**郑 州 轻 工 业 学 院**

**课程设计报告**

**课程名称：** 网络协议分析

**院 （系）：** 计算机与通信工程学院

**专业班级：** 网络工程15-1

**姓 名：** 杨超众

**学 号：** 541507030146

**指导教师：** 张然

2017 **年** 12 **月** 20 **日**

**课程设计任务书**

**实验名称：**发现服务器上已开启的 TCP 服务

**目的：**通过发现服务器开启的 TCP 服务，了解客户机/服务器与端口扫描 的工作原理，熟悉 TCP 协议和端口的概念。

**任务和要求：**

（1） 学习端口扫描技术，理解其工作原理。

（2） 编写程序，发现服务器正在使用的 TCP 端口号，并输出结果。

（3） 输入：服务器的 IP 地址 命令行形式运行（ScanPort 为程序名）： ScanPort server\_address

输出：服务器正在使用的 TCP 端口号

**目录**

[课程设计目的： 3](#_Toc501552587)

[课程设计基础知识： 4](#_Toc501552588)

[课程设计分析： 5](#_Toc501552589)

[课程设计设计思想 5](#_Toc501552590)

[课程设计流程图 6](#_Toc501552591)

[课程设计实验结果： 7](#_Toc501552592)

[课程设计中遇到的问题及解决方法 8](#_Toc501552593)

[课程设计收获 9](#_Toc501552594)

[参考文献 9](#_Toc501552595)

[附录 10](#_Toc501552596)

## 课程设计目的：

Internet是以客户机/服务器模式进行工作的，服务器在某些端口上开起一些网络服务，等侯客户端请求的到来。传输层提供的网络服务有TCP服务和UDP服务两种。传输层在网络体系结构中是承下启上的以层。可以把传输层下面的几层称作传输服务提供者（transport service provider），上面的几层称着传输服务使用者（transport service user）。传输层处于中间，它既使用传输服务提供者提供的服务，又为传输服务使用者提供服务。传输层是网络体系结构中最重要的一层，所以深入理解传输层的作用和协议，对于网络课程的学习极为关键。通过本课程设计的目标是熟悉TCP协议和端口的概念。

## 课程设计基础知识：

端口扫描，顾名思义，就是逐个对一段端口或指定的端口进行扫描。通过扫描结果可以知道一台计算机上都提供了哪些服务，然后就可以通过所提供的这些服务的己知漏洞就可进行攻击。其原理是当一个主机向远端一个服务器的某一个端口提出建立一个连接的请求，如果对方有此项服务，就会应答，如果对方未安装此项服务时，即使你向相应的端口发出请求，对方仍无应答，利用这个原理，如果对所有熟知端口或自己选定的某个范围内的熟知端口分别建立连接，并记录下远端服务器所给予的应答，通过查看一记录就可以知道目标服务器上都安装了哪些服务，这就是端口扫描，通过端口扫描，就可以搜集到很多关于目标主机的各种很有参考价值的信息。例如，对方是否提供FPT服务、WWW服务或其它服务

## 课程设计分析：

（1）常用的端口扫描技术有很多，如TCP connect扫描、TCP SYN扫描以及TCP FIN扫描等。

C#语言提供的TcpClient对象用来与每一个感兴趣的目标计算机的端口进行连接。如果端口处于侦听状态，那么就能定义一个TcpClient对象，连接成功。否则，这个端口是不能使用的，即没有提供服务。这个技术的一个最大优点是，不需要任何限制。系统中任何用户都有权限使用这个调用。

如果对每个目标端口以线性方式进行扫描，即用单独的TcpClient对象调用，那么扫描速度会很慢，因此我们可以利用多个线程的并行来加快扫描速度。

用C#高级语言设计，思路相对比较简单，会根据提供的参数，构造TcpClient对象，并调用其构造函数来测试连接是否可达，来判断端口是否打开

## 课程设计设计思想

●建TcpClient对象，TcpClient用于为TCP网络服务提供客户端连接

TcpClient objTCP = null;

●待扫描端口，成功则写入信息下面给出本课程设计的核心代码。

try

{

//用TcpClient对象扫描端口

objTCP = new TcpClient(IP, portnow);

lbResult.Items.Add("端口 " + portnow.ToString() + " 开放!");

}

catch

{

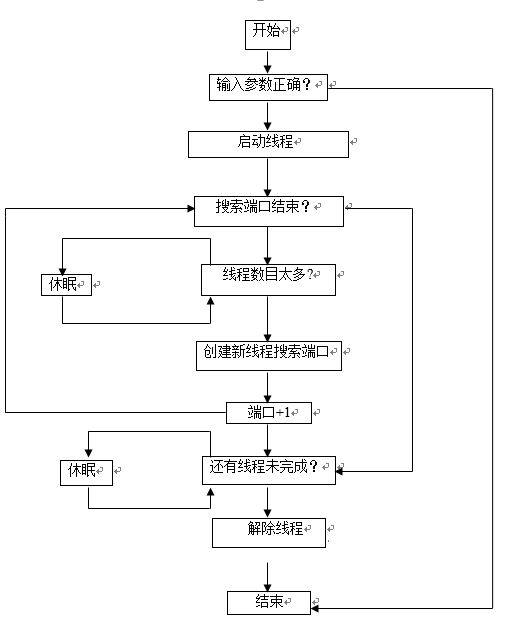
lbResult.Items.Add("端口 " + portnow.ToString() + " 未开放!");

}

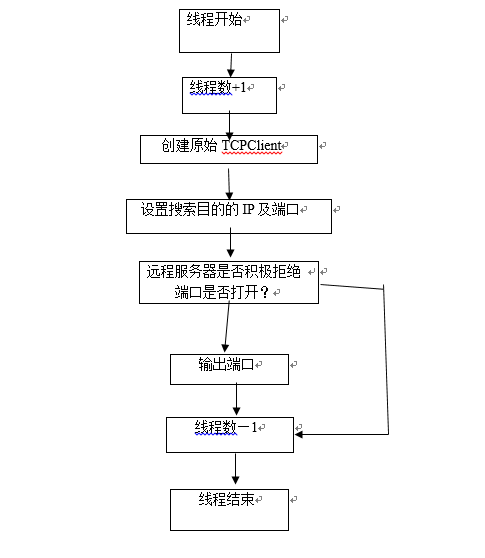
●代码中，创建TcpClient对象，把TcpClient用于为TCP网络服务提供客户端连接，创建线程，并创建ThreadStart委托对象，端口扫描范围为[0-65536]，也可自定义扫描范围。

## 课程设计流程图

主程序流程图

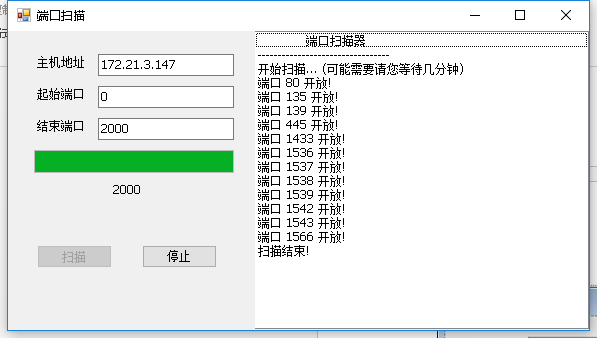


主程序使用的线程流程图



## 课程设计实验结果：

扫描172.21.3.147 的0-2000端口结果



实验结果分析：

IP地址是网络层的地址，端口是传输层的地址。计算机使用IP地址和端口号来对某个服务器进行定位。在实验源代码中使用C#提供TCPClient类,来与每一个感兴趣的目标计算机的端口进行连接。如果端口处于侦听状态即端口已在使用中，那么TcpClient对象就能新建，连接成功，否则，远程服务器积极拒绝，这个端口是不能使用的，即没有提供服务。

## 课程设计中遇到的问题及解决方法

遇到的问题

1.程序在设计初未考虑终止，只有开始扫描，直到把所有63336个端口扫描完才结束；

2.程序在设计初不能实现扫描端口段；

3.未用多线程编程，程序执行缓慢，容易卡死；

解决方法：

1. 加上停止按钮，在扫描过程中可以随时终止；
2. 加上端口段输入框，实现可以扫描任意端口段；
3. 使用多线程技术；

## 课程设计收获

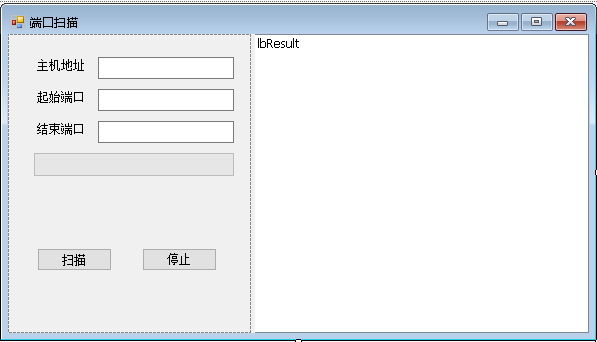
在这次课程设计中，遇到了许多的困难，但同学和老师的帮助下，还是顺利完成了。通过此次课程设计，学到了不少的东西，在此期间，我们通过网上搜索资料，逛图书馆，看视屏教程了解本次课程设计的相关知识，真的使人受益匪浅，收获颇多！通过大家的努力，在同学和指导老师的帮助下，成功的完成了本次课程设计，对与他们的帮助，我们组在此表示感谢！

## 参考文献

1. 《计算机网络》，第5版 ，谢希仁，电子工业出版社，2010.7
2. 《TCP/IP协议族》，第四版，清华大学出版社
3. 《计算机网络基础》，王寅涛、李兴元，北京理工大学出版社
4. 《C#高级编程》，第9版，清华大学出版社
5. 《计算机网络技术》，陆瑶远，高等教育出版社

## 附录

（1）软件界面设计图



（2）软件源代码

/// <summary>

/// 端口号

/// </summary>

private int port;

/// <summary>

/// IP地址

/// </summary>

private string IP;

/// <summary>

/// 数组

/// </summary>

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

/// <summary>

/// 开始端口号

/// </summary>

private int start;

/// <summary>

/// 结束端口号

/// </summary>

private int end;

/// <summary>

/// 线程

/// </summary>

private Thread scanThread;

/// <summary>

/// 是否完成

/// </summary>

private bool isOK;

/// <summary>

/// 构造函数

/// </summary>

public Form1()

{

InitializeComponent();

CheckForIllegalCrossThreadCalls = false;

}

/// <summary>

/// 扫描线程

/// </summary>

private void PortScan()

{

start = Int32.Parse(txtStart.Text);

end = Int32.Parse(txtEnd.Text);

//检查输入范围合法性

if ((start >= 0 && start <= 65536) && (end >= 0 && end <= 65536) && (start <= end))

{

lbResult.Items.Add("开始扫描... (可能需要请您等待几分钟)");

IP = txtAddr.Text;

for (int i = start; i <= end; i++)

{

port = i;

//使用该端口的扫描线程

scanThread = new Thread(new ThreadStart(Scan));

scanThread.Start();

//使线程睡眠

System.Threading.Thread.Sleep(100);

progressBar1.Value = i;

lblNow.Text = i.ToString();

}

//未完成时情况

while (!isOK)

{

isOK = true;

for (int i = start; i <= end; i++)

{

if (!done[i])

{

isOK = false;

break;

}

}

System.Threading.Thread.Sleep(1000);

}

lbResult.Items.Add("扫描结束!");

}

else

{

MessageBox.Show("输入错误，端口范围为[0-65536]");

}

}

private void Scan()

{

int portnow = port;

//创建线程变量

Thread Threadnow = scanThread;

done[portnow] = true;

//创建TcpClient对象，TcpClient用于为TCP网络服务提供客户端连接

TcpClient objTCP = null;

//扫描端口，成功则写入信息

try

{

//用TcpClient对象扫描端口

objTCP = new TcpClient(IP, portnow);

lbResult.Items.Add("端口 " + portnow.ToString() + " 开放!");

}

catch

{

//lbResult.Items.Add("端口 " + portnow.ToString() + " 未开放!");

}

}

public Thread process;

private void btnScan\_Click\_1(object sender, EventArgs e)

{

btnScan.Enabled = false;

if (txtStart.Text.ToString() == "")

{

MessageBox.Show("IP地址不能为空");

return;

}

//创建线程，并创建ThreadStart委托对象

process = new Thread(new ThreadStart(PortScan));

//线程开始

process.Start();

//显示端口扫描的范围

progressBar1.Minimum = Int32.Parse(txtStart.Text);

progressBar1.Maximum = Int32.Parse(txtEnd.Text);

//显示框初始化

lbResult.Items.Clear();

lbResult.Items.Add(" 端口扫描器 ");

lbResult.Items.Add("---------------------------------");

}

private void button1\_Click(object sender, EventArgs e)

{

process.Abort();

btnScan.Enabled = true;

// progressBar1.Value = 0;

}

**课程设计报告成绩评定表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评定项目 | 内 容 | 满 分 | 评 分 | 总 分 |
| 平时表现 | 学习认真，态度端正，遵守纪律，按时出勤。 | 10 |  |  |
| 系统实现 | 具有良好的编程习惯，能按要求完成各种功能或操作，运行结果正确。 | 30 |  |
| 答辩 | 思路清晰，讲解清楚，回答问题准确。 | 20 |  |
| 课设报告 | 报告字迹整洁、书写规范、条理清楚；图、表、文字表达准确规范，上交及时。 | 40 |  |
| 总成绩 | 采用五级分制：优、良、中、及格、不及格 |  | | |
| **评语：**  **指导老师签字：**  **年 月 日** | | | | |