**一、实验目的：**

1. 了解Linux编程环境；
2. 掌握在Linux环境下编程的常用工具，例如：shell, vim, make, gedit, git及各种语言的集成开发环境。
3. 了解Linux系统的内存、进程、线程、同步、通信的基本原理和其在实际程序实际中的应用。
4. 掌握Linux环境下的应用程序设计、开发和项目管理。
5. 通过16学时的实验，在Linux环境下设计并实现一个代码量不小于2000的项目。

**二、实验安排：**

**第一次实验 ：**确定项目的需求和模块设计，并将项目的需求文档和模块设计方案提交到本Repository。  
**第二到六次实验 ：**细化模块设计方案，完成各模块的接口设计，实现各个模块的功能。  
**第七次实验 ：**整理实验报告。  
**第八次实验 ：**系统功能展示。

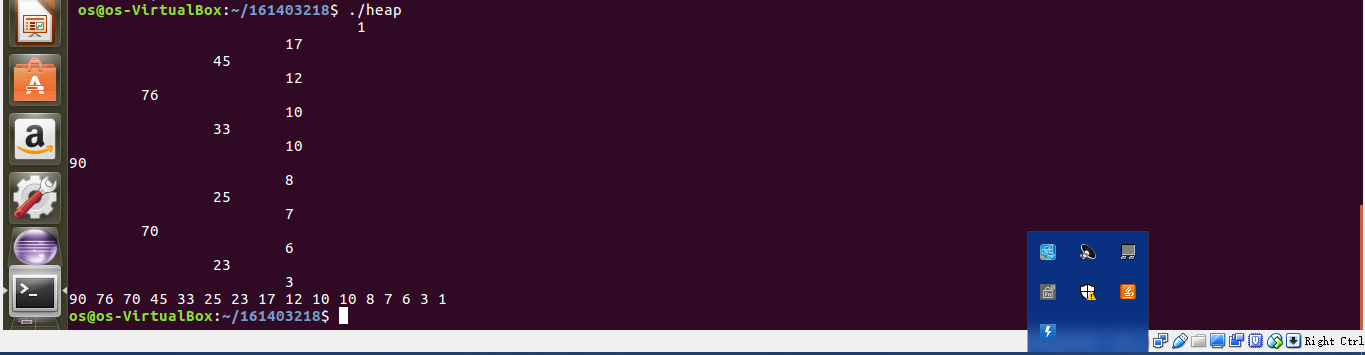
**三、实验内容：**

* **堆**

Heap.cpp

堆的子节点的上层是完全树，而子节点从左向右严格排列，这种数据结构可以用数组模拟。从第一层第一个节点为1开始，从左到右从上到下依次存储。则，对于某个索引为i的节点，其左子节点和右子节点的索引分别为：i \* 2， i \* 2 + 1

**实验结果：**



* **栈**

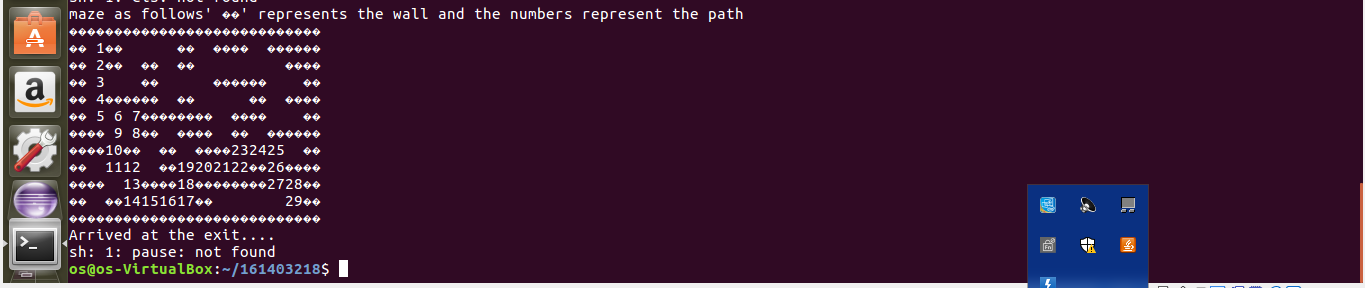
maze.cpp maze.h

以一个m\*n的长方阵表示迷宫，0和1分别表示迷宫中的通路和障碍，求出从入口（1，1）到出口（m,n）的一条通路

（i，j）指示迷宫中的一个坐标， d表示走到下一坐标的方向。 左上角（1，1）为入口，右下角（m,n）为出口。

**实验结果：**



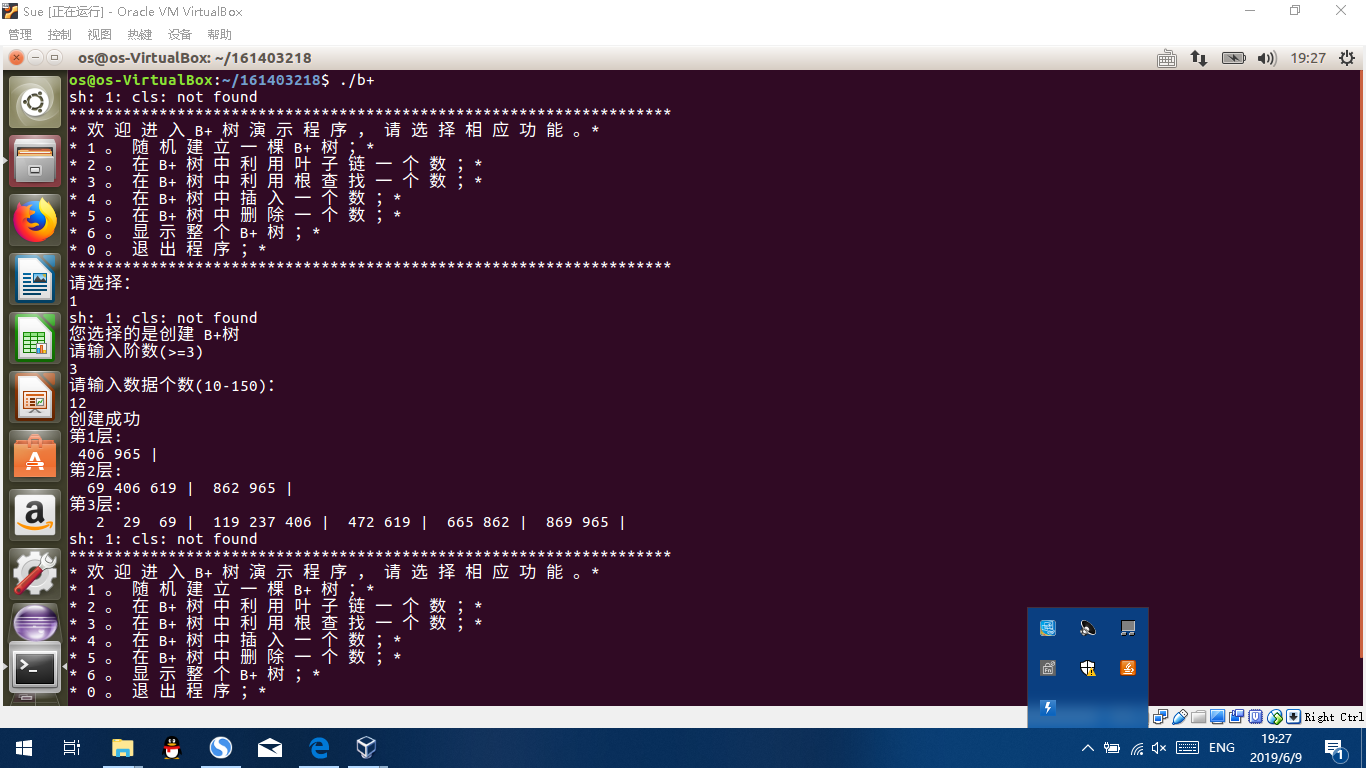


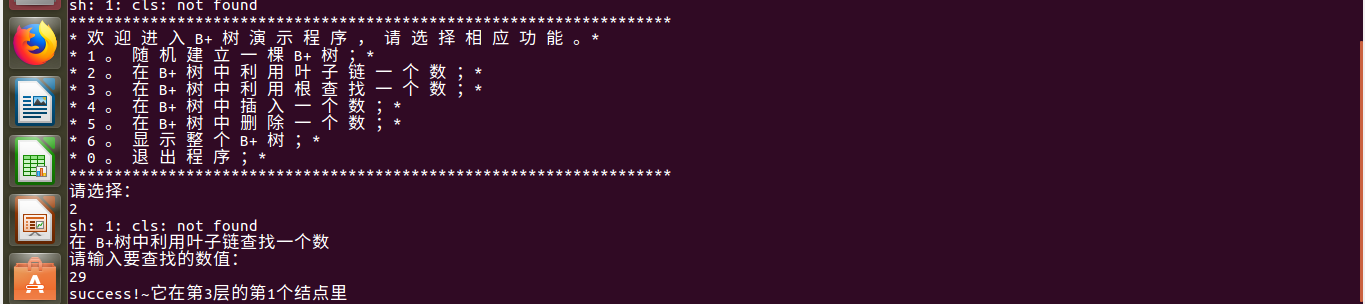
* **B+树**

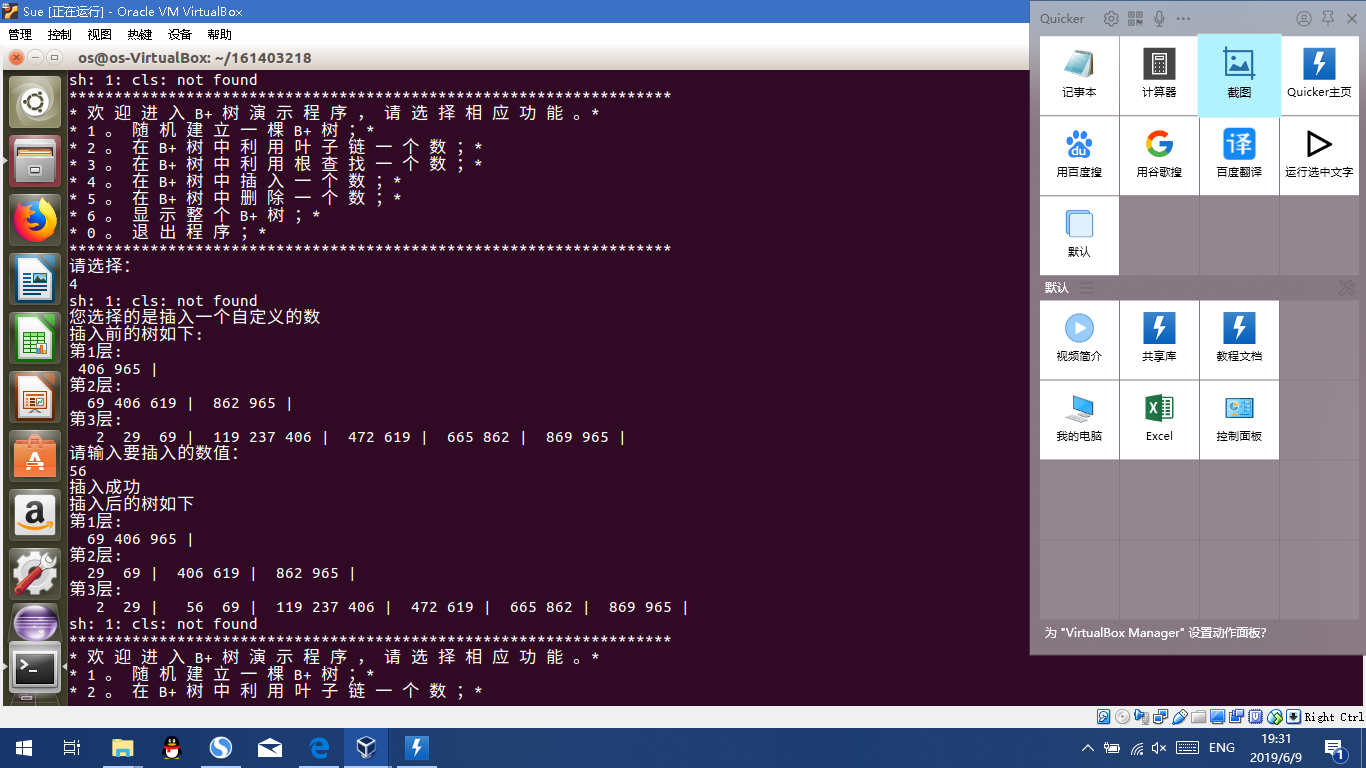
b+.cpp

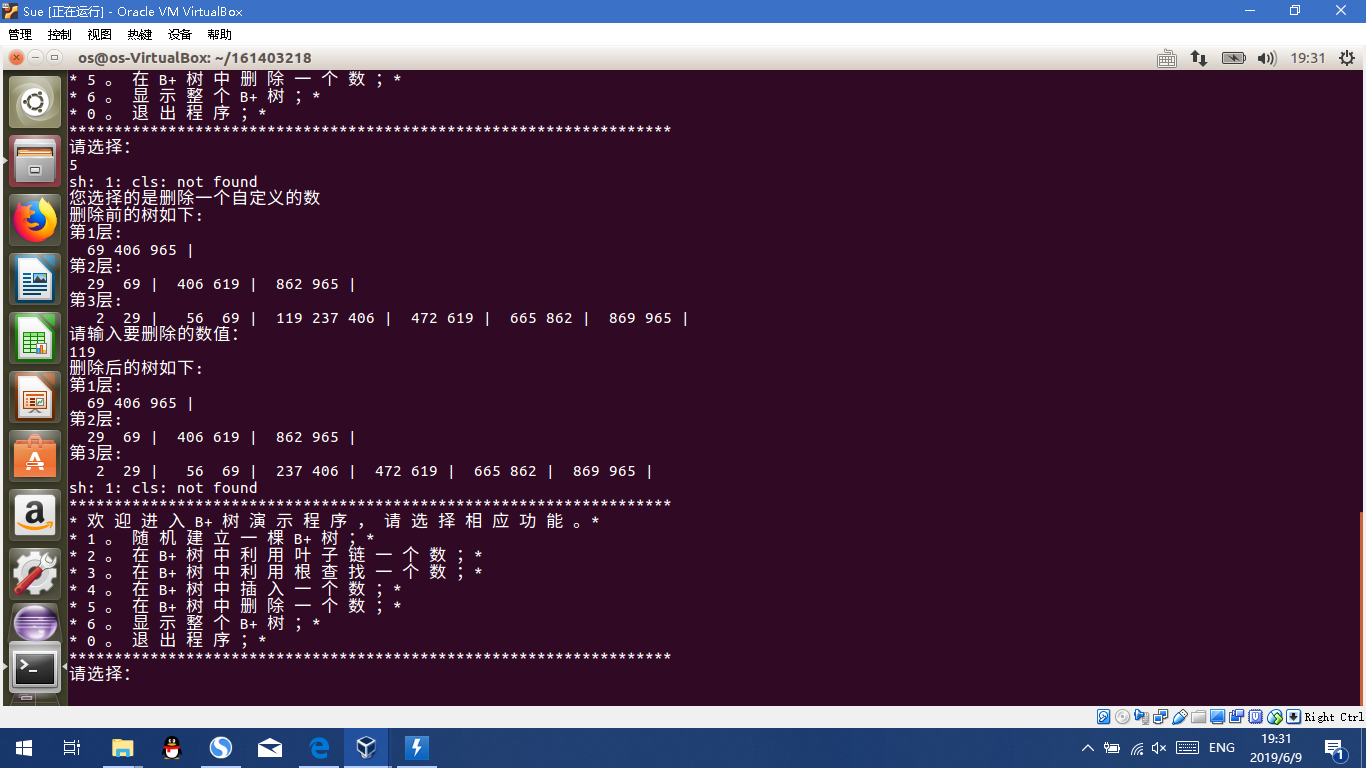
B+树中，所有记录节点都是按键值的大小顺序存放在同一层的叶节点中，各叶节点指针进行连接。

**实验结果：**

****





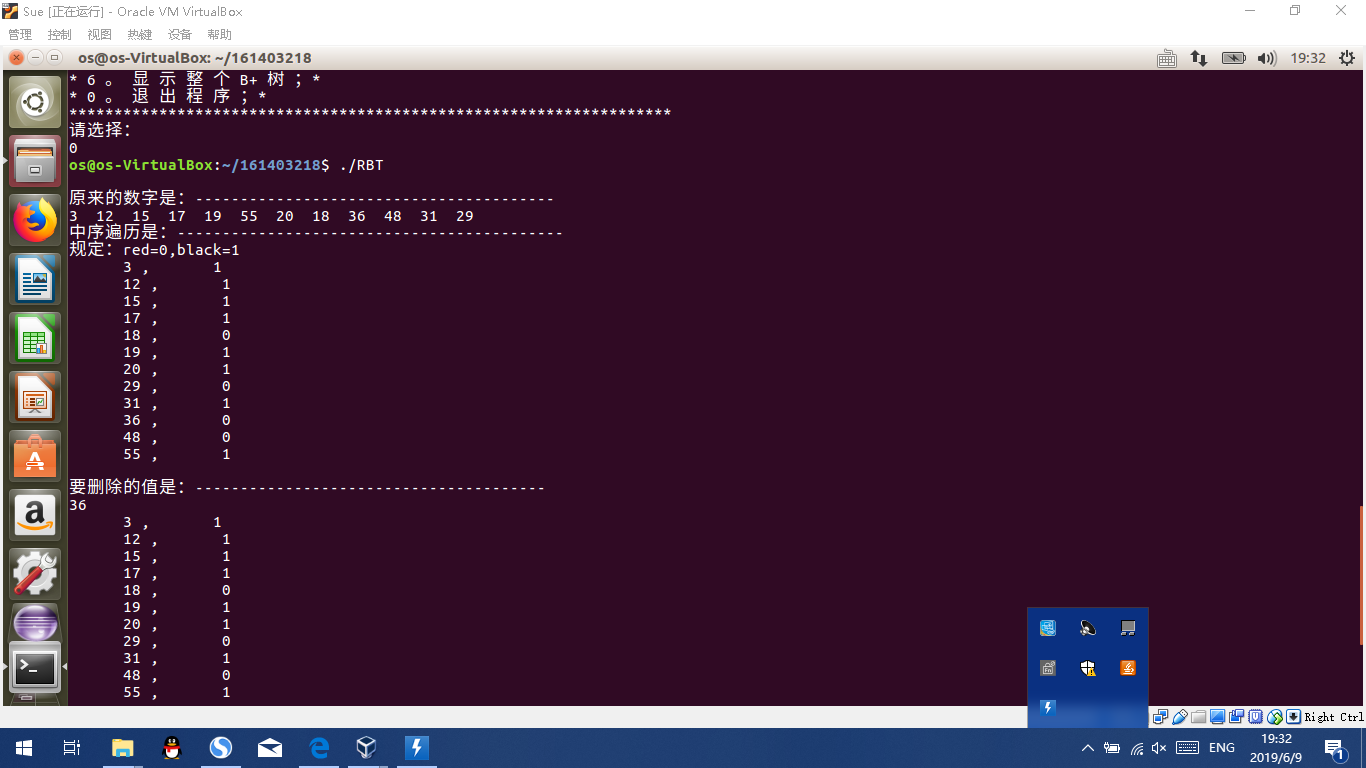


* **红黑树**

RBT.cpp

通过左旋、右旋、重新着色这三种方法，解决二叉树因插入或删除节点产生的不平衡问题。

**实验结果：**

****

****

**四、注意事项：**

Linux和Windows系统下函数需要引用的系统自带头文件不同，如sleep（）函数，在Windows下使用#include<windows.h>，而在gcc编译器中使用#include<unistd.h>，且Linux中没有Sleep（）（单位为毫秒）函数，只有sleep（）（单位为秒）和usleep（）（单位为微秒）函数。

且在Linux下语法要求更为严谨，一些有一点语法错误的程序文件在Windows下可以正确运行，但在gcc编译器中会报错，需要注意，在编写时语法更要严谨。

**五、实验小结：**

通过这次实验，我对Linux的使用更加熟练，对几种数据结构也有了更多的认识和理解，也有了更多兴趣。