JuggFS文件系统

简易文件系统，不考虑什么因素？达到什么目的？（支持简易机制）如何调试？V1.0 迅速学习版本

V2.0 重写版本

1，64位文件系统？那样的话一个地址至少8个字节—一个分区超过2TB，应该还很少见吧，故还是使用32位的文件系统。

2，Block大小是否可变，这个在格式化的时候自适应吧，作为一个const常量。

3，如何测试大容量文件系统（超过2G）的支持？可以申请10G的内存吧，那么寻址很明显使用的是64位的地址。

4，起始固定4KB（为了以后扩容和支持4KB大小的Block考虑），里面包含Block的大小。先使用2KB作为实验。

5，使用位图文件来指示是否使用。（都这样做，对吧）

6，如何串行化？（>>/<<来实现？）必须要填充那些结构，对吧。

信息区（装入“内存”+数据区“直接操作”），故退出时要进行保存or随时保持一致？（暂时确定为退出时保存吧）。

7，使用混合索引分配模式，inode大小也设为128字节吧。（是否固定inode区域的大小？简单来说，这样是可行的，每个块可以有16个文件）--juggNode table有多少个呢？比方说16 \*1024\*10个，，对应2\*1024\*10=20MB，160 K个文件，需要20KB的位图来指示，那么目录节点

8，多级目录的组织：使用孩子兄弟表示法，集成到JuggNode里面去。并且使用固定方式，Node总数量在格式化的时候确定，不可更改。参见superblock，对吧。

增加一倍便可。

9，类ext2表示法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JuggRule | JuggNode 位图 | JuggBlock位图 | JuggNode表（内含关系） | JuggBlock表 |

如何定义JuggRule，都是类，对吧。

10，JuggEngine的接口设计，因为很多函数需要传递很多参数，按正规参数处理吧。

操作目录使用CString来表示，对吧。那么那里面char，需要转换为CString类型。准确的说应该是wchar\_t吧统一使用UTF-16，仅适用于Windows平台。

11,JuggEngine中绝对不使用进在windows下定义的内容，为了方便移植，那么windows客户端再使用一个引导端。该引导先散列组织，最后应该集成到可视化界面中去。

12，大文件的操作，其实也是，好像分类关系很难了，因为集成到那里面去了。特别是文件操作中。

13，目录的遍历，先跟序，好像比较简单，只要将有双孩子的节点入栈就可以了。

block的计数是int，那么支持最大的分区为2K\*2G=4T.

512\*512\*512\*2K=2^27 \*2K=2^8 G 达256G最大文件 可以了。

512\*512 \*2K=2^18 \*2K=512M

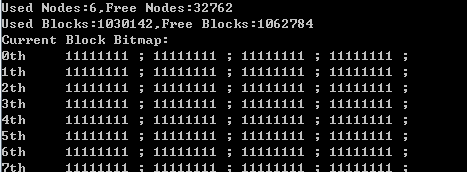
512

14，过多使用Flush会导致速度严重变慢，至少保证一个大函数里面只有一处flush。特别是复制大文件的时候。 删除文件，另外一个重点。

a,释放node对应的bitmap

b,释放block对应的bitmap（空文件夹没有block）

c,修改父节点（孩子节点）的指针指向。(修改文件夹时该项就不需调用了)



pic.jpg:

2410+1+5=2416(blocks)

末尾block内容是635B

最后索引block的数量是362个。

羽毛球赛.rar:

9399+1+19=9419;

错误检测，暂时不支持

15+4+4=23\*4=92+1=93个字节

256-93=163 最多支持80个字节的字符了