

Huffman 文件压缩系统

1. Functions:

1. 对任意文件进行压缩 并且生成压缩文件保存在磁盘上
 1. 读入文件并且保存在一个字符串数组内 或者保存文件指针 或文件流对象
 2. 根据读入的文件 统计出字符出现的频度 频数 保存在一个优先队列内 为下一步编码做准备
 3. 根据字符的频度 建立Huffman树(静态链表结构) 并且保存根节点的标号
 4. 通过Huffman树来建立符号表
 5. 用符号表对原始文件进行编码 并使用位操作生成新的压缩文件并且输出到文件
2. 对经过Huffman压缩的文件进行解压，把解压后的文件保存在磁盘上
 1. 读入压缩后的文件并且保存在一个字符串数组内 或者文件指针 文件流对象
 2. 读入压缩时使用的Huffman树
 3. 依次读取文件的每一个位 并通过Huffman树对文件进行解码 并导出解压后的文件
 4. (选作) 将Huffman 的树型结构展示出来(可视化)

2. Class Definition

```
//Here to Put class HuffmanTree

class HuffmanTree{
private:
    void createHuffmanTree();
    void countFreq();
    void generateCodingTable();
    void pr_encoding();
    void pr_decoding();
    int root;
    vector<StaticHuffmanNode> HuffmanT;
    string codingTable[300];
    string fileStr;
public:
    void encoding();
    void decoding();
    void open();
    HuffmanTree();
    ~HuffmanTree();
    HuffmanTree(const HuffmanTree& T);
};
```

3. Library to Use

1. Qt (For Graphic User Interface)
2. STL (vector priority_queue sort ...)

4. Develop Plan [Hard Deadline Week 19 Date:2014-1-14]

1. Core [Time Planned **5Days**]
 1. Structure Build & Function Prototype(vector and class) [5-6h and 2 days]
 2. make the Functions work [15h and 2days]
 3. Debugging and inmproving [5h 1day]
 2. GUI [Time Planned **5Days**]
 1. Study Qt Lib in 5 Days
 2. Visualizing
-

2014-12-30 By VOID001