Hash功能分析

* 首先创建文本文件（文件相关报错检查？），并在文本文件中打印#!/bin/bash
* 递归遍历目录下所有文件，调用Hash函数计算每个文件的哈希值，如有重复则按照格式打印shell语句到stdout。可以使用stat函数获取文件完整路径，或使用其他方法
* 常见的hash函数，是否考虑采用MD5检验？
* <http://blog.csdn.net/yangyu19910407/article/details/11527153>
* 使用map来存储hashvalue与文件路径的pair，从而达到O(logn)或unordered\_map的O(1)
* 还是不太明白，为什么用map时间复杂度是O(1)
* 需要的功能列表：文件目录读取、hash计算、文件读取写入、文件比较、文件删除（shell语句输出）
* Hash本身是否浪费时间？对于输入数据长度是否为线性关系？
* Hash函数、比较函数、删除函数、递归读取函数、主函数

调用关系一览：（中文）

主函数 - // 循环调用目录递归函数

目录递归函数 - // 调用文件比较函数

// 打开目录 – 需要包含dirent.h头文件以实现目录相关功能

// 遍历每一个文件，计算其哈希值

// 调用比较函数，如果find相同调用脚本生成函数并删除键值，如果不同则insert pair 到myhashmap

// 关闭目录

首先检查目录是否存在

问题：

1. c++读取目录是否有更好的实现方法？目前使用C的dirent

应该将pair存成文件名，文件内容的形式，这样最保险。Unorderedmap是依赖hash value进行存储的，因此如果将hash后的文件的hash\_value存进unorderedmap，则会hash两次以达到最快的访问速度