# 磁盘格式说明文档-2021-5-1

## 1.整体布局

文件系统再磁盘上布局：

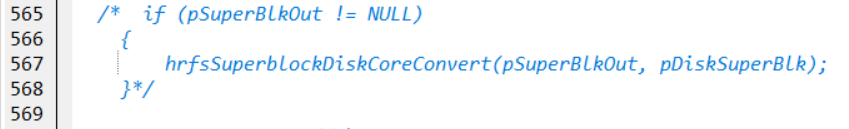
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PBR** | **FSB0** | **FIB0** | **FSB1** | **FIB1** | **INODE\_SET** | **INODE\_LOG** | **FSB\_META**  **BITMAP0** | **FSB\_META**  **BITMAP1** | **FIB\_META**  **BITMAP0** | **FIB\_META**  **BITMAP1** | **GLOBAL**  **\_INFO0** | **GLOBAL**  **\_INFO1** | **DTAT**  **SPACE** | **SUPER**  **BLOCK** |

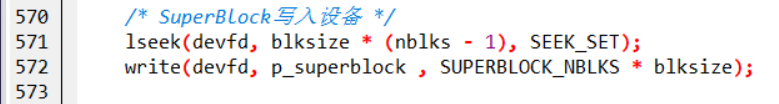
说明：

1.此版本一个块的大小为4096字节，也就是4KB。

2.此版本一个32768块，所以整个文件系统一共128MB。



 3.此版本对于SUPERBLOCK不考虑内存和磁盘的问题，直接写入incore版本。



4.此版本采用的是从PBR到SUPERBLOCK的依次写入的方式。

5.此版本的部分区域申请缓冲过大，需要修改成循环写。

6.此版本未实现元数据占用inode。//暂时不考虑

7.此版本的默认设备为“D:\\个人事务\\操作系统项目\\操作系统大赛\\2021.4.2\\test.txt“，如需改变需要手动修改源码。



8.此版本的有些结构体声明多余，需要精简。

9.此版本声明了默认使用小端字节序。实际上比较混乱，需要重新考虑。//暂时不考虑

## 2.PBR

PBR中存放了x86的跳转指令（3个字节）、系统的ID号（8个字节）、PBR中块的数量、每个块的字节数，对于PBR，考虑了大小端的问题。

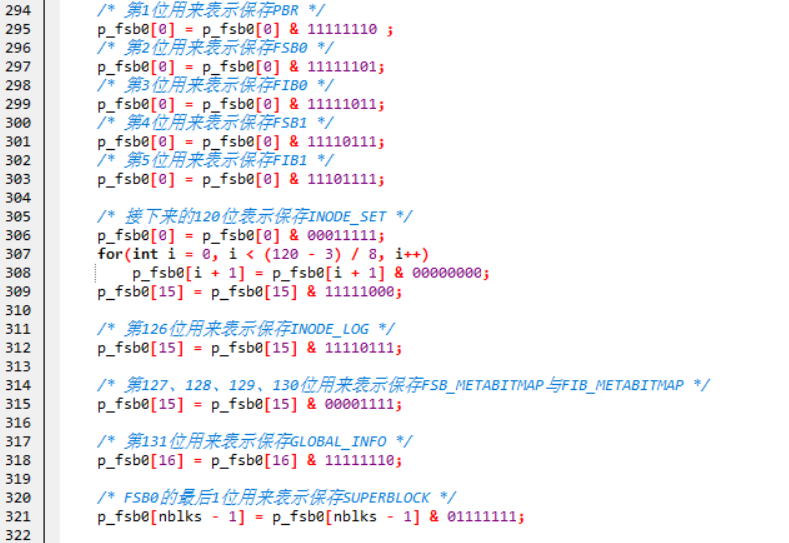
PBR单独占用了1个块。

## 3.FSB

FSB中存放块的使用情况，因为一个块4096字节，4096\*8=32768，所以占用1个块刚好可以表示整个文件系统的所有块。

默认1为空闲，0为使用，其中INODE\_SET占了120位，其余个区域都只占1位

FSB要写两块，但是并不连续。



## 4.FIB

FIB中存放的是inode的使用情况，因为此版本一共有4096个inode，因此FIB占1个块即可满足要求。

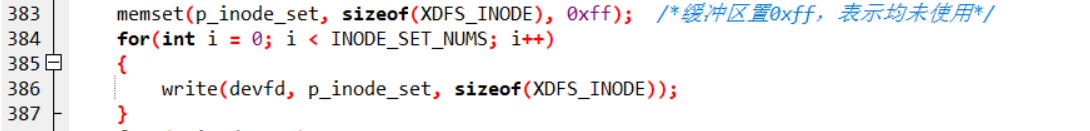
默认1为空闲，0为使用，最初所有的inode均未使用，所以全部初始化为1，此版本未实现元数据占用inode。



## 5.INODE\_SET

INODE\_SET中存放的是inode数据，因为XDFS\_INODE的大小为120字节，所以INODE\_SET的占用1个块，其块数算法暂定为：XDFS\_INODE所占字节\*4096/4096字节每块= INODE\_SET所占字节，此时未考虑对齐问题。

INODE\_SET所有位均初始化为1，INODE\_SET采用了循环写入的方法。



## 6.INODE\_LOG

INODE\_LOG的大小恰好为512字节，因此占1个块即可，同时INODE\_LOG所有位均初始化为1



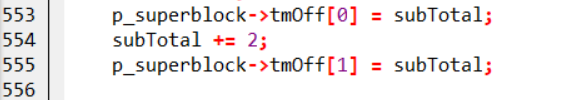
## 7.FSB\_METABITMAP与FIB\_METABITMAP

FSB\_METABITMAP与FIB\_METABITMAP各要写两个，其初始化时，所有的位均为0，表示对应的bitmap块全部已被占用。

FSB\_METABITMAP与FIB\_METABITMAP每一个都占用1块。







## 8. GLOBAL\_INFO

GLOBAL\_INFO中记录了事务的ID号、时间戳、空闲索引节点、空闲数据块、待删除的节点等。

GLOBAL\_INFO有两个，每一个都占用1个块，初始化时每一位都设置为0。

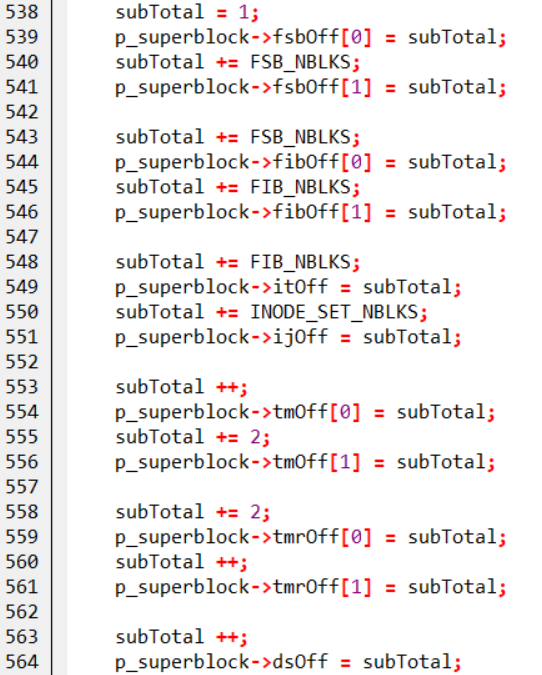


## 9.DATASPACE

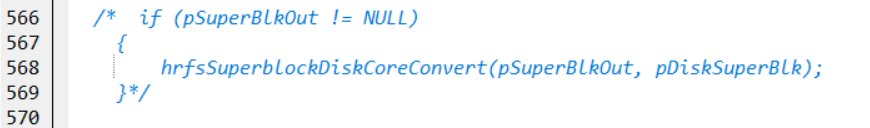
此区域是数据区域，不用做任何处理。

## 10.SUPERBLOCK

SUPERBLOCK中储存着关于文件系统的重要参数，例如：SUPERBLOCK的创建时间、文件系统的版本号、文件系统的ID号、磁盘块的大小和数量、保留的磁盘块的空间大小、inode的数量、存放数据的空间的大小、文件系统在磁盘上磁盘块组的大小和数量、空闲空间位图偏移量、空闲索引节点位图偏移量、INODE\_SET偏移量、INODE\_LOG偏移量、METABITMAP的偏移量、GLOBAL\_INFO的偏移量、DATAPACE的偏移量等。



对于SUPERBLOCK，没有进行disk和incore之间的转化，直接将incore的数据写入。



SUPERBLOCK的大小为96字节，其占用1个块。

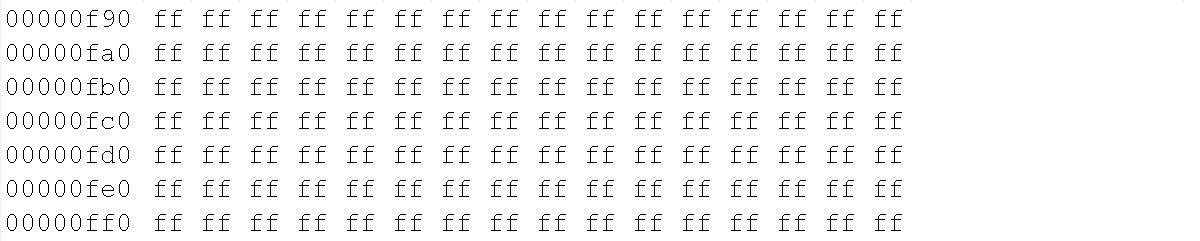
## 11.写入后正确性检查

### 11.1PBR正确性检查

0x0字节写入0xe9，0x3字节到0x6字节写入“xdfs“字符串，0xb字节到0xc字节写入0x0010是blksize的小端字节序，0x20字节到0x23字节写入0x00800000是nblks。

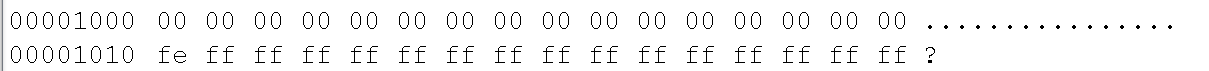


其余各位均填充0xff知道第4096个字节，也就是0xfff字节（因为从0x0开始计数）。



### 11.2FSB正确性检查

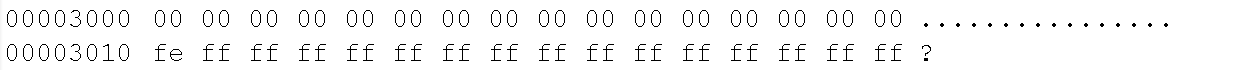
FSB的前15个字节均为0，用来保存PBR、FSB0和FIB0、FSB1和FIB1、INODE\_SET、INODE\_LOG、FSB＿METABITMAP和FIB\_METABITMAP，第16个字节中的第一位用来表示GLOBAL\_INFO。



FSB中的最后一块用来表示SUPERBLOCK，其余区域填充0xff。



FSB有两块，上两幅图位FSB0；下两幅图位FSB1，两者的排布一致。





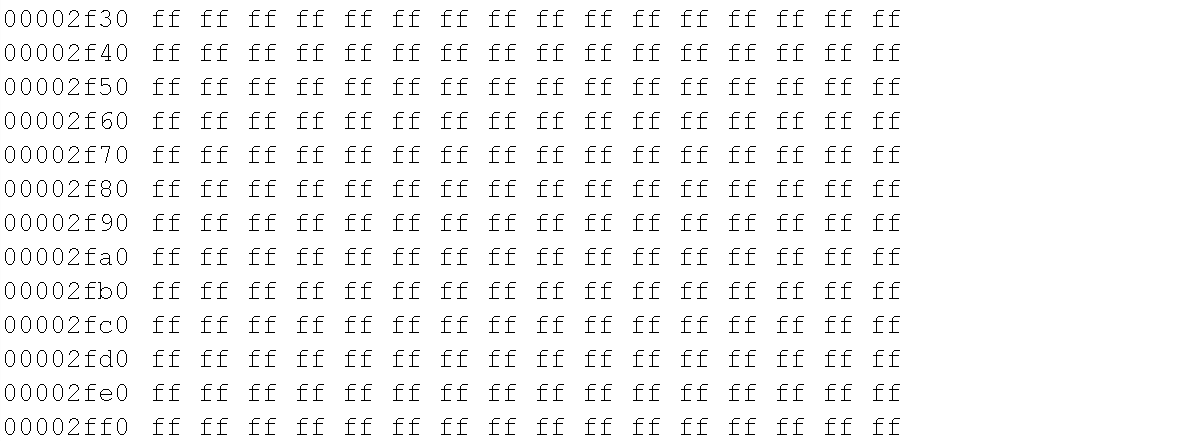
需要注意的是FSB0是从0x1000到0x1fff，FSB1是从0x3000到0x3fff。

### 11.3FIB正确性检查

FIB所有的位都初始化为0。

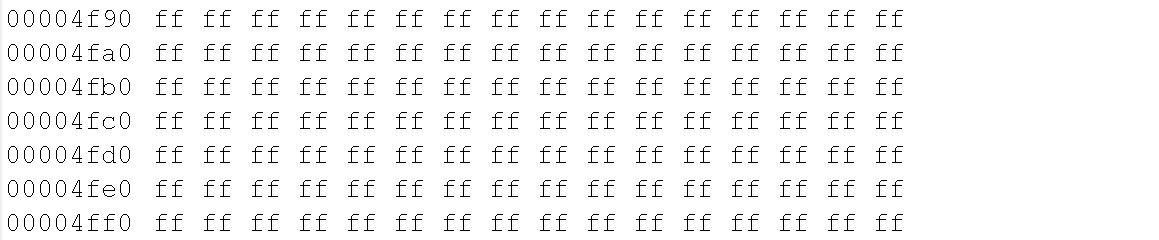
FIB0：





FIB1：





### 11.4INODE\_SET正确性检查

开头为0x5000，跳过PBR、FSB0和FIB0以及FSB1和FIB1。

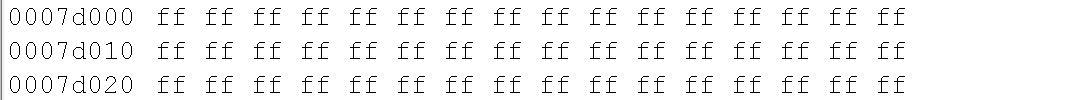


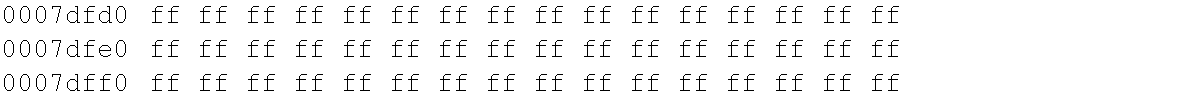
末尾为0x7cfff，因为125\*4096=0x7d000，因为从0开始计数，所以第125块的最后一个字节为0x7cfff。



### 11.5INODE\_LOG正确性检查

开头为0x7d000，末尾为0x7dfff。

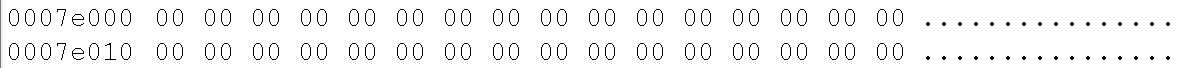


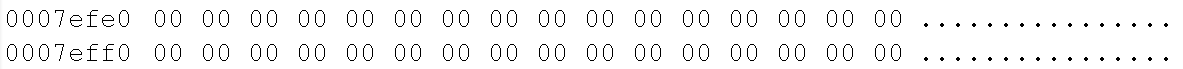


### 11.6FSB\_METABITMAP和FIB\_METABITMAP正确性检查

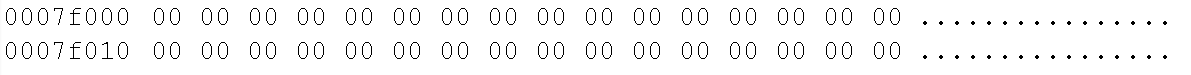
FSB\_METABITMAP和FIB\_METABITMAP每一个都有两块，所以一共有四块均初始化为0x00。

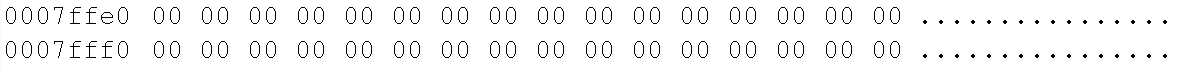
FSB\_METABITMAP0：



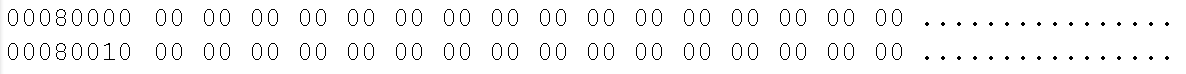


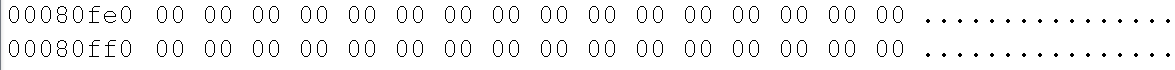
FSB\_METABITMAP1：



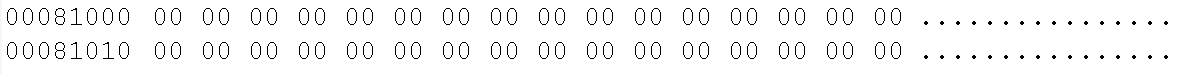


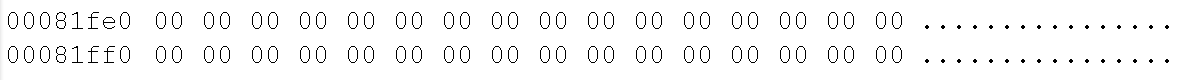
FIB\_METABITMAP0：





FSB\_METABITMAP1：

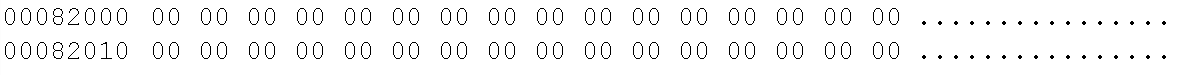


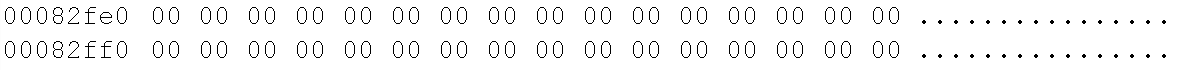


### 11.7GLOBAL\_INFO正确性检查

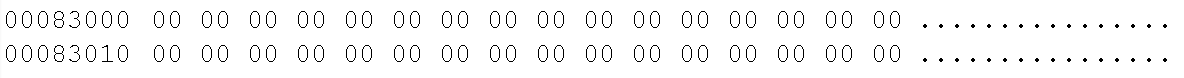
GLOBAL\_INFO有两块，每一块的每一字节都全置为0x00。

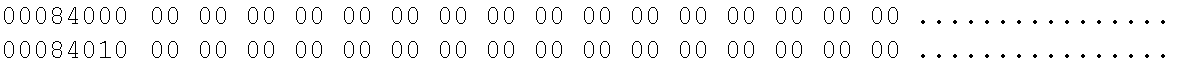
GLOBAL\_INFO0：





GLOBAL\_INFO1：



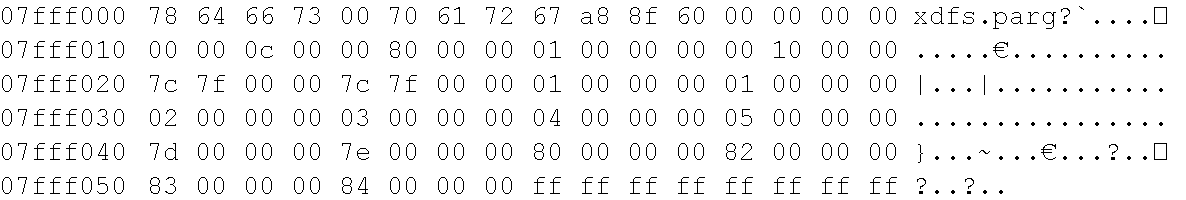


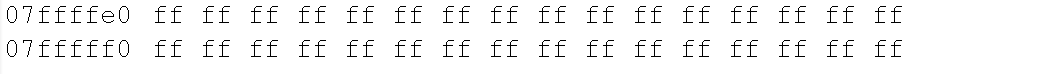
### 11.8DATASPACE正确性检查

未进行任何操作。

### 11.9SUPERBLOCK正确性检查

SUPERBLOCK写到了最后一块，最前面储存数据，其余区域设置为0xff。





### 11.10总结

经检查，磁盘布局写入无误，仍需要考虑大小端的问题，其次，申请缓冲区较大的问题也需要考虑，最大申请了4 KB。