockaddr)))
39.	{
40.	printf("bind fail !\r\n");
41.	return -1;
42.	}
43.	printf("bind ok !\r\n");
44.	/* 开始监听相应的端口 */
45.	if(-1 == listen(sfp,5))
46.	{
47.	printf("listen fail !\r\n");
48.	return -1;
49.	}
50.	printf("listen ok\r\n");
51.	
52.	while(1)
53.	{
54.	sin_size = sizeof(struct sockaddr_in);
55.	/* accept服务端使用函数，调用时即进入阻塞状态，等待用户进行连接，在没有客户端进行连接时，程序停止在此处，
56.	不会看到后面的打印，当有客户端进行连接时，程序马上执行一次，然后再次循环到此处继续等待。
57.	此处accept的第二个参数用于获取客户端的端口和地址信息。
58.	*/
59.	nfp = accept(sfp, (struct sockaddr *)&c_add), &sin_size);
60.	if(-1 == nfp)
61.	{
62.	printf("accept fail !\r\n");
63.	return -1;
64.	}
65.	printf("accept ok!\r\nServer start get connect from %x : %x\r\n",ntohl(c_add.sin_addr.s_addr,
66.	
67.	/* 这里使用write向客户端发送信息，也可以尝试使用其他函数实现 */
68.	if(-1 == write(nfp,"hello,welcome to my server \r\n",32))
69.	{
70.	printf("write fail!\r\n");
71.	return -1;
72.	}
73.	printf("write ok!\r\n");
74.	close(nfp);
75.	
76.	}
77.	close(sfp);
78.	return 0;
79.	}

[cpp] view plain copy print ?	
01.	/******
02.	文件名: server.c
03.	linux 下socket网络编程简例 - 服务端程序
04.	服务器端口设为 0x8888 （端口和地址可根据实际情况更改，或者使用参数传入）
05.	服务器地址设为 192.168.1.104
06.	作者:kikilizhm#163.com（将#换为@）
07.	*/
08.	
09.	#include <stdlib.h>
10.	#include <sys/types.h>
11.	#include <stdio.h>
12.	#include <sys/socket.h>
13.	#include <linux/in.h>
14.	#include <string.h>
15.	
16.	int main()
17.	{
18.	int sfp,nfp; /* 定义两个描述符 */
19.	struct sockaddr_in s_add,c_add;
20.	int sin_size;
21.	unsigned short portnum=0x8888; /* 服务端使用端口 */
22.	
23.	printf("Hello,welcome to my server !\r\n");
24.	sfp = socket(AF_INET, SOCK_STREAM, 0);
25.	if(-1 == sfp)
26.	{

hello.ko

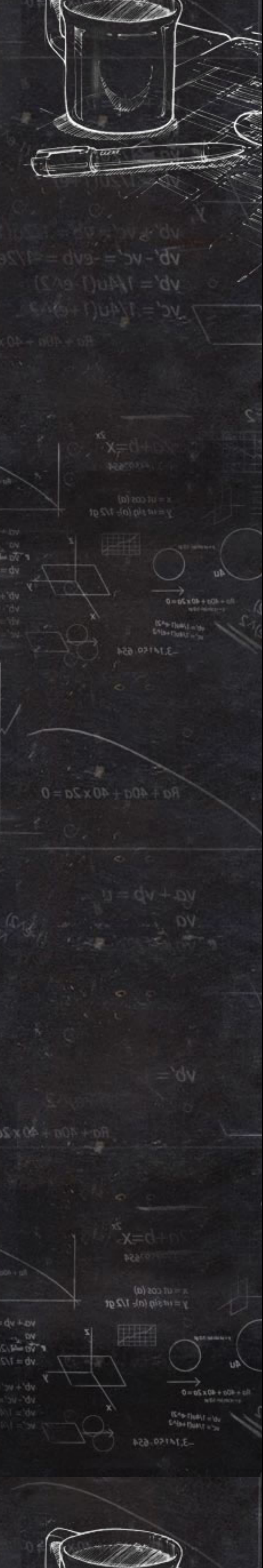
uboot中lowlevel_init运行不了的理

kikilizhm: @liwendovo:那目标函数的最后有 mov pc, lr 吗 ,

```
27.     printf("socket fail ! \r\n");
28.     return -1;
29. }
30. printf("socket ok !\r\n");
31.
32. /* 填充服务器端口地址信息，以便下面使用此地址和端口监听 */
33. bzero(&s_add,sizeof(struct sockaddr_in));
34. s_add.sin_family=AF_INET;
35. s_add.sin_addr.s_addr=htonl(INADDR_ANY); /* 这里地址使用全0，即所有 */
36. s_add.sin_port=htons(portnum);
37. /* 使用bind进行绑定端口 */
38. if(-1 == bind(sfp,(struct sockaddr *)&s_add, sizeof(struct sockaddr)))
39. {
40.     printf("bind fail !\r\n");
41.     return -1;
42. }
43. printf("bind ok !\r\n");
44. /* 开始监听相应的端口 */
45. if(-1 == listen(sfp,5))
46. {
47.     printf("listen fail !\r\n");
48.     return -1;
49. }
50. printf("listen ok\r\n");
51.
52. while(1)
53. {
54.     sin_size = sizeof(struct sockaddr_in);
55.     /* accept服务端使用函数，调用时即进入阻塞状态，等待用户进行连接，在没有客户端进行连接时，程序停止在此处，
56.        不会看到后面的打印，当有客户端进行连接时，程序马上执行一次，然后再次循环到此处继续等待。
57.        此处accept的第二个参数用于获取客户端的端口和地址信息。
58.        */
59.     nfp = accept(sfp, (struct sockaddr *)&c_add, &sin_size);
60.     if(-1 == nfp)
61.     {
62.         printf("accept fail !\r\n");
63.         return -1;
64.     }
65.     printf("accept ok!\r\nServer start get connect from %x : %x\r\n",ntohl(c_add.sin_addr.s_addr),ntohl(sin_addr.sin_port));
66.
67.     /* 这里使用write向客户端发送信息，也可以尝试使用其他函数实现 */
68.     if(-1 == write(nfp,"hello,welcome to my server \r\n",32))
69.     {
70.         printf("write fail!\r\n");
71.         return -1;
72.     }
73.     printf("write ok!\r\n");
74.     close(nfp);
75.
76. }
77. close(sfp);
78. return 0;
79. }
```

[cpp] view plain copy print ?

```
01. /******
02. 文件名: client.c
03. linux 下socket网络编程简例 - 客户端程序
04. 服务器端口设为 0x8888 （端口和地址可根据实际情况更改，或者使用参数传入）
05. 服务器地址设为 192.168.1.104
06. 作者:kikilizhm#163.com (将#换为@)
07. */
08.
09. #include <stdlib.h>
10. #include <sys/types.h>
11. #include <stdio.h>
12. #include <sys/socket.h>
13. #include <linux/in.h>
14. #include <string.h>
15.
```

```
16. int main()  
17. {  
18.     int cfd; /* 文件描述符 */  
19.     int recbytes;  
20.     int sin_size;  
21.     char buffer[1024]={0};    /* 接受缓冲区 */  
22.     struct sockaddr_in s_add,c_add; /* 存储服务端和本端的ip、端口等信息结构体 */  
23.     unsigned short portnum=0x8888; /* 服务端使用的通信端口，可以更改，需和服务端相同 */  
24.  
25.     printf("Hello,welcome to client !\r\n");  
26.     /* 建立socket 使用因特网，TCP流传输 */  
27.     cfd = socket(AF_INET, SOCK_STREAM, 0);  
28.     if(-1 == cfd)  
29.     {  
30.         printf("socket fail ! \r\n");  
31.         return -1;  
32.     }  
33.     printf("socket ok !\r\n");  
34.     /* 构造服务器端的ip和端口信息，具体结构体可以查资料 */  
35.     bzero(&s_add,sizeof(struct sockaddr_in));  
36.     s_add.sin_family=AF_INET;  
37.     s_add.sin_addr.s_addr= inet_addr("192.168.1.104"); /* ip转换为4字节整形，使用时需要根据服务端ip进行更改 */  
38.     s_add.sin_port=htons(portnum); /* 这里htons是将short型数据字节序由主机型转换为网络型，其实就是  
39.         将2字节数据的前后两个字节倒换，和对应的ntohs效果、实质相同，只不过名字不同。htonl和ntohl是  
40.         操作的4字节整形。将0x12345678变为0x78563412，名字不同，内容两两相同，一般情况下网络为大端，  
41.         PPC的cpu为大端，x86的cpu为小端，arm的可以配置大小端，需要保证接收时字节序正确。  
42.     */  
43.  
44.     printf("s_addr = %x ,port : %x\r\n",s_add.sin_addr.s_addr,s_add.sin_port); /* 这里打印出的是  
45.         小端  
46.         和我们平时看到的是相反的。 */  
47.  
48.     /* 客户端连接服务器，参数依次为socket文件描述符，地址信息，地址结构大小 */  
49.     if(-1 == connect(cfd,(struct sockaddr *)&s_add, sizeof(struct sockaddr)))  
50.     {  
51.         printf("connect fail !\r\n");  
52.         return -1;  
53.     }  
54.     printf("connect ok !\r\n");  
55.     /*连接成功,从服务端接收字符*/  
56.     if(-1 == (recbytes = read(cfd,buffer,1024)))  
57.     {  
58.         printf("read data fail !\r\n");  
59.         return -1;  
60.     }  
61.     printf("read ok\r\nREC:\r\n");  
62.  
63.     buffer[recbytes]='\0';  
64.     printf("%s\r\n",buffer);  
65.  
66.     getchar(); /* 此句为使程序暂停在此处，可以使用netstat查看当前的连接 */  
67.     close(cfd); /* 关闭连接，本次通信完成 */  
68.     return 0;  
69.  
70.  
71. }
```

```
[cpp] view plain copy print ?  
  
01. /*****  
02. 文件名: client.c  
03. linux 下socket网络编程简例 - 客户端程序  
04. 服务器端口设为 0x8888 （端口和地址可根据实际情况更改，或者使用参数传入）  
05. 服务器地址设为 192.168.1.104  
06. 作者:kikilizhm#163.com (将#换为@)  
07. */  
08.  
09. #include <stdlib.h>  
10. #include <sys/types.h>  
11. #include <stdio.h>  
12. #include <sys/socket.h>  
13. #include <linux/in.h>  
14. #include <string.h>  
15.
```



```

16. int main()
17. {
18.     int cfd; /* 文件描述符 */
19.     int recbytes;
20.     int sin_size;
21.     char buffer[1024]={0};    /* 接受缓冲区 */
22.     struct sockaddr_in s_add,c_add; /* 存储服务端和本端的ip、端口等信息结构体 */
23.     unsigned short portnum=0x8888; /* 服务端使用的通信端口，可以更改，需和服务端相同 */
24.
25.     printf("Hello,welcome to client !\r\n");
26.     /* 建立socket 使用因特网，TCP流传输 */
27.     cfd = socket(AF_INET, SOCK_STREAM, 0);
28.     if(-1 == cfd)
29.     {
30.         printf("socket fail ! \r\n");
31.         return -1;
32.     }
33.     printf("socket ok !\r\n");
34.     /* 构造服务器端的ip和端口信息，具体结构体可以查资料 */
35.     bzero(&s_add,sizeof(struct sockaddr_in));
36.     s_add.sin_family=AF_INET;
37.     s_add.sin_addr.s_addr= inet_addr("192.168.1.104"); /* ip转换为4字节整形，使用时需要根据服务端ip进行更改 */
38.     s_add.sin_port=htons(portnum); /* 这里htons是将short型数据字节序由主机型转换为网络型，其实就是
39.         将2字节数据的前后两个字节倒换，和对应的ntohs效果、实质相同，只不过名字不同。htonl和ntohl是
40.         操作的4字节整形。将0x12345678变为0x78563412，名字不同，内容两两相同，一般情况下网络为大端，
41.         PPC的cpu为大端，x86的cpu为小端，arm的可以配置大小端，需要保证接收时字节序正确。
42.     */
43.
44.     printf("s_addr = %x ,port : %x\r\n",s_add.sin_addr.s_addr,s_add.sin_port); /* 这里打印出的是
45.         小端
46.         和我们平时看到的是相反的。 */
47.
48.     /* 客户端连接服务器，参数依次为socket文件描述符，地址信息，地址结构大小 */
49.     if(-1 == connect(cfd,(struct sockaddr *)&s_add, sizeof(struct sockaddr)))
50.     {
51.         printf("connect fail !\r\n");
52.         return -1;
53.     }
54.     printf("connect ok !\r\n");
55.     /*连接成功,从服务端接收字符*/
56.     if(-1 == (recbytes = read(cfd,buffer,1024)))
57.     {
58.         printf("read data fail !\r\n");
59.         return -1;
60.     }
61.     printf("read ok\r\nREC:\r\n");
62.
63.     buffer[recbytes]='\0';
64.     printf("%s\r\n",buffer);
65.
66.     getchar(); /* 此句为使程序暂停在此处，可以使用netstat查看当前的连接 */
67.     close(cfd); /* 关闭连接，本次通信完成 */
68.     return 0;
69.
70.
71. }

```

运行截图：

```

[root@localhost test]# ls
cli cli.c ser ser.c
[root@localhost test]# ./cli
hello,welcome to client !
socket ok !
s_addr = 0x6801a8c0 ,port : 0x8888
connect ok !
read ok
REC:
hello,welcome to my server

```

```

[root@localhost test]#
[root@localhost test]#
[root@localhost test]#
[root@localhost test]# ls
cli cli.c ser ser.c
[root@localhost test]# ./ser
hello,welcome to my server !
socket ok !
bind ok !
listen ok
accept ok!
Server start get connect from 0x6801a8c0 : 0x409b
write ok!

```


⬆️ 上一篇

基于windows IIS的C语言CGI WEB服务器环境搭建

⬇️ 下一篇

基于VB2008的winsocket控件网络编程

顶

28

踩

0

主题推荐

- 网络编程
- c语言
- socket
- linux
- 结构

猜你在找

- 使用socket的Linux上的C语言文件传输顺序服务器
 - libpcap使用
 - 一个华为人辞职创业后的几个反思
 - c++ using namespace详解
 - linux驱动面试题目汇总
- Linux企业常用文件管理命令详解
 - C语言基础视频教程
 - Linux核心必备技能-用户及权限详解
 - 疯狂IOS讲义之Objective-C面向对象设计
 - C语言及程序设计提高
 - Linux企业常用文件管理命令详解
 - C语言基础视频教程
 - Linux核心必备技能-用户及权限详解
 - 疯狂IOS讲义之Objective-C面向对象设计
 - PHP网站搭建入门

准备好了么？跳 吧！

更多职位尽在 CSDN JOB

C语言开发工程师	我要跳槽	C语言软件开发助理工程师	我要跳槽
上海新网程信息技术股份有限公司	10-15K/月	北京南天软件有限公司	4-5K/月
C语言开发工程师	我要跳槽	C语言开发工程师	我要跳槽
上海明运信息科技有限公司	6-8K/月	北京南天软件有限公司	5-10K/月



3 Steps to Watch Free Online TV

1. Click "Start Download"
2. Get Free Access – No Sign up!
3. Watch Full Episodes & Sports Online!

查看评论

- 24楼 qq_24782791 2015-05-28 23:41发表

C

楼主的例子非常棒啊，适合初学者。

我是在一个电脑里面的VMware开两个redhat虚拟机尝试的，两台虚拟机之间能够相互ping通，但就是出现connct fail。IP地址也已经修改为相对于的IP。

楼主有什么意见没，谢谢。
- 23楼 Echo_Wei1991 2014-11-27 21:07发表



总是bind fail! T_T

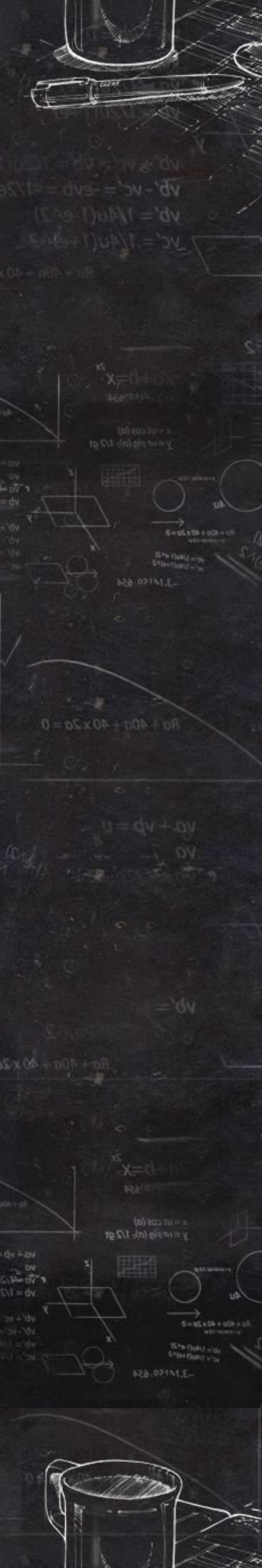
cli中把

s_addr.sin_addr.s_addr=htonl(inet_addr("192.168.1.104"));

改成了自己的ip 也不行
- Re: kikirizhm 2014-12-01 13:23发表



回复Echo_Wei1991： 端口修改为0x400以上试试，应该是端口的问题。
- Re: Echo_Wei1991 2014-12-19 17:37发表



回复kikirizhm：不知道哪里出问题了 没改端口 重新运行一下就好了
很奇怪 要不就是我小伙伴给我鼓捣好了我不知道
谢谢博主啦

Re: kikirizhm 2014-11-29 22:47发表



回复Echo_Wei1991：bind fail 应该是在服务端吧，是不是端口已经被用了，

22楼 汐月淋枫 2013-07-05 08:53发表



在不？我在linux下运行后怎么会出现怎么着问题呢 accept fail
和connect fail 是怎么回事啊，帮忙讲讲，可以的加我QQ406148020

Re: kikirizhm 2014-11-29 22:50发表



回复汐月淋枫：客户端的地址有改成你自己的服务器的地址吗、

21楼 liuyang19890710 2012-08-26 19:04发表



写得不错，通俗易懂。希望循序渐进，。。

20楼 zhlemond 2012-08-22 13:29发表



还没搞明白怎样设置服务器的端口号

Re: kikirizhm 2012-08-22 23:49发表



回复zhlemond：服务器端是通过bind函数绑定此次通信的端口的，可以找资料详细看看那个结构体，客户端连接时只需要设置服务器的端口，客户端本身使用的端口是随机的，服务端可以得到连接的客户端所使用的具体端口号。就比如AB两人打电话，A要打给B，B的手机号必须是确定的，A必须知道B的手机号，至于A的号码是多少都无所谓，都可以通，在B的手机上可以通过来电显示看到A的号码。

19楼 Human陈思宇 2012-08-18 20:33发表



说的不错

18楼 fdl19881 2012-08-17 21:52发表



如果真想学socket编程的话，就看《unix网络编程卷一》进行系统的学习。上面有大量的例子，讲解也清楚。
只是在网上随便看看这样的例子的话，是学不好网络编程的。

Re: kikirizhm 2012-08-17 23:08发表



回复fdl19881：嗯 好的 谢谢了~

17楼 wzqqz 2012-08-17 17:04发表



你能不能把你的头文件给我的邮箱发一份呢，非常感谢

16楼 luomianhe 2012-08-16 16:58发表



学习学习

15楼 test_lockxxx 2012-08-16 16:42发表



对初学者来讲，非常好的一篇文章。

14楼 qiuchen1512 2012-08-16 10:28发表



网路变成讲究的流程，其实怎么写实个人的习惯罢了。

13楼 quotee 2012-08-15 16:36发表



学习学习

12楼 heymye 2012-08-15 08:54发表



这个例子用于理解socket的原理可以。服务端的代码建议看下epoll或者select。

11楼 ruleless 2012-08-15 00:01发表



CSDN还是很水地！！

10楼 w835369950 2012-08-14 20:16发表



请问，我用VC6.0能跑这程序吗？不行的话，要怎么修改才可以

Re: [kikilizhm](#) 2012-08-14 21:39发表



回复w835369950： 不行的~ 你可以使用winsock控件，或者调用win的api函数，我没有看过，不过使用winsock我在后一篇里用vb做了个例子，也是和这里的主机通信的，你可以借鉴一下在vc里是一样的~

9楼 [binarie](#) 2012-08-14 18:29发表



感觉还是C写socket比较舒服 可能是个人喜好问题用C++实在是有点不伦不类的感觉

8楼 [taozi343805436](#) 2012-08-14 17:48发表



写的不错，有时间仔细看看

7楼 [liulzju](#) 2012-08-14 11:35发表



学习了

6楼 [allen9172](#) 2012-08-14 11:02发表



linux下的，看看听好的

5楼 [Temo](#) 2012-08-14 09:56发表



这种文章也能上CSDN首页，CSDN真是要完蛋了！！！

Re: [望舒](#) 2012-08-18 13:25发表



回复Temo： 我觉得CSDN就是个给初学者,入门的平台,你可以看看除了那几个,像版主的老家伙,有哪个人会写博客啊,写的也是东抄抄西抄抄,或者就是抄英文demo文档里的例子什么的.基本上学到一定程度都忙着赶工去了,有人写点东西应该高兴和满足了.

Re: [kikilizhm](#) 2012-08-14 12:43发表



回复Temo： 写着文章的目的一是自己学习，一是和初学者交流。也许我等菜鸟确实很多~

Re: [whitepop](#) 2012-08-14 13:45发表



他的意思是说，文章水平越来越低层次了。建议楼主看一下理查德史蒂芬的书，哈哈。你会觉得写几个例子多么无聊，当初我和你一样想写个例子玩玩，看完他的书，我对网上一切例子都嗤之以鼻了，因为都是小儿科。UNP卷一卷二就是权威。其他例子不需要了

回复kikilizhm：

Re: [kikilizhm](#) 2012-08-14 18:31发表



回复whitepop：好的，初学，所以也写不出太深入的，只是写来练练手，所以特地用了“简例”~ 就怕别人笑话~ 同时我觉得发出来对跟我一样的初学者还是有用的，往往很多简单问题牛人懒得理。但是自己明白过来之后觉得确实很简单的~ 我觉得技术博客对于很多人来说可能只是一个记录自己的过程，其次才是对外的一个分享~

Re: [whitepop](#) 2012-08-17 13:21发表



嗯，很好的学习习惯~ 向你学习，我现在都没动力写博客了。

回复kikilizhm：

Re: [jiangkaiyao001](#) 2012-08-14 17:21发表



回复whitepop：帅哥，给一下你说的理查德史蒂芬的书名呗，全一点的，比如说第几版，出版社，多谢

Re: [kikilizhm](#) 2012-08-14 23:09发表



回复jiangkaiyao001：TCP/IP详解 verycd上有~ 总共3卷，值得慢慢看~

Re: [jiangkaiyao001](#) 2012-08-15 13:39发表



回复kikilizhm：多谢

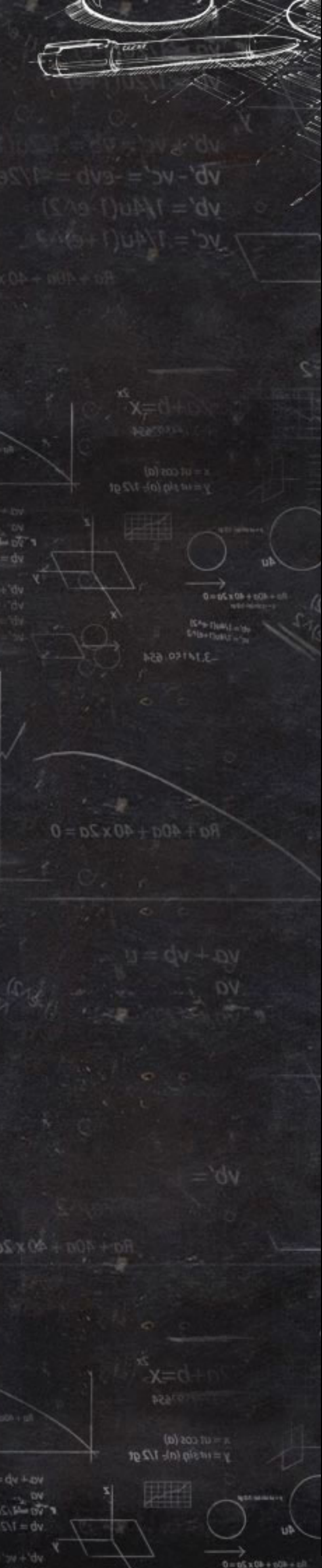
4楼 [cheniwantyou](#) 2012-08-13 20:59发表



C写SOCKET真的是有够蛋疼的。。。

3楼 [jiangkaiyao001](#) 2012-08-13 16:56发表

hi，楼主你好



有个问题问一下，使用socket编程是不是需要新增加什么socket包啊，我运行你这个程序的时候有报错“d:\jiangkaiyao\myself file\vc6.0\vc++\socket_server.c(13) : fatal error C1083: Cannot open include file: 'sys/socket.h': No such file or directory”
看起来是我没有这个socket的头文件，请问你在编译这个之前有添加过socket包吗？能否将你的头文件给我发一个呢？非常感谢

Re: kikirizhm 2012-08-13 18:17发表



回复jiangkaiyao001： 我的这个例子是linux下的，需要在linux环境下编译~ 看你的目录貌似是在win下吧~

Re: jiangkaiyao001 2012-08-14 17:19发表



回复kikirizhm： 那不是重点啊，呵呵，重点是我在所有的头文件中都找了没有的，你能不能把你的头文件给我的邮箱发一份呢，非常感谢

Re: kikirizhm 2012-08-14 18:19发表



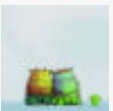
回复jiangkaiyao001： win下和linux下所带的文件应该都不一样的，肯定有关系吧~ 你要是在linux下肯定能编过~ win下vc的话你可以用winsock的控件，或者win下的api写~ 但是跟这个例子肯定没关系了~

Re: jiangkaiyao001 2012-08-15 13:40发表



回复kikirizhm： 哦，这样子，多谢

Re: ValentinsLee 2012-08-15 16:18发表



在WIN下SOCKET编程1: D 话要下载SDK，要么就用VS2008以上的版本。回复jiangkaiyao001：

2楼 dxxang 2012-08-13 09:24发表



学习了

1楼 honghongsummer 2012-08-13 09:19发表



最近在研究socket，借鉴一下~~~

Re: kikirizhm 2012-08-13 09:45发表



回复honghongsummer： 我也是刚学习，多多交流~

Re: honghongsummer 2012-08-30 09:21发表



回复kikirizhm： 嗯嗯。。。



您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场


核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack		
VPN	Spark	ERP	IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	
BI	HTML5	Spring	Apache	.NET	API	HTML	SDK	IIS	Fedora	XML	LBS	Unity
Splashtop	UML	components	Windows Mobile	Rails	QEMU	KDE	Cassandra	CloudStack				
FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	SpringSide	Maemo			
Compuware	大数据	apttech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr		
Angular	Cloud Foundry	Redis	Scala	Django	Bootstrap							

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

 网站客服  杂志客服  微博客服  webmaster@csdn.net  400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

 网站客服  杂志客服  微博客服  webmaster@csdn.net  400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 

