**MPI key-value系统阶段性报告**

1. **程序演示**

见小视频

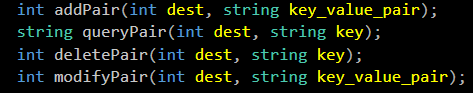
1. **实现细节**
2. **准备工作**

在个人PC上下载了微软的MPI相关软件，用vs 2017配置MPI运行环境，用一个进程模拟一个计算节点。

1. **程序结构**

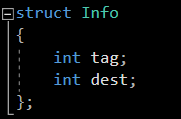
主从结构。以10个程序为例，进程0为主进程，进程1-9为从进程。主进程负责处理与用户的交互与对MPI系统发出增删改查命令(主进程不存储数据)，从进程响应主进程的命令并负责key-value对的增删改查。

程序实现了增删改查的功能，函数定义如下：



1. **主从进程交互过程**

以添加键值对为例，当住进程接收到用户添加键值对的请求时，**首先会通过哈希函数计算key对应的哈希值，用所得到的哈希值对9求余数后再加一，此结果决定了该键值对存储在哪个节点（以10个进程为例）**。决定存储节点后，主进程将会把存储节点的编号以及操作对应的tag(以添加键值对为例，添加键值对的tag为1)广播给所有的从进程，这里我定义了一个Info结构体用于传递对应信息：



从进程接收到主进程的广播，若发现自己是目标存储节点，则会先与主进程交流得知本次所传输数据大小，并在本地申请对应的内存空间，做好接收数据的准备后接收主进程传递过来的数据，并将对数据的操作结果返回给主进程，由主进程显示给用户。在这里，从进程使用的存储结构是C++ STL库中的map，增删很方便，查找的效率也很快（C++的map使用了红黑树的相关知识，查找的时间复杂度为logN）。