中央任务调度任务—通信开发

技术说明

参赛学校 温州大学

队伍名称

指导老师 任明

参赛学生：董相铭 王雪奇 杨京东 闫世为 谭祖春

二零二二年

目录

1. 项目算法功能...................................................................................................3
2. 项目实践过程...................................................................................................4

3.创新点.................................................................................................................10

4.总结.....................................................................................................................10

5.参考文献............................................................................................................10

**1.项目算法功能**

我们的这个算法能够实现Windows和Ubuntu下信息传递，可以传递中文，英文，数字等信息，为后续的进程提供了保障。



ping.c(Ubuntu端源代码)

在Ubuntu端的代码中，我们设置了当接受到的字符串长度小于0时，显示Server Receive Data Failed！，当接收到的字符串以‘q’开头时，则显示Quit！后自动退出，当接收到正常的字符串时，则显示接收到的内容。



源2.cpp（Windows端源代码）

在Windows端的代码中，我们设置了当成功连接socket后显示Socket Connect Succeed! Input Data:

**2.项目实践过程**

第一步：

更改网络IP地址，将Windows端IP地址改成192.168.0.2；子网掩码自动生成为255.255.255.0。



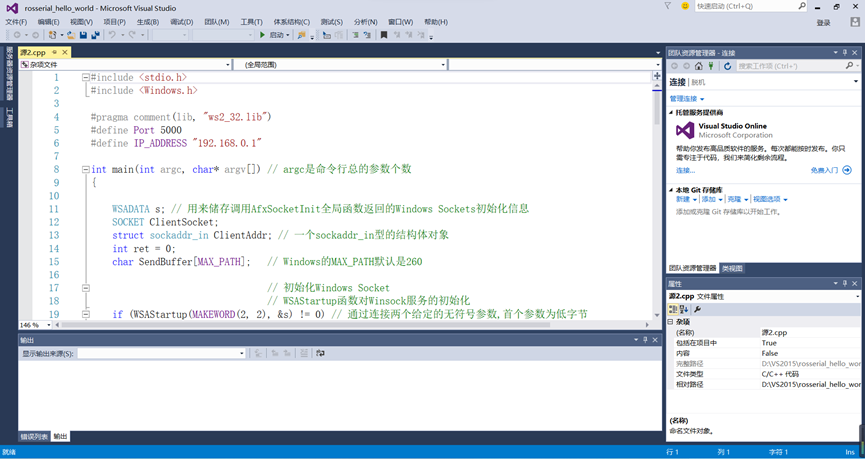
然后将Ubuntu端IP地址改成192.168.0.1；子网掩码改为255.255.255.0。

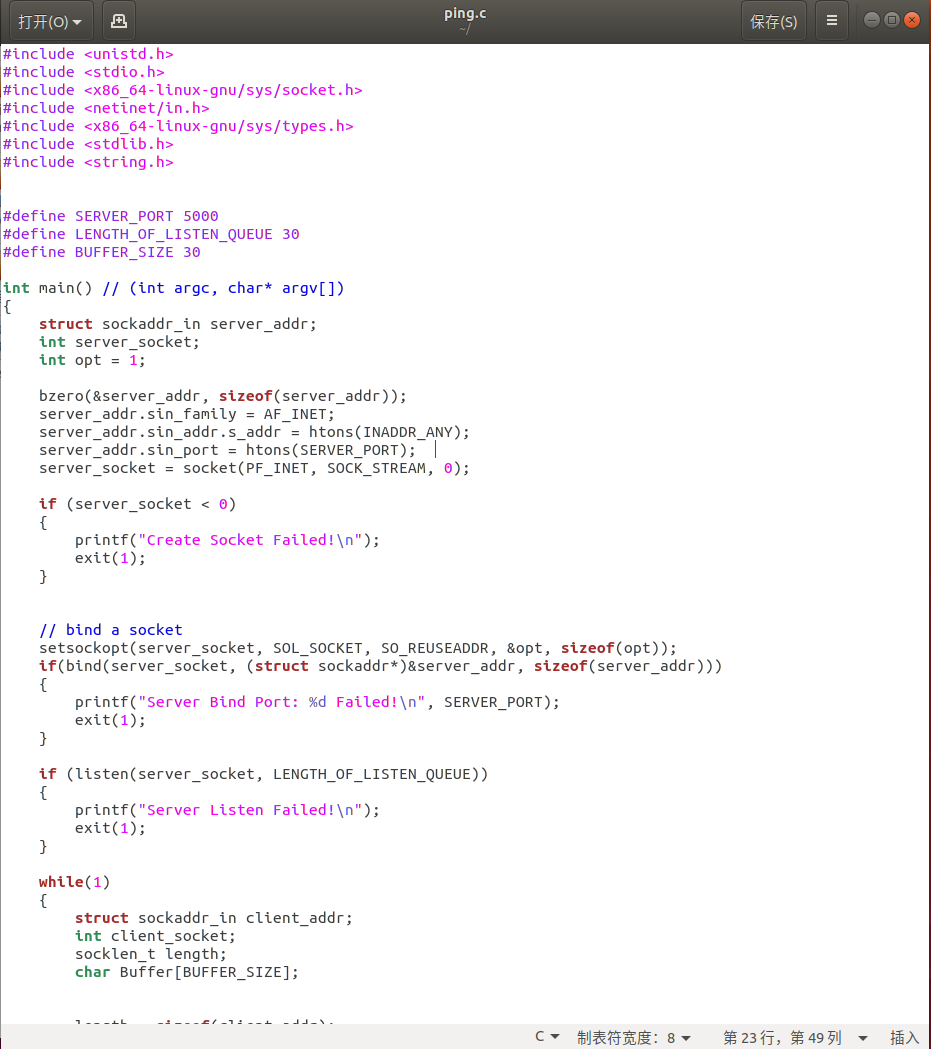


第二步：

编写服务端的代码和客户端的代码

客户端：



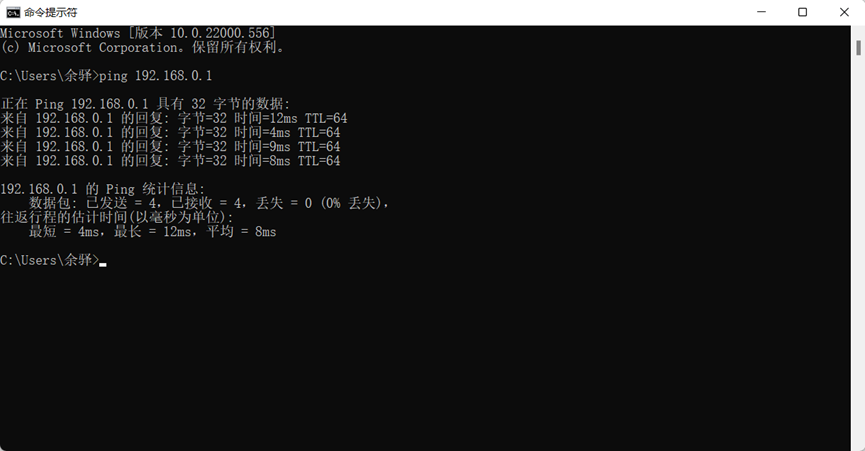
服务端：

第三步：

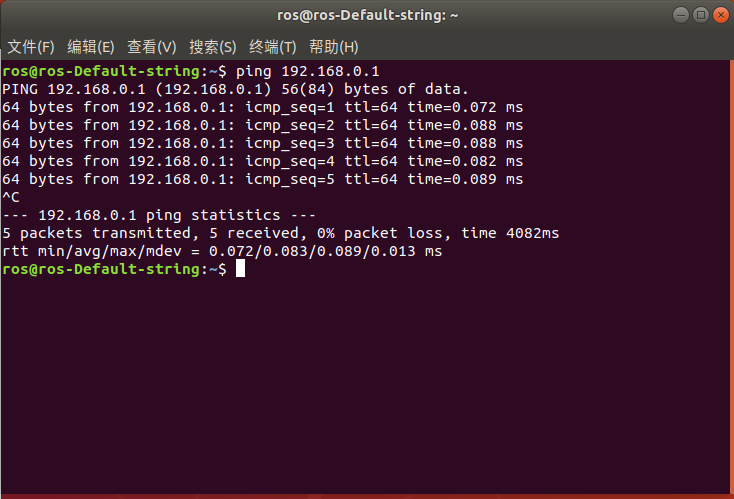
编译我们写好的代码，windows端我们用的VS2015编译，Ubuntu端我们用的gcc编译

第四步：

我们进行了通信前的测试，首先从windows端向Ubuntu端ping

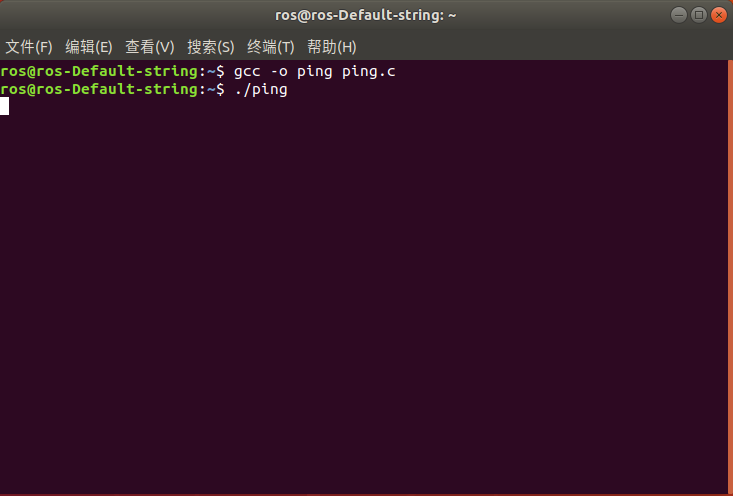


然后从Ubuntu端向windows端ping



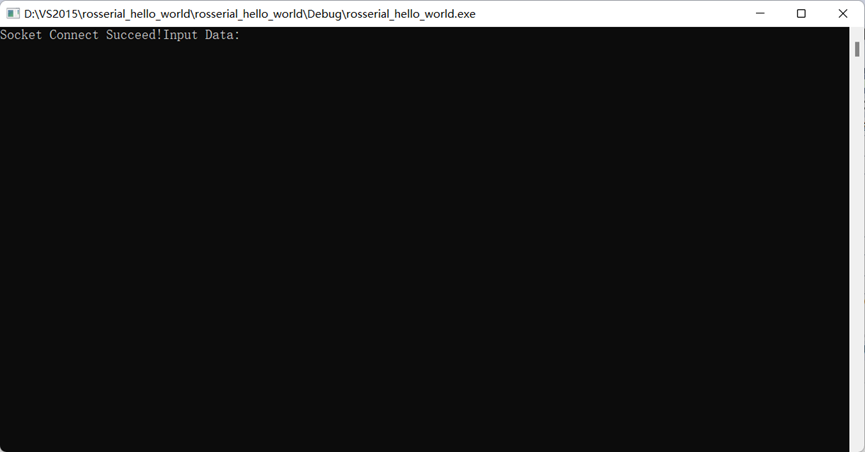
第五步：

先打开Ubuntu终端，运行我们编译的可执行程序，



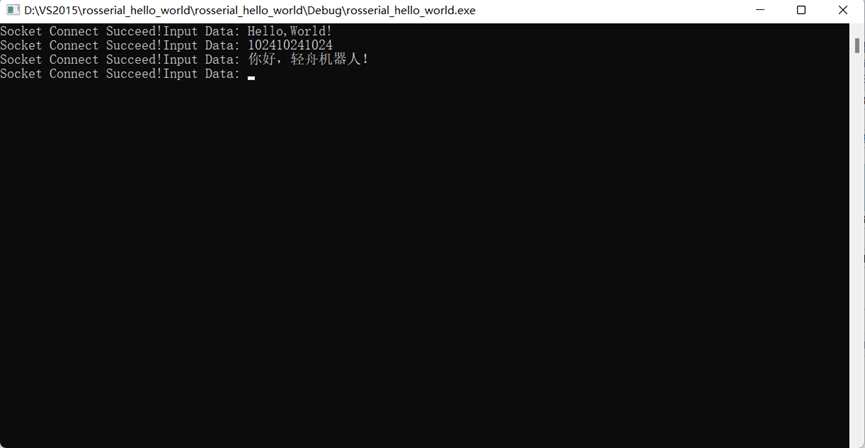
然后再打开VS2015，运行客户端程序

出现如下界面，我们成功建立起了Windows和Ubuntu的通信。

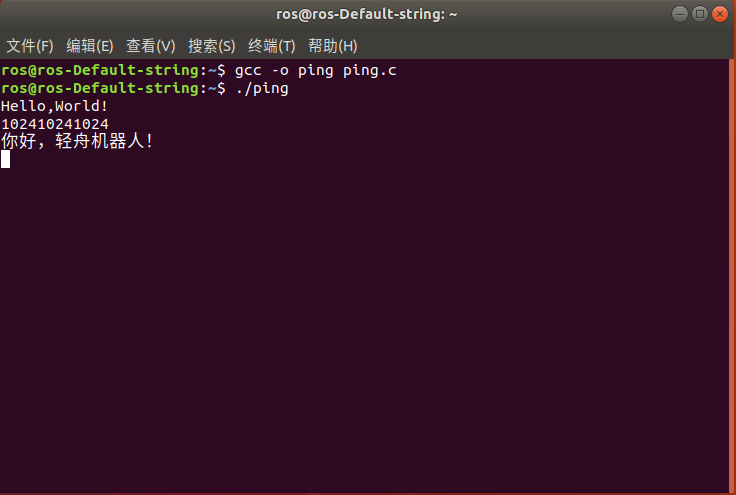


第六步：

Windows端输入我们需要传递的字符，



可以看到我们的Ubuntu端成功接收到字符。



我们的Windows与Ubuntu的通信到这就已经完全成功了。

**3.创新点**

1.我们在实践的过程中发现由Ubuntu端向Windows端ping时输入192.168.0.2并没有成功，但经过我们的摸索发现应该输入192.168.0.1才能ping成功。

2.在windows端输入中文时Ubuntu端接收到的是乱符，我们了解到的就是因为Windows与Ubuntu编码格式不一样，就只需要在终端的上面工具栏里点击终端，选择第一个设定字符编码，然后选择“简体中文--GBK”即可。

3.我们也尝试过两台Ubuntu设备之间的ssh通信，在实践过程中，我们能明显的感觉到两台Ubuntu设备之间的ssh通信要比Windows与Ubuntu的socket通信更加便捷，可以直接打开服务端的终端进行操作，而不是简单的传输数据，并且ssh有着较好的安全性。

**4.总结**

这次的Windows与Ubuntu的通信我们选择的是VS2015作为上位机的开发工具，在这之前我们也没有接触过这款开发工具，完全是看到任务后开始学习的，所以我们在完成这个任务的过程中遇到了很多问题，有一些问题是可以在网上找到解决办法的，但有的问题却找不到解决办法，甚至是我们因为自己设备的问题而遇到了别人不会遇到的错误，这个时候我们只有自己去尝试，去摸索，在一遍遍的操作后才解决这些问题，成功完成了给定的任务。这让我们学会了如何去学习一些自己以前没接触过的东西，也加强了我们的自主探索能力。

讲解视频地址如下：【window\_ros 进行通信并消息互传-哔哩哔哩】 https://b23.tv/MHCl8XS

1. **参考文献**

【1】使用socket 情况下的windows系统与ubuntu16.04系统通信详解.CSDN.Robot\_hfut

【2】 Ubuntu下进行ssh 详解.CSDN.五新