

人工智能学院

计算机网络

实 验 报 告

|  |  |
| --- | --- |
| **实验题目：** | CRC编程实现 |
| **学号、姓名：** | 202435710252钟润柯、202435710118练沛良 |
| **实验日期：** | 2024年9月26日 |
| **实验环境：** | Windows11电脑、vscode、gcc |

一、实验目的

掌握网络的基本知识，对常用的通信协议有深入的了解，理解CRC校验算法的原理，并且有思路用代码实现。

二、实验内容与完成情况：

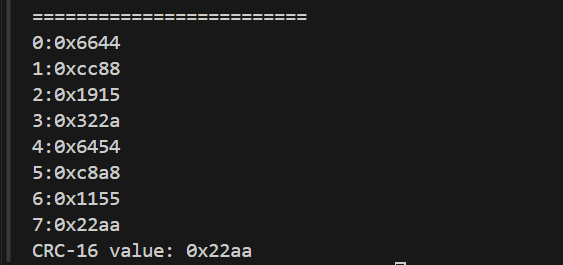
实验内容：

1.设计CRC校验算法（生产多项式为CRC-16：x16+x15+x2+1）；

2.运用VC++或者C++或者其他熟悉的编程语言和环境实现CRC校验；

完成情况：

|  |
| --- |
| def crc16(data: bytes, poly=0x8005, init=0xFFFF):      """      计算输入数据的CRC-16校验值      """      # 初始化寄存器      crc = init      for byte in data:          # print(byte)          # 将当前字节左移8位并与CRC寄存器异或          crc ^= byte << 8          print("=========================")          for i in range(8):              if (crc & 0x8000):                  crc = (crc << 1) ^ poly              else:                  crc = crc << 1              # 按位与              crc &= 0xFFFF              print(f"{i}:{crc:#04x}")      # return byte      return crc  if \_\_name\_\_ == '\_\_main\_\_':      data = b"Hello, CRC-16!"      print(data)      crc\_value = crc16(data)      print(f"CRC-16 value: {crc\_value:#04x}") |



初步理解是在数据后面加上一些0，这些0的数量与一个特定的多项式的最高次幂相同，然后使用模2除法除以生成多项式，得到CRC校验码；如果余数为0，则说明数据在传输过程中没有发生错误；如果余数不为0，则说明数据在传输过程中可能发生了错误。

那么在实现时将字符串转为bytes，初始化多项式，初始化寄存器，循环遍历每一个字节，将字节左移8位并与寄存器异或，再对8个比特进行处理，如果为1，则左移1位并与多项式poly异或，如果为0，则左移1位。

三、实验中出现的问题：

最大的问题就是要理解crc生成的基本原理和步骤。

四、解决方案（列出遇到的问题和解决办法，列出没有解决的问题）：

翻看书籍，查阅相关资料。

五、实验心得

通过本次实验，我们了解了CRC的实现原理，并且通过实操编写了crc-16的代码实现。