**北京航空航天大学软件学院**

**<多线程端口扫描器>**

**课程设计报告**

|  |  |
| --- | --- |
| 姓 名： |  |
| 学 号： |  |
| 班 级： |  |

**年 月 日**

**School of Software, BeiHang** **University**

目录

[1. 概述 3](#_Toc407303033)

[1.1 课程设计目的 3](#_Toc407303034)

[1.2 课程设计内容 3](#_Toc407303035)

[1.3 运行环境 3](#_Toc407303036)

[2. 系统分析与设计 3](#_Toc407303037)

[2.1 原理概述 3](#_Toc407303038)

[2.2 程序流程图 3](#_Toc407303039)

[2.3 主要数据结构 3](#_Toc407303040)

[2.4 主要算法 3](#_Toc407303041)

[2.5 主要函数说明 3](#_Toc407303042)

[3. 用户使用手册 4](#_Toc407303043)

[4. 心得体会与总结 4](#_Toc407303044)

[4.1 心得体会 4](#_Toc407303045)

[4.2 总结 4](#_Toc407303046)

[参考资料 4](#_Toc407303047)

<多线程端口扫描器>

## 概述

### 课程设计目的

加深TCP/IP协议的理解，掌握TCP四次握手机制，同时熟悉socket编程。

### 课程设计内容

实现一个端口扫描器：

（1），支持多进程/线程；

（2），能对单个指定主机进行扫描或扫描指定网段内的主机；

（3），能扫描特定的部分端口或对指定的端口段内的端口进行逐个扫描；

（4），能够显示所开放端口的服务名称；

### 运行环境

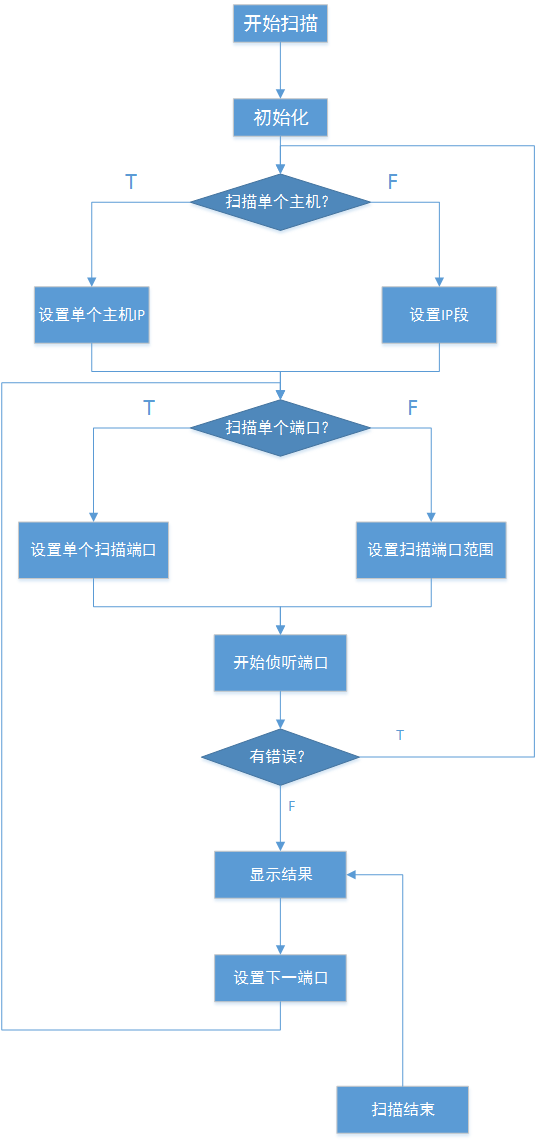
Windows xp、Windows 7、Windows8，Visual Studio2013

## 系统分析与设计

### 原理概述

端口扫描有好几种，但其中TCP connect扫描是最基本的扫描，我们可以利用系统提供的connect()用来与每一个目标计算机的端口进行连接。如果端口处于侦听状态，那connect()就能成功。否则，这个端口即是不可用的，也就是说没有提供服务。这个技术的一个最大的优点是，你不需要任何权限。系统中的任何用户都有权利使用这个调用。另一个好处就是速度，如果对每个目标端口以线性的方式，使用单独的connect()调用，那么将花费相当长的时间，使用者可以通过多线程同时打开多个套接字来加速扫描。使用非阻塞I/O允许设置一个低的时间用尽周期，同时观察多个套接字。但这种方法的缺点是很容易被察觉，并且被防火墙将扫描信息包过滤掉。目标计算机的Logs文件会显示一连串的连接和连接出错消息，并且能很快使它关闭。

### 程序流程图



说明：扫描器可实现对单个主机单个端口的扫描也可实现对网段内主机和范围内端口进行逐个扫描，因此操作时要注意是单个主机扫描还是多个IP地址扫描，以及扫描端口范围。确定扫描IP和扫描端口后点击开始即可对相应IP下的端口进行侦听，并返回端口状态，如果端口开放则同时返回端口服务，扫描结束后线程停止。

### 主要数据结构

private string ipStart;//起始IP地址

private string ipEnd;//终点IP地址

private int portStart;//开始端口

private int portEnd;//结束端口

private int numThread=20;//分配的线程数，默认最小为20

private int overTime;//超时限制

private Thread t;//定义一个线程

private Thread scanthread;//端口扫描线程

private bool[] done = new bool[65536];

List<string> str;//扫描结果集

### 主要算法

设置好IP地址和端口范围后，开辟线程，开始扫描：

利用PingReply对象试探目标主机，如果超时则表示不可达，否则连通，显示端口扫描状态，如果端口开放，显示服务。

详细请见源代码注释。

### 主要函数说明

**private void Form1\_Load(object sender, EventArgs e) //界面初始化函数**

**private void button1\_Click(object sender, EventArgs e)**//**开始按钮函数**

**public void wait()//扫描IP地址线程**

**public string State(int i)//判断端口状态函数，返回（open或closed）**

**public string Service(int i)//根据开放端口号返回具体服务**

**public void Scan(object Point)//扫描端口线程**

**private void button2\_Click(object sender, EventArgs e)//停止按钮事件函数**

**private void button3\_Click(object sender, EventArgs e)//关于按钮事件函数**

**private void trackBar1\_Scroll(object sender, EventArgs e)//超时设置函数**

**private void checkBox2\_CheckedChanged(object sender, EventArgs e)//点选按钮函数**

## 用户使用手册

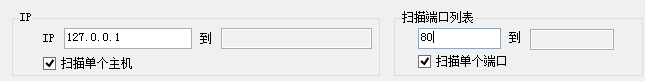
**（1），端口扫描器主界面如图所示**：



运行程序系统进入主界面，主界面中主要包括以下布局：IP地址设置、端口范围设置、线程数设置、Ping超时时间限制、扫描进度、显示扫描结果、开始停止和关于按钮。

**（2），扫描单个主机或单个端口：**

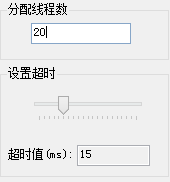
当需要扫描单个主机或者单个端口是，需要点选“扫描单个主机”“扫面单个端口”选框，这时IP范围设置和端口范围设置各自的第二个编辑框变为只读属性，无法再填写相应字段。



**（3），设置线程数和超时限制**：

由于多线程，可以分配多个线程，但为了避免消耗过多资源导致主机奔溃建议1-30；

超时限制可通过TrackBar滑动选择，默认为10-30s，注意下方的编辑框是只读属性。



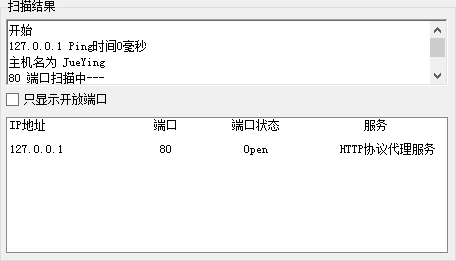
**（4）开始、停止、关于按钮：**

设置好需要扫描的IP和端口后，点击开始按钮，即可进行扫描，点击停止，将线程挂起，扫描结束。点击关于按钮，弹出个人信息及软件反馈地址。



**（5），显示扫描结果：**

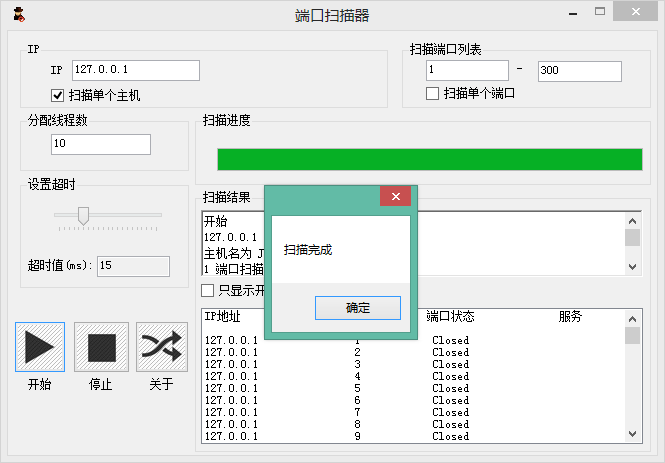
通过richTextBox控件显示扫描结果找出开放端口，ListBox控件动态显示扫描结果，包括扫描IP地址，端口号，端口状态（closed、open），以及开放端口服务名称。



## 心得体会与总结

### 心得体会

如图：对主机就127.0.0.1进行测试，符合预期结果。





根据计算机网络知识，可以看到通过对单个主机和网段内的多个主机进行端口扫描都得到了正确结果。创建一个socket，通过TCP connect()测试该主机的某个端口是否能够连通，获得该端口的状态，如果open则获知它的服务。

当然由于采用TCP connect()方法，不可避免的有它的缺陷，因为大量无需权限的访问，容易被防火墙过滤掉，因此可以进行这方面的改进。可以采用TCP SYN扫描、TCP FIN扫描、TCP反向ident扫描、FTP返回攻击等。

### 总结

通过这次课程设计，一方面我熟悉了C#方面基本的网络编程和socket编程，同时对TCP连接的过程有了更深入的理解，包括封装API的调用，握手机制等。为了解决这一过程中遇到的问题，除了课本还查阅了很多资料，对网络编程也有了不少的心得，总得来说收获满满！

## 

## 参考资料

[1]<谢希仁>.《计算机网络》[M].电子工业出版社，2013.6第6版

[2]<明日科技>.《C#从入门到精通》[M].清华大学出版社，2012.9第3版