

东南大学自动化学院

实 验 报 告

课程名称：_____信息通信网络概论_____

第 1 次实验

实验名称：_____TCP/IP 协议与 Socket 编程认识实验_____

院（系）：_____自动化_____专 业：_____自动化_____

姓 名：_____陈鲲龙_____学 号：_____08022311_____

实 验 室：_____金智楼_____实验组别：_____

同组人员：_____实验时间： 2024 年 5 月 8 日

评定成绩：_____审阅教师：_____

目 录

一. 实验目的和要求.....	3
二. 实验原理.....	3
三. 实验方案与实验步骤.....	5
四. 实验设备与器材配置.....	5
五. 实验记录.....	5
六. 实验总结.....	7
七. 思考题或讨论题.....	7

实验报告内容:

一. 实验目的和要求

1. 实验目的:

- 1) 学习使用 MFC 制作界面
- 2) 学会利用已有函数添加进入 MFC 内容中
- 3) 了解学习 DNS 域名解析协议的机制

2. 实验要求:

- 1) 理解掌握 WSStartup()、WSACleanup() 函数的使用; 知道 WSADATA、HOSTENT 结构成员表示意义和 inet_ntoa()、inet_addr() 函数的使用
- 2) 使用 Socket 函数编写一个小程序实现查询主机地址即输入 IP 地址能够给出该 IP 地址对应的域名; 给出域名可以得到 IP 地址。

二. 实验原理

1. MFC:

微软基础类库(Microsoft Foundation Classes, 简称 MFC)是一个在 Microsoft Windows 操作系统上开发桌面应用程序的 C++类库。它为开发者提供了一组面向对象的类、函数和工具, 用于简化 Windows 应用程序的开发过程。MFC 采用了面向对象的设计理念, 提供了许多类和框架, 使开发者能够使用类的继承、封装和多态等特性来构建复杂的 Windows 应用程序; MFC 封装了许多 Windows API, 使得开发者无需直接操作底层 API, 而是通过调用 MFC 提供的类和方法来实现对 Windows 操作系统的操作和管理; MFC 特别适用于开发图形用户界面(GUI)应用程序, 它提供了丰富的控件(如按钮、文本框、列表框等)和视图(如文档视图模型), 帮助开发者轻松构建复杂的用户界面; MFC 支持事件驱动编程模型, 开发者可以通过响应用户操作(如按钮点击、菜单选择等)来执行相应的操作和处理; MFC 提供了对 Windows 资源(如图标、位图、对话框模板等)的管理和使用, 使得应用程序的开发更为高效; MFC 引入了多文档视图架构(Document-View Architecture), 使得应用程序能够处理多个文档和视图, 支持复杂的文档编辑和管理; MFC 提供了一些数据处理类, 支持数据库访问和数据绑定, 使得开发数据库驱动的应用程序更为便捷; MFC 设计时考虑了 Windows 操作系统的兼容性和未来的扩展需求, 因此应用程序可以比较轻松地迁移到新的 Windows 版本上, 而不需要大规模的重写。MFC 作为一个经典的 C++类库, 曾经是 Windows 桌面应用程序开发的主流选择, 直到后来 WPF 和 UWP 等技术的出现。尽管如此, MFC 仍然被广泛使用于一些传统的 Windows 应用程序开发中, 特别是需要与底层 Windows API 直接交互的应用程序。

2. WSStartup 函数:

E.g: `int WSStartup (WORD wVersionRequested, LPWSADATA lpWSADATA);`

本函数就可以用于初始化的目的, 使用 Socket 的程序在使用 Socket 之前必须调用 WSStartup 函数。该函数的第一个参数指明程序请求使用的 Socket 版本, 其中高位字节指明副版本、低位字节指明主版本; 操作系统利用第二个参数返回请求的 Socket 的版本信息。当一个应用程序调用 WSStartup 函数时, 操作系统根据请求的 Socket 版本来搜索相应的 Socket 库, 然后绑定找到的 Socket 库到该应用程序中。以后应用程序就可以调用所请求的 Socket 库中的其它 Socket 函数了。系统申请使用 wVersionRequested 版本的 Socket, 并将返回的 Socket 版本信息存储在 lpWSADATA 中, 以调用所请求的 Socket 库中的其他 Socket 函数。若申请成功, 返回 0; 否则返回错误代码之一。

3. WSACleanup 函数:

E.g: `int PASCAL FAR WSACleanup (void);`

应用程序或 DLL 在成功调用 `WSAStartup()` 函数使用 Windows Sockets 服务之后, 必须调用 `WSACleanup()` 将其从 Windows Sockets 的实现中注销, 并且该实现释放为应用程序或 DLL 分配的任何资源。

4. HOSTENT 结构体:

```
struct hostent {
```

```
char FAR * h_name;
```

正规的主机名字 (PC)

```
char FAR * FAR * h_aliases;
```

一个以空指针结尾的可选主机名队列

```
short h_addrtype;
```

返回地址的类型, 对于 Windows Sockets, 总是 `PF_INET`

```
short h_length;
```

每个地址的长度 (字节数), 对应于 `PF_INET` 为 4

```
char FAR * FAR * h_addr_list;
```

以空指针结尾的主机地址列表, 以网络顺序排列

```
};
```

5. Gethostbyname 函数:

E.g: `struct hostent FAR *PASCAL FAR gethostbyname(const char FAR * name);`

用于返回对应于给定主机名 `name` 的包含主机名字和地址信息的 `hostent` 结构指针。

使用实例: `lpHostEnt = gethostbyname((LPSTR)T2A(m_Url1));`

由对话框中输入主机名 `m_Url1` 得到对应的包含主机 `ip` 的结构变量 `lpHostEnt` (其中 `ip` 保存在 `lpHostEnt->h_addr_list` 成员中)。若输入的是 `www.seu.edu.cn`, 得到包含对应 IP 为 `11000000001000000100000000000010` 的结构变量。

6. inet_ntoa 函数:

E.g: `char FAR* PASCAL FAR inet_ntoa(struct in_addr in);`

用于将一个用 `in` 参数所表示的 Internet 地址结构转换成以“.”间隔的诸如“a.b.c.d”的字符串形式

使用实例: `m_IP1 = inet_ntoa(*(LPIN_ADDR)*(lpHostEnt->h_addr_list));`

把 `lpHostEnt` 结构中成员变量 `h_addr_list` 包含的网络位顺序的 IPv4 地址转化为常见的点间隔格式 IP 以供显示; 若输入的是 `11000000001000000100000000000010`, 则得到 `192.32.64.2`。

7. inet_addr 函数:

E.g: `unsigned long PASCAL FAR inet_addr(const struct FAR* cp);`

用于将一个点间隔地址字符串 `cp` 转换成一个以网络字节顺序表示的 32 位二进制 IP 地址。

使用实例: `dwIPAddr = inet_addr((LPSTR)T2A(m_IP2));`

由对话框中输入的点间隔格式的 IP 地址 `m_IP2` 转化为一个以网络字节顺序表示的 32 位二进制 IP 地址 `dwIPAddr`; 若输入 `192.32.64.2`, 转化后为 `11000000000100000010000000000010`。

8. gethostbyaddr 函数:

E.g: `struct hostent FAR *PASCAL FAR gethostbyaddr(const char FAR * addr, int len, int type);`

用于返回对应于给定地址的包含主机名字和地址信息的 `hostent` 结构指针。

其中 addr: 指向网络字节顺序地址的指针; len: 地址的长度, 在 PF_INET 类型地址中为 4; type: 地址类型, 应为 PF_INET。

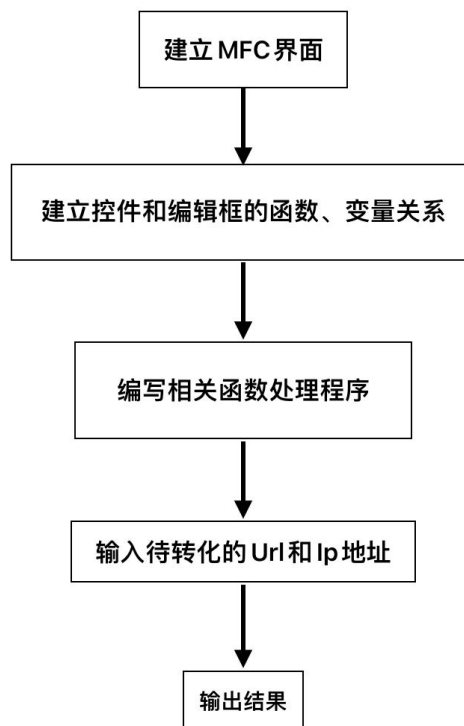
使用实例: `lpHostEnt = gethostbyaddr((LPSTR)&dwIPAddr, 4, PF_INET);`
把 IP 地址 dwIPAddr 转化为对应的包含网络名称的结构变量 lpHostEnt; 若输入的是 110000000001000000010000000000010, 得到 www.seu.edu.cn。

三. 实验方案与实验步骤

1. 实验方案:

- 1) 利用 VC6.0 建立以自己名字命名的 MFC 工程
- 2) 设置 MFC 界面, 并建立控件和函数、编辑框和变量之间的联系
- 3) 填充相应的函数命令, 实现 url 和 ip 之间的转换

2、流程图:



四. 实验设备与器材配置

- 1、 VC6.0

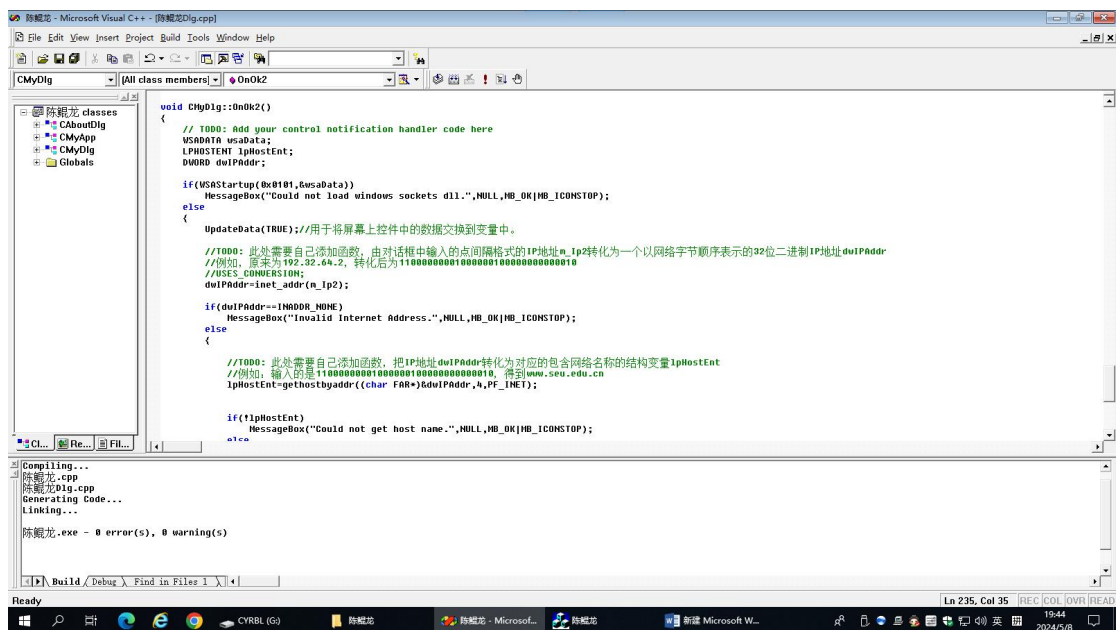
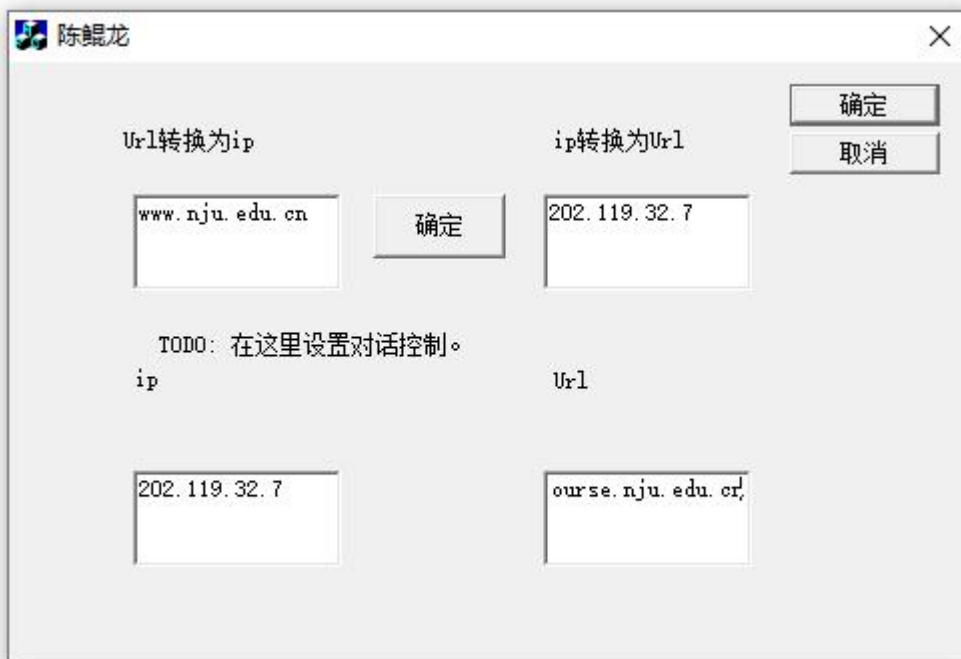
五. 实验记录

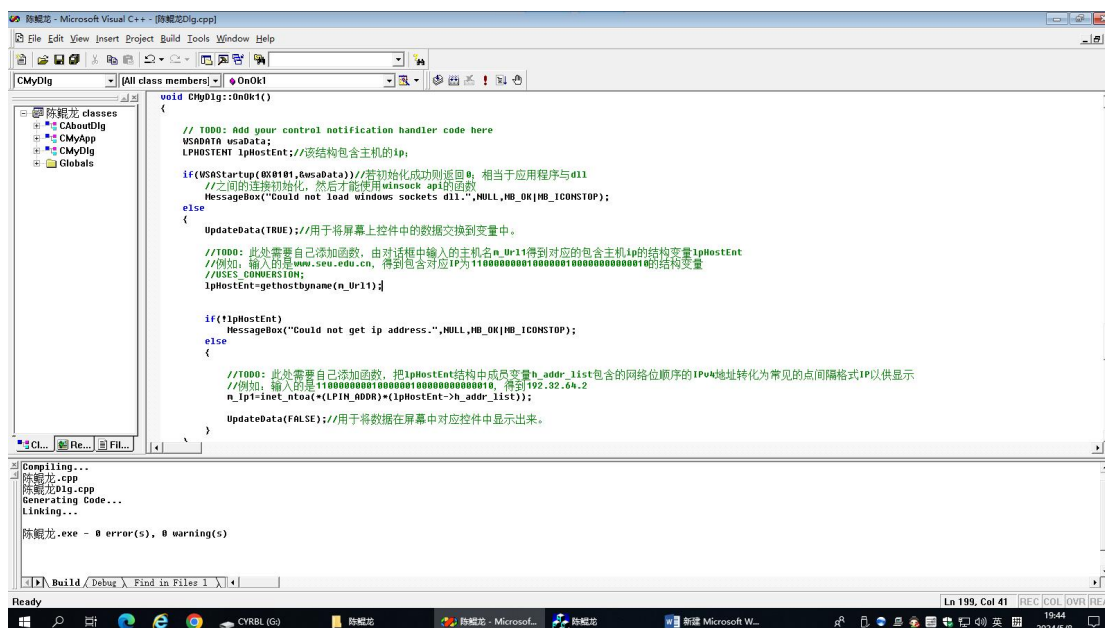
- 1、 输入 www.nju.edu.cn

输出 202.119.32.7;

输入 202.119.32.7

输出 www.nju.edu.cn





六. 实验总结

1、例程结果分析：

1) 输入 url: bbs.seu.edu.cn, 得到 IP: 58.192.114.8;

输入 IP: 58.192.114.8, 返回相同的 url: bbs.seu.edu.cn

2) 输入 url: www.seu.edu.cn, 得到 IP: 121.248.63.91;

输入 IP: 121.248.63.91, 返回不同的 url: wpidc91.seu.edu.cn。

一个域名对应一个 IP 地址，一个 IP 地址可以绑定多个域名。

2、本次实验在搜寻资料的过程中对 `socket` 函数有了更全面的理解：例如本次试验中使用 `inet_ntoa()`、`inet_addr()` 函数时，发现需要对参数进行强制类型转换，转换成功后才能成功运行。

七. 思考题或讨论题

1. WinInet 有什么作用？

WinInet 是微软开发的一个库，可以完成 http ftp 客户端的工作，功能比较稳定，可以让程序员减少工作量。Win32 Internet 扩展提供了 HTTP、FTP 和 Gopher 的访问。借助于 WinInet 编程接口，我们不必去了解这些特定协议的细节就可以编写出 Internet 客户端程序。WinInet 为这些协议提供了函数集。利用这些函数集，大大简化了编程，从而轻松地将 Internet 集成到自己的应用程序中。WinInet 的主要功能和作用包含：

1. 网络资源访问：WinInet 允许应用程序通过 HTTP、FTP 等协议访问网络上的资源，如网页、文件、目录等。这使得开发者可以轻松地实现网络数据的下载和上传；
2. HTTP 通信：支持 HTTP 协议，包括 HTTP 请求的发送和接收，允许应用程序与 Web 服务器进行通信，从而实现 Web 内容的获取和交互；
3. FTP 操作：提供 FTP 协议的支持，使应用程序能够连接到 FTP 服务器，进行文件的上传和下载，以及文件和目录的管理；
4. Gopher 协议：虽然现在不太常用，但 WinInet 也支持 Gopher 协议，允许应用程序通过 Gopher 服务器访问信息资源；
5. 缓存管理：WinInet 提供了对网络数据的缓存管理功能，可以有效地处理数据的存储和访问，从而提升应用程序对网络资源的访问效率；
6. 安全功能：支持 SSL/TLS 协议，提供加密通信的功能，确

保网络传输的安全性；7. 代理服务器支持：允许应用程序通过代理服务器访问互联网资源，支持代理服务器的配置和管理。WinInet 主要用于 Windows 桌面应用程序，如浏览器、下载管理器、自动更新程序等，它为开发者提供了一种方便且高效的方式来实现与互联网的交互操作。

2. 结合本次实验，谈谈使用 MFC WinInet 类有哪些优点？

MFC WinInet 类内置函数多且功能强大，操作方便。MFC 的 WinInet 类封装了原始 WinInet API，提供了更高级别的抽象接口和更简单的方法调用，使得开发者可以更快速实现互联网功能而无需深入了解每一个 API 调用的细节；WinInet 类库也是基于面向对象设计的，提供了类、方法和属性，使得代码更具可读性和可维护性。开发者可以通过调用类的方法来实现与互联网资源的交互，而不是直接调用繁琐的原始 API 函数。