计算机网络编程
实验报告

班级：07111707

组长：1120171189 崔程远

成员：1120172149 吴沁璇

1120172153 张澈

1120172163 王晓媛

1120172736 张鉴昊

1120172765 曾煜瑾

1120173326 曾紫飞

北京理工大学

计算机学院

2020 年 5 月

第六章 实验 1　UDP 协议服务器和客户

1. 实验目的

学习 UDP 通信协议，掌握 UDP socket 通信方法。

2. 实验内容

客户发送命令行文本给服务器，服务器转换大写后返回给客户并显示。

配置文件关键要点：

无，对方的 IP 地址、端口以及发送串以命令行参数的形式提供程序运行

3. 实验原理

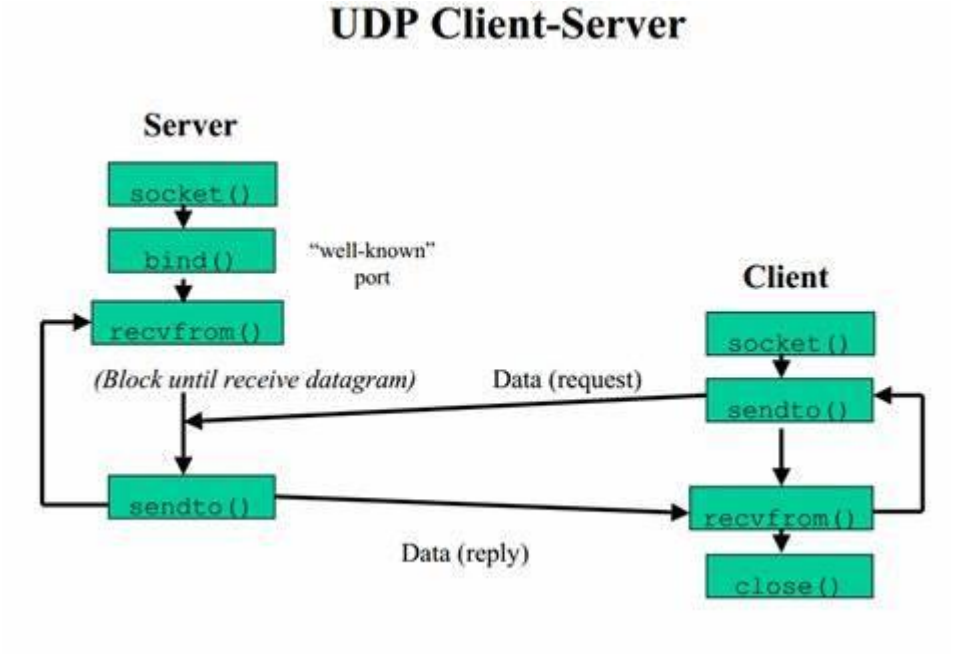
socket 起源于 Unix，而 Unix/Linux 基本哲学之一就是“一切皆文件”，都可以用“打开 open –> 读写 write/read –> 关闭 close”模式来操作。我的理解就是 Socket 就是该模式的一个实现，socket 即是一种特殊的文件，一些 socket 函数就是对其进行的操作（读/写 IO、打开、关闭）。

UDP 编程的服务器端一般步骤是：

1. 创建一个 socket，用函数 socket()；
2. 设置 socket 属性，用函数 setsockopt();* 可选
3. 绑定 IP 地址、端口等信息到 socket 上，用函数 bind()；
4. 循环接收数据，用函数 recvfrom()；
5. 关闭网络连接；

UDP 编程的客户端一般步骤是：

1. 创建一个 socket，用函数 socket()；
2. 设置 socket 属性
3. 设置对方的 IP 地址和端口等属性；
4. 发送数据，用函数 sendto()；
5. 关闭网络连接；



图源网络

4. 实验环境

语言	集成开发环境	编译器
C++	Visual Studio 2017	gcc version 4.8.1
Java	Eclipse 2019	java version "1.8.0_65"
Python	Pycharm 2017	Python 3.7.0

5. 实验步骤

各个语言的实现都分为服务端和客户端。

• C 语言版本

服务端首先判断接收参数数量是否符合要求，之后构建 socket 并绑定到本地相应端口；通过 resvMsg(server_fd, &str, &cli_addr)函数接收来自客户端的字符串并获得客户端的信息；通过 sendMsg(server_fd, str, cli_addr)函数和之前获得的客户端信息向客户端发送经过处理的字符串。

客户端首先判断接收参数数量是否符合要求，之后构建 socket 并绑定到本地相应端口（客户端实际可以不需要绑定），通过 sendMsg(server_fd, str, cli_addr)函数向服务端发送待处理的字符串；通过 resvMsg(server_fd, &str, &cli_addr)函数接收处理过的字符串。

由于刚接触 socket 编程这部分实现较为冗杂，后续 Java 和 Python 版本更简洁一些。

• Java 语言版本和 Python 语言版本

服务端判断参数数量后创建 Socket 对应语言支持的 socket 对象，通过该对象的方法来接收和发送字符串。

Java 中 DatagramPacket 的 getSocketAddress()方法可用于获得包的来源地址，用于确定客户端地址。

Python 中 socket 的 recvfrom()方法同时返回通信内容和来源地址信息，可据此确定客户端地址。

6. 运行结果

Send 函数中将 01100000000000111110000000000110 作为待发送的数据信息。

• C

服务端：输入开放的端口号

```
[chez@chez-laptop C]$ ./UDPserver 11918
Waiting for message
Message recived
Message:sdfsaffgh
Sending message
Message sent
```

客户端：依次输入目标 IP、本地开放端口、目标端口、待处理字符串

```
[chez@chez-laptop C]$ ./UDPclient 127.0.0.1 11900 11918 sdfsaffgh
Sending message
Message sent
Waiting for message
Message recived
toUpper:SDFSAFFGH
```

• Java

服务端：输入开放的端口号

```
[chez@chez-laptop Java]$ java UDPserver 10000
Waiting for message
Message received
Message:sdfasfd
Sending message
Message sent
```

客户端：依次输入目标 IP、本地开放端口、目标端口、待处理字符串

```
[chez@chez-laptop Java]$ java UDPclient 127.0.0.1 9182 10000 sdfasfd
Sending message
Message sent
Waiting for message
Message received
toUpper:SDFASFD
```

• Python

服务端：输入开放的端口号

```
[chez@chez-laptop Python]$ python UDPserver.py 10010
Waiting for Message
Message recived
Message:lsdajfasdfio
Sending message
Message sent
```

客户端：依次输入目标 IP、目标端口、待处理字符串

```
[chez@chez-laptop Python]$ python UDPclient.py 127.0.0.1 10010 lsdajfasdfio
Sending message
Message sent
Waiting for Message
Message recived
toUpper:LSDAJFASDFIO
```

7. 实验总结

实验内容相对简单，但由于对 Socket 的使用还不熟练，在第一个程序（C 版本）的编写过程中把简单问题复杂化了，之后的编写就很轻松。由于各种语言对 UDP socket 的实现机制不同，不同系统对 socket 的支持也不同（POSIX 标准和 Winsocket），因此需要查阅相关资料。

总的来说，这个实验使我加深了对 UDP 通信机制的理解，提高了编程能力。