**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**

**1120172733 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**

**北京理工大学**

**第六章 实验3 UDP校验和计算程序**

**1. 实验目的**

UDP校验和计算

**2. 实验内容**

计算UDP校验和，UDP段从wireshark中截获一个作为计算基础放入配置文件。程序程序运行屏幕输出要点：

输出伪首部各字段值

输出UDP首部各字段值

计算产生显示UDP校验和

**3. 实验原理**

用户数据报UDP有两个字段：数据字段和首部字段。首部字段很简单，只有8个字节，有四个字段组成，每个字段的长度都是两字节。各段意义如下：

源端口：源端口号。在需要对方回信时选用。不需要时可用全0。

目的端口：目的端口号。这在终点交付报文时必须要使用到。

长度：UDP用户数据报的长度，其最小值是8（仅首部）。

校验和：检测UDP用户数据报在传输中是否有错。有错就丢弃

**4. 实验环境**

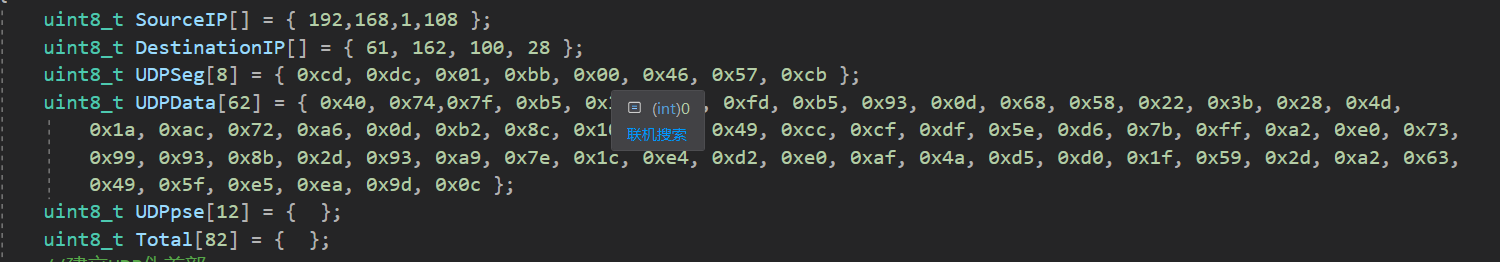
Windows 10 Visual Studio 2019 Pycharm Intellij

**5. 实验步骤**

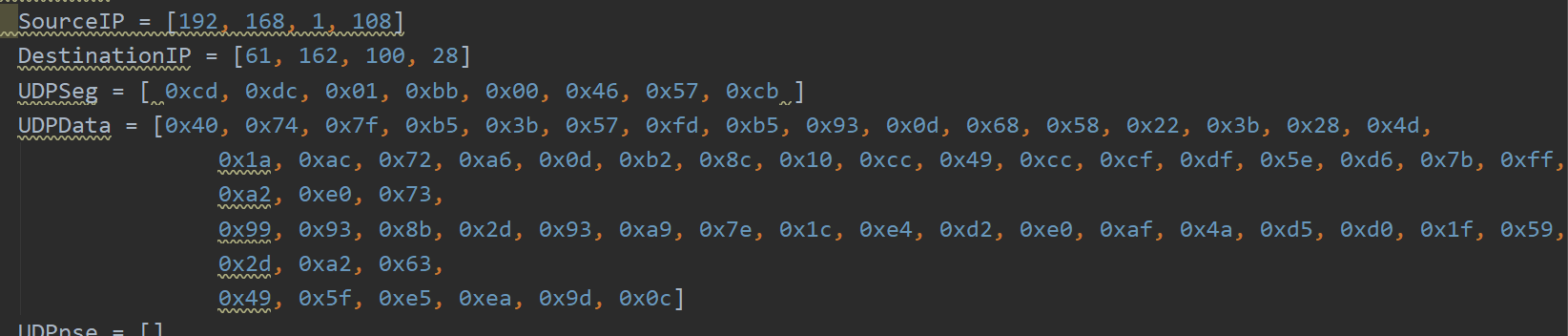
三个平台或语言的主要流程相似。

* 首先初始化UDP首部数组、数据数组、源IP和目的IP，其中存放UDP首部和数据的16进制值。

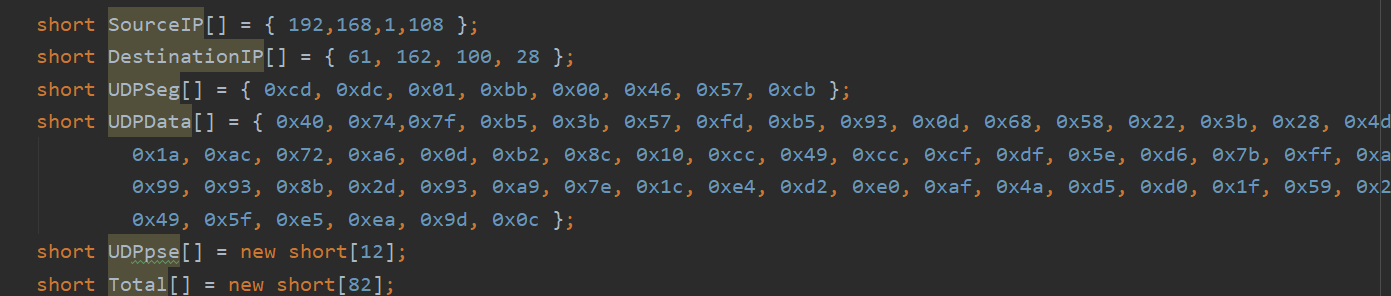
C语言：



Python：

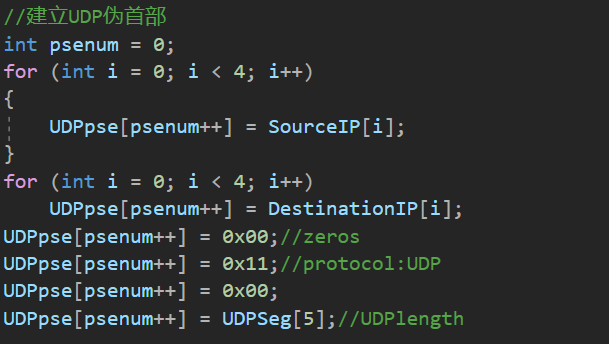


Java：

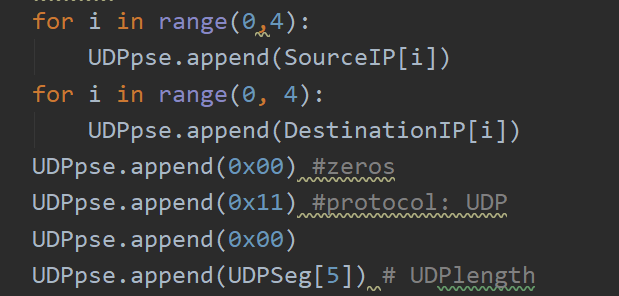


* 之后结合源IP和目的IP以及Protocol段和UDP段长度构建UDP伪首部数组

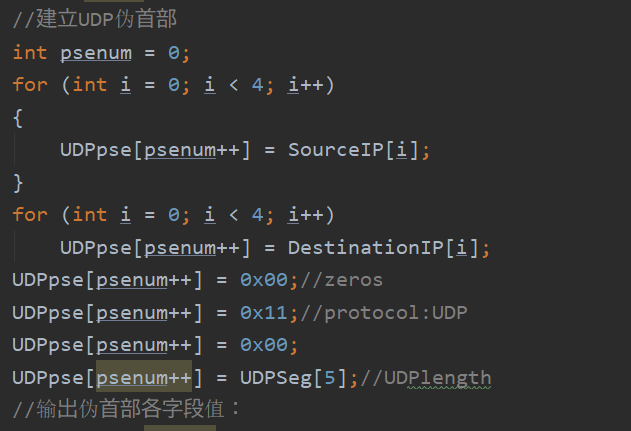
C语言：



Python：



Java：



* 通过计算出伪首部和首部的对应字节的用途进行输出。



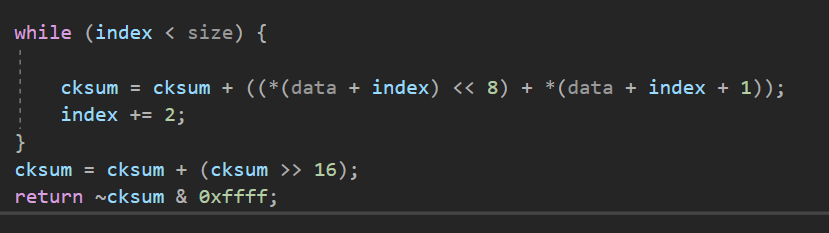
上图中上半部分为UDP伪首部，下半部分为UDP首部，代码中数组或列表中一个项代表一字节，按照上表计算得出对应字节的作用和字段的值。

以下仅放出python代码，具体其他代码在文件中可看

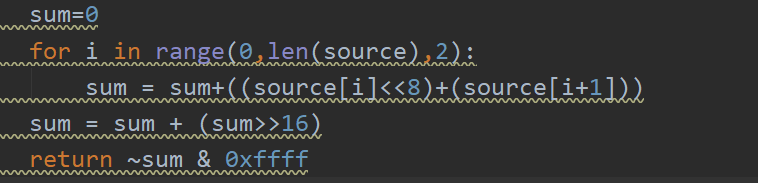


* 对于计算校验和，本实验将三部分字段放到Total数组中，将原有checksum部分置0后，将数组中两两组成16位进行累加，循环计算得到的值加上左移16位后的值用于加进位，最后取反得到checksum。其中与0xfffff与是为了将16位以外的数清零。

C语言：



Python：

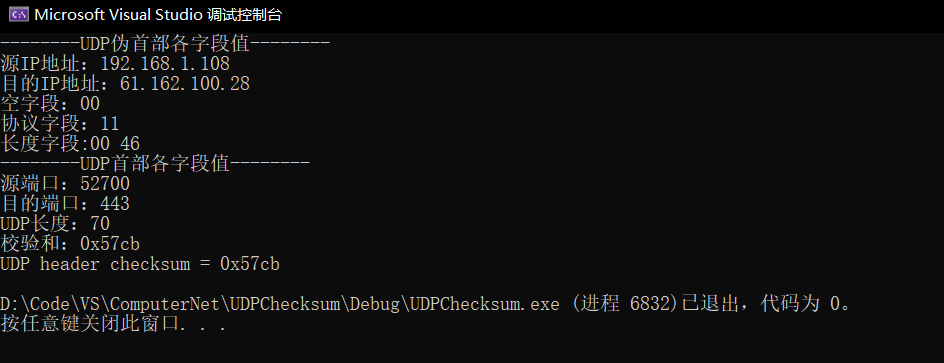


Java：

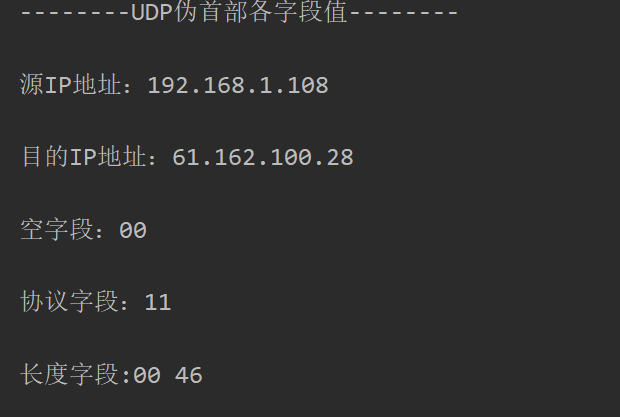


* 实验效果：

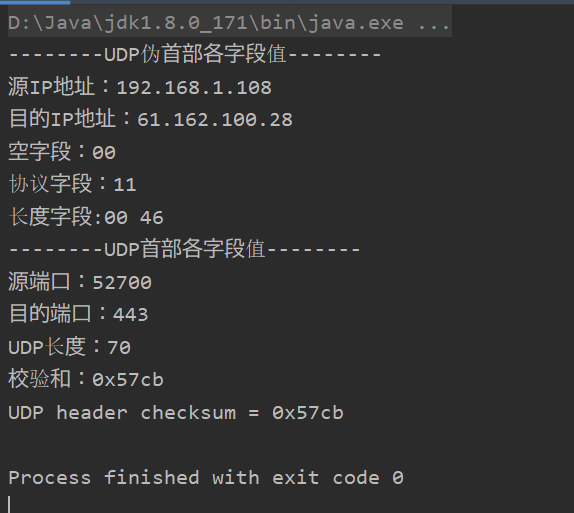
C语言：



Python：



Java：



可见结果中输出了首部和伪首部的相关字段，且校验和计算一致

**6. 实验总结**

本次实验实现了三种语言进行UDP校验码的计算以及UDP首部和伪首部的对应字段。在实验中我熟悉了UDP各段的含义和UDP校验码的计算，以及使用C语言、python和Java三种语言进行位运算；实验中也遇到了不少问题和困难，如如何巧妙地通过数组中存放的字节计算校验和等，通过和同学的讨论以及GitHub上相关代码的阅读解决了这个问题，熟悉了位运算如左右移、按位与、非等运算。本次实验使得我计算机网络相关知识的学习更加扎实，编程能力得到提高。