Project 5　File Defragmentation

README

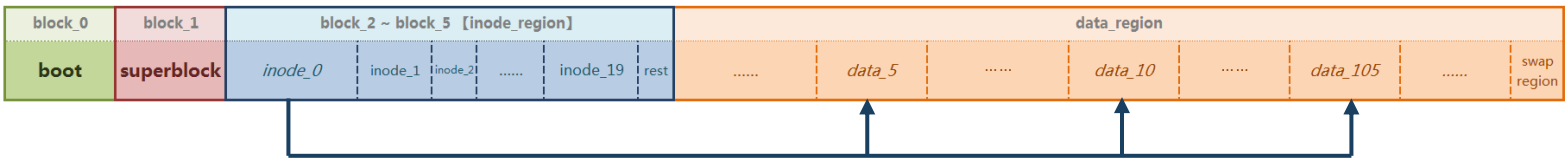
10152130137 汪贻俊
10152130220 邢思远
10152130145 许　倩

1. Introduction

这是一个文件系统的碎片化整理程序。本次作业首先要理解 Linux 下的文件系统的数据结构和文件系统的组成方式，即文件系统由多个区域构成，比如 superblock 块、inode 区、数据区等。其次需要理解文件系统碎片化的原因，然后再去实现文件系统的反碎片化程序来整理文件系统的碎片。

2. Implement

为了描述方便，下面以本次作业所给的示例文件映像为例来介绍实现的方法：



本次作业所给的示例文件映像（图）

Step 1

读取 superblock 块，获取文件系统块的大小，inode region 和 data region 的偏移量，这样就可以确定 inode region 和 data region 的具体位置。与此同时也要将 superblock 写入新的文件系统，尽管此时写入的 superblock 仍需再修改，但此时写入是为了安排好这些区域在新建文件系统中的分布。

Step 2

确定 inode region 的位置后，也可以计算出 inode region 的空间，然后就是读取所有的 inode 然后将这些 inode 同时也写进新的文件系统，这些 inode 之后也是要重新修改的，但同样也是为了安排好新文件系统的分布。**这里要注意可能不是所有的空间都被用于构造 inode，最后的剩余空间也要填充进新的文件系统。**

Step 3

写入了 superblock 和 inode region 后，就可以向文件系统写入数据块了，因为文件系统的所有数据块都是靠 inode 节点维护的，因此就依次读取源文件系统的各个 inode 节点来获取文件系统的所有数据块，读取的顺序就是

直接索引块 => 一级间接索引块 => 二级间接索引块 => 三级间接索引块

此处的处理方式是碎片整理程序的关键:

[1] 直接索引块就是直接读取然后写入；

[2] 一级索引块是先确定一级索引块管理的数据块个数然后计算好一级索引块中对应的索引块编号，构件好一级索引块后将其写入新文件系统，同时缓存对于 inode 节点的修改，然后读取，最终写入数据块；

[3] 处理二级索引块就要先计算好对应的一级索引块对的编号，构建好二级索引块后写入新文件系统，然后就采用 2 步骤中的方式同样处理一级索引块；

[4] 处理三级索引块的方式也是先构建好三级索引块然后写入文件，之后采用 3 步骤处二级索引块。

Step 4

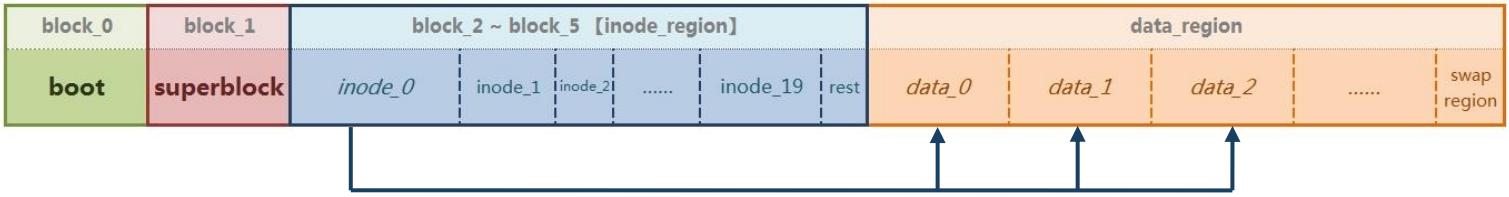
完成数据块的读取和写入后，文件系统还有一部分空闲的数据块，这些数据块也要填充进新的文件系统，同时要注意这些空闲块的首部的 4 个字节要指向下一个空闲数据块。

Step 5

将之前缓存的对 inode 节点的修改写入 inode 节点，同时将空闲的 inode 节点的 nlink 置为 0，同时 next_inode 指向下一个空闲 inode，此处的 inode region 也进行了整理，将所有的空闲 inode 节点全部放到 inode region 的尾部。

Step 6

最后就是修改 superblock 中的 free_inode 和 free_iblock 即可。



整理完成后的文件系统映像（图）