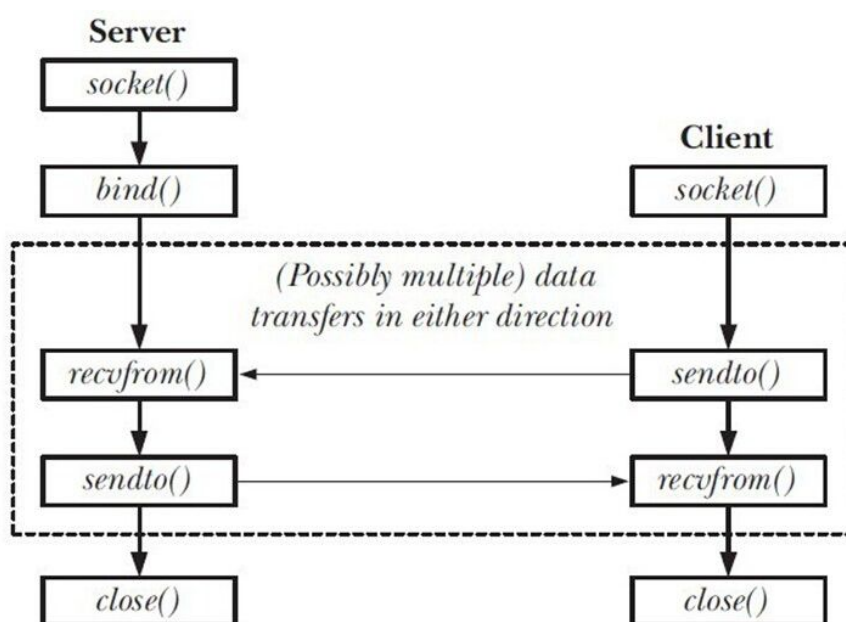


iTOP-4412 实现基于 UDP 的 socket 编程

大家好，上一章我们讲了基于 TCP 的 socket 编程，这一章我们讲解一下基于 UDP 的 socket 编程，UDP 是用户数据报协议，它是一种无连接的传输层协议，提供面向事物的简单不可靠信息传送服务，所以在一些网络质量不满意的坏境下，UDP 协议数据包丢失会比较严重，会造成数据的丢失。UDP 的特点是他不属于连接型协议，所以资源消耗小。处理速度快的优点，所以通常音频，视频和普通数据在传送时使用 UDP 较多，因为它们即使偶尔丢失一两个数据包，也不会对接收结果产生太大影响。

UDP 的通信框图如下图所示：



从上图可以看到基于 UDP 通信的程序也是由服务器和客户端组成。与 TCP 的服务器不同，UDP 的服务器不需要 `listen` 和 `accept` 函数，客户端不需要 `connect` 函数。

下面我们通过一个例子来了解一下 UDP 的应用程序，首先我们来看下服务器(server)端的程序 `server.c`：

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <sys/socket.h>

#include <netinet/in.h>


int main(int argc, char **argv)
{

    int n;

    char recvline[1024] = {0};


    int sockfd;

    struct sockaddr_in servaddr;


    /* 创建一个 UDP 连接的 socket */

    sockfd = socket(PF_INET, SOCK_DGRAM, 0);


    /* 变量 servaddr 清零 */

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin_family = AF_INET;

    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);

    servaddr.sin_port = htons(50001);


    /* 绑定 servaddr 到创建的 socket 上 */
```

```
bind(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr));
```

```
/* 接收客户端发送的数据 */
```

```
recvfrom(sockfd, recvline, 1024, 0, NULL, NULL);
```

```
printf("%s\n", recvline);
```

```
/* 关闭 socket 连接 */
```

```
close(sockfd);
```

```
}
```

上面的代码和我们前面一章的 TCP 的服务器程序差不多，首先也是定义一些变量，然后调用 socket 函数创建一个套接字，这里 socket 函数的第二个参数是 SOCK_DGRAM，即用户数据报，也就是创建的 UDP 连接，然后调用 bzero 函数清空变量 servaddr 的值，然后设置 servaddr 结构的值，接着调用 bind 绑定函数，然后调用 recvfrom 函数直接接收数据，而没有调用 listen 和 accept 函数。

下面我们来看看客户端（client）的程序 client.c:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

```
int main(int argc, char **argv)
```

```
{  
  
    int sockfd;  
  
    struct sockaddr_in servaddr;  
  
  
    if(argc != 2)  
    {  
  
        printf("usage: ./client [ip]\n");  
  
  
        return -1;  
    }  
  
    /* 创建一个 UDP 的 socket 连接 */  
  
    sockfd = socket(PF_INET, SOCK_DGRAM, 0);  
  
  
    /* 变量 servaddr 清零 */  
  
    bzero(&servaddr, sizeof(servaddr));  
  
    servaddr.sin_family = AF_INET;  
  
    servaddr.sin_port = htons(50001);  
  
    servaddr.sin_addr.s_addr = inet_addr(argv[1]);  
  
  
    char sendline[100];
```

```
sprintf(sendline, "Hello, world!");
```

```
/* 发送数据 */
```

```
sendto(sockfd, sendline, strlen(sendline), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
```

```
/* 关闭 socket 连接 */
```

```
close(sockfd);
```

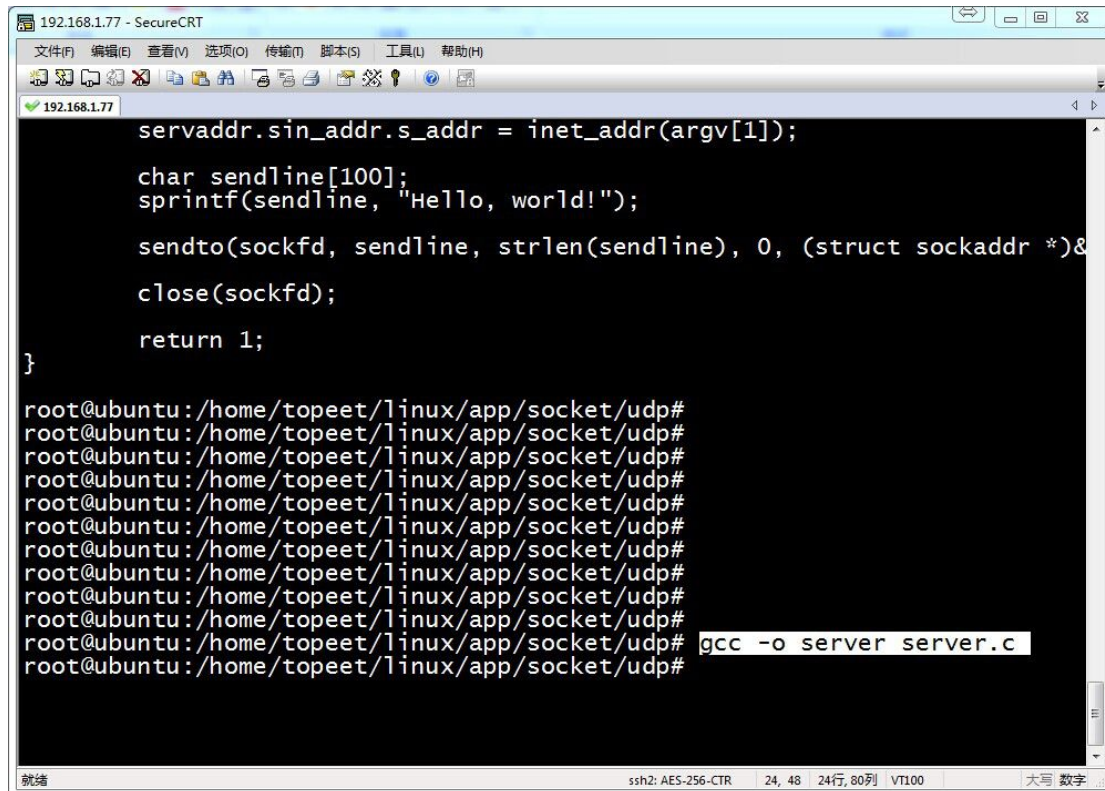
```
return 1;
```

```
}
```

客户端 (client) 的程序首先是定义一些使用的变量 , 然后也是调用 socket 函数创建一个套接字 , socket 函数的第二个参数是 SOCK_DGRAM , 即用户数据报 , 也就是 UDP 连接 , 然后调用 bzero 函数把 servaddr 结构的变量清零 , 然后设置 servaddr 结构变量的值 , 然后调用 sendto 发送函数 , 这里客户端程序没有调用 bind 函数和 connect 函数。现在我们使用命令

```
gcc -o server server.c
```

执行结果如下图 :



```
192.168.1.77 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(I) 帮助(H)
192.168.1.77
servaddr.sin_addr.s_addr = inet_addr(argv[1]);

char sendline[100];
sprintf(sendline, "Hello, world!");

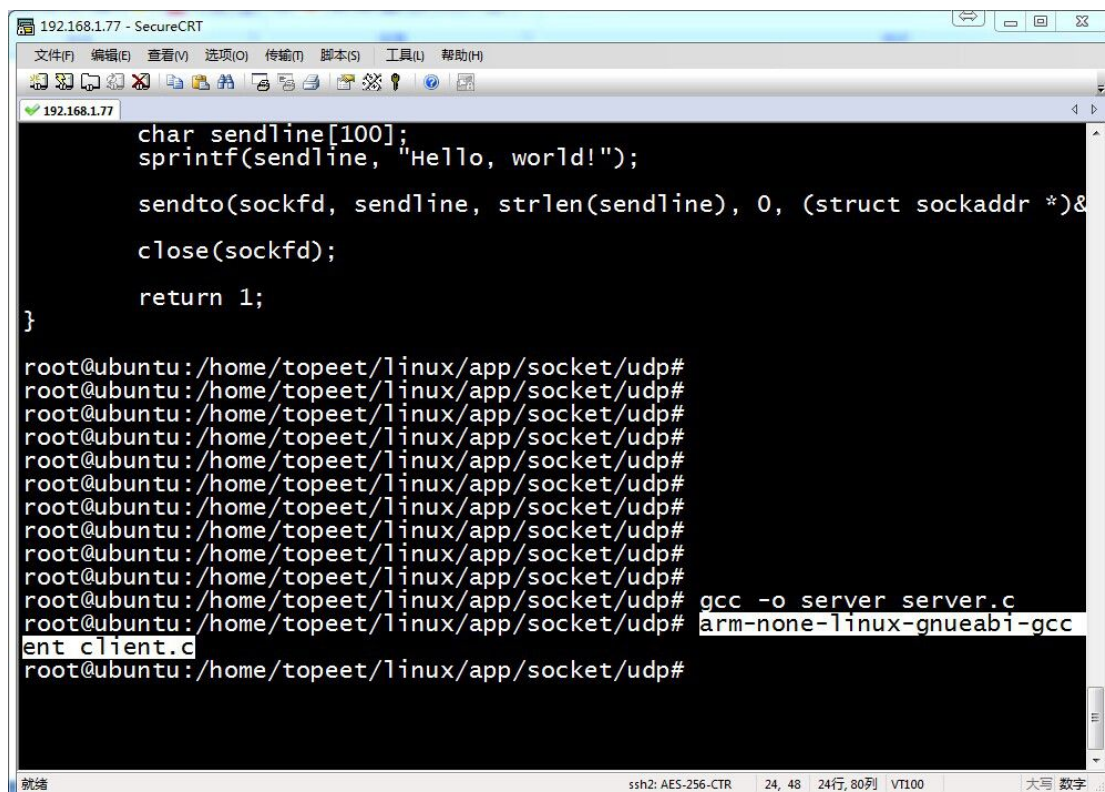
sendto(sockfd, sendline, strlen(sendline), 0, (struct sockaddr *)&

close(sockfd);

return 1;
}

root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# gcc -o server server.c
root@ubuntu:/home/topeet/linux/app/socket/udp#
```

使用命令 `arm-none-linux-gnueabi-gcc -o client client.c` 编译 `client.c` , 执行结果如下图：



```
192.168.1.77 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(I) 帮助(H)
192.168.1.77
char sendline[100];
sprintf(sendline, "Hello, world!");

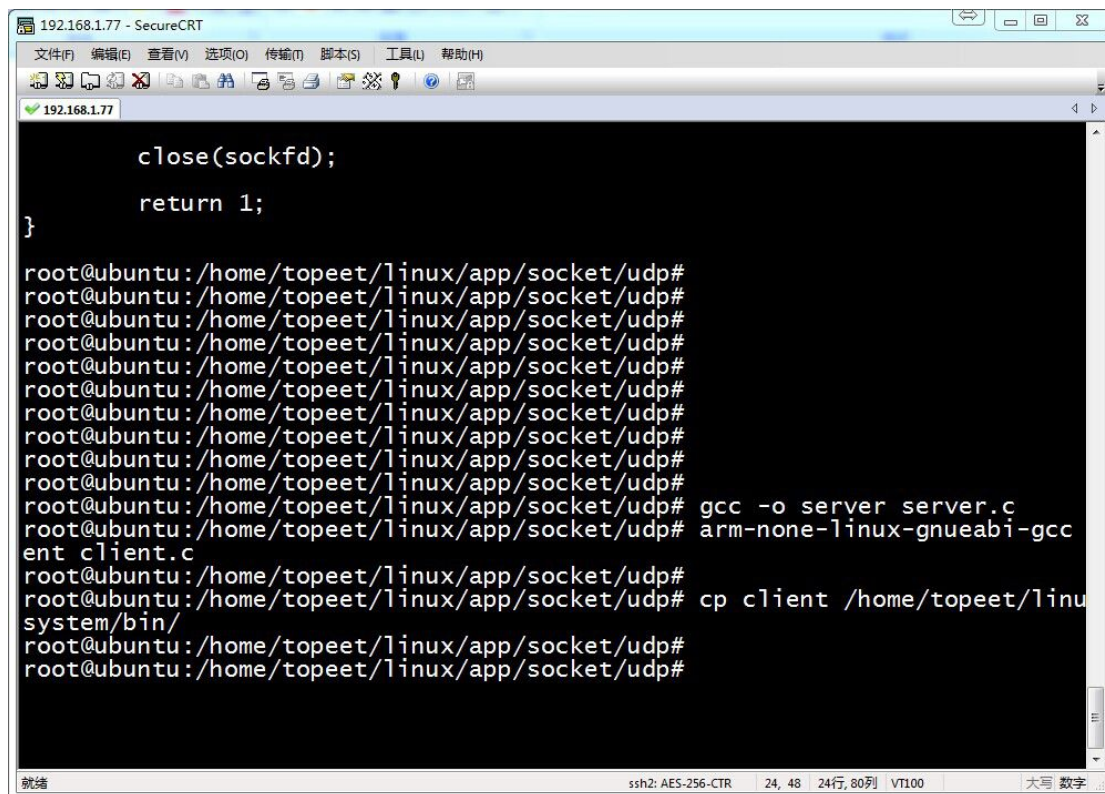
sendto(sockfd, sendline, strlen(sendline), 0, (struct sockaddr *)&

close(sockfd);

return 1;
}

root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# gcc -o server server.c
root@ubuntu:/home/topeet/linux/app/socket/udp# arm-none-linux-gnueabi-gcc
ent client.c
root@ubuntu:/home/topeet/linux/app/socket/udp#
```

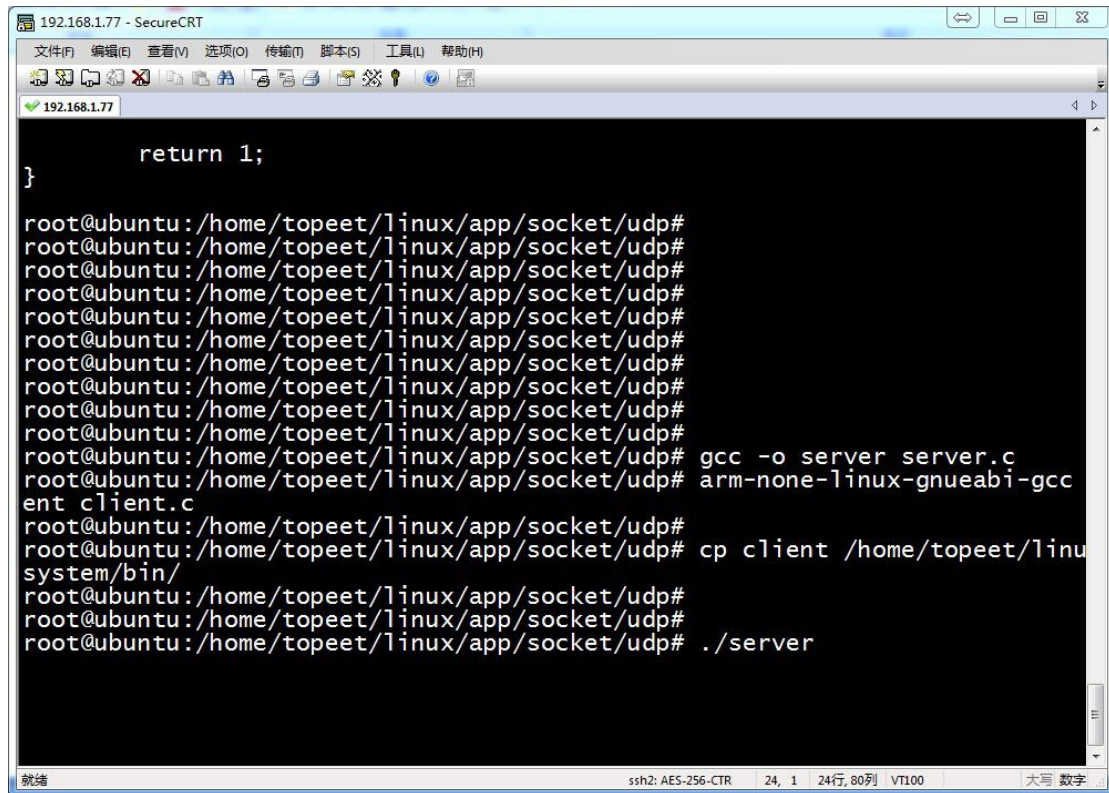
我们把生成的 client 拷贝到我们的 NFS 文件系统的 bin 目录下，如下图：



```
close(sockfd);
return 1;
}

root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# gcc -o server server.c
root@ubuntu:/home/topeet/linux/app/socket/udp# arm-none-linux-gnueabi-gcc
ent client.c
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# cp client /home/topeet/linux
system/bin/
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
```

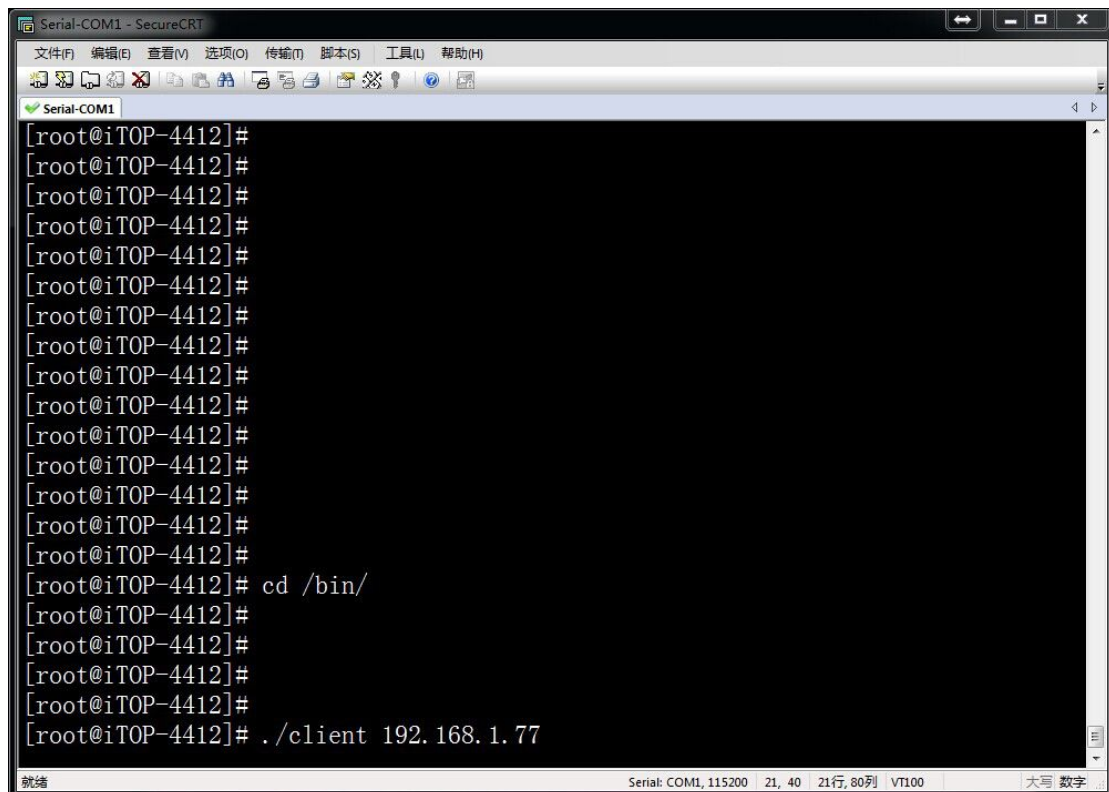
然后我们在虚拟机 Ubuntu 上运行 server 程序，如下图：



```
return 1;
}

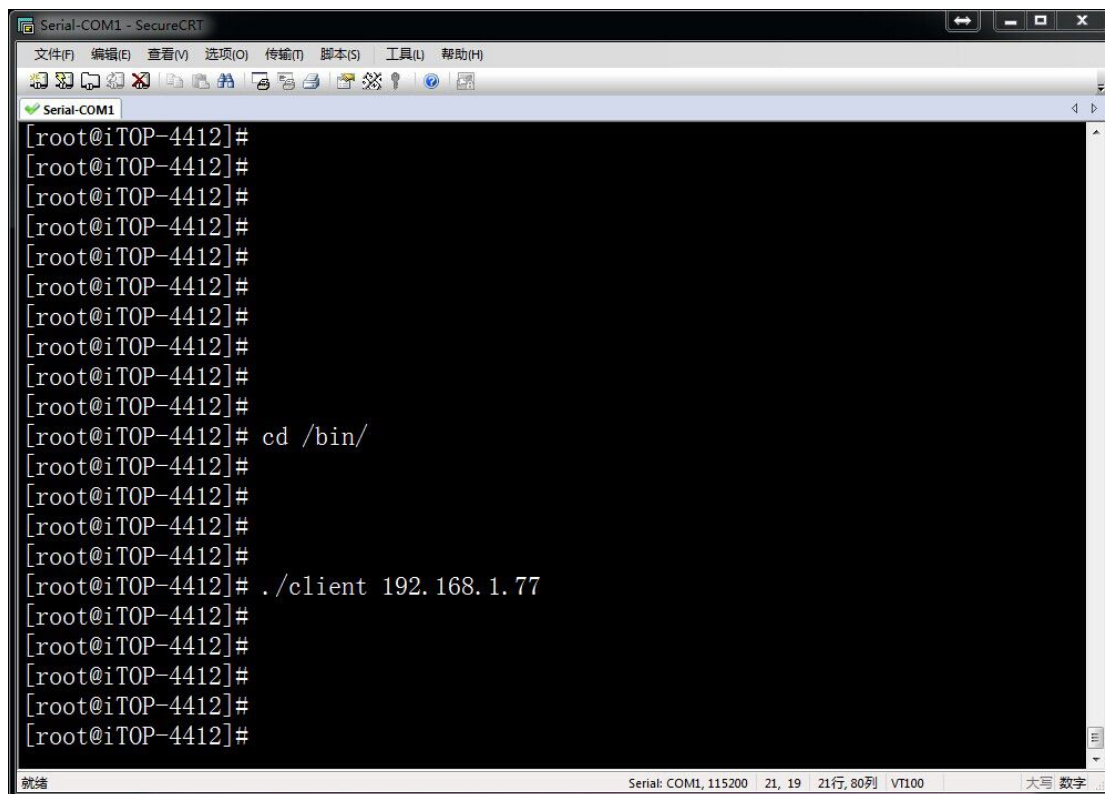
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# gcc -o server server.c
root@ubuntu:/home/topeet/linux/app/socket/udp# arm-none-linux-gnueabi-gcc
ent client.c
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# cp client /home/topeet/linux
system/bin/
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# ./server
```

在 4412 开发板上运行 client 程序，如下图：



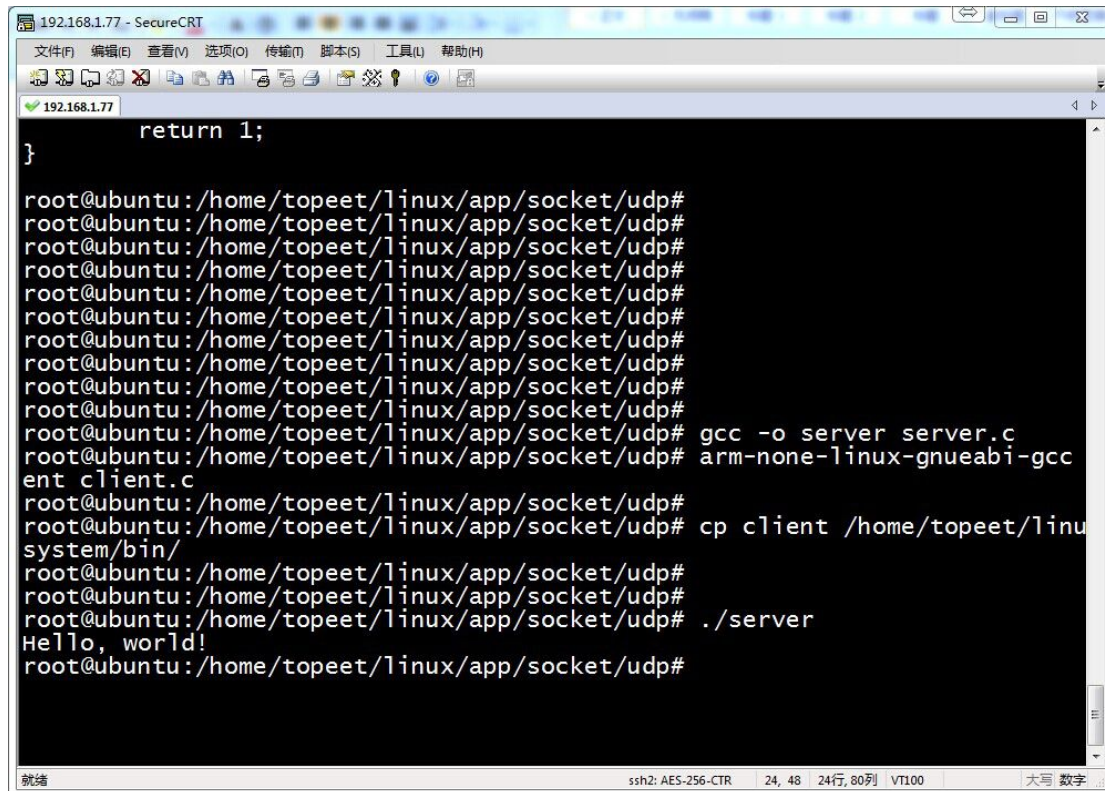
```
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]# cd /bin/
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]# ./client 192.168.1.77
```


执行结果如下图：



```
Serial-COM1 - SecureCRT
文件(F)  编辑(E)  查看(V)  选项(O)  传输(T)  脚本(S)  工具(U)  帮助(H)
Serial-COM1
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]# cd /bin/
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]# ./client 192.168.1.77
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[就绪] Serial: COM1, 115200 21, 19 21行, 80列 VT100 大写 数字
```

现在我们看下虚拟机 Ubuntu 的 server 程序的输出，如下图：



```
192.168.1.77 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(U) 帮助(H)
192.168.1.77
}
    return 1;
}

root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# gcc -o server server.c
root@ubuntu:/home/topeet/linux/app/socket/udp# arm-none-linux-gnueabi-gcc
ent client.c
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# cp client /home/topeet/linux
system/bin/
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp#
root@ubuntu:/home/topeet/linux/app/socket/udp# ./server
Hello, world!
root@ubuntu:/home/topeet/linux/app/socket/udp#
```

我们可以看到 server 程序输出 “Hello , word”，上面的打印信息就是 client 发送的数据。至此基于 UDP 的 socket 通讯就实现了。