计算机系统工程Project 1项目文档

17302010002 黄元敏

一、简介

本Project主要是需要实现`Inode`、`File`、`Directory`层的一些接口函数，以实现一个文件系统对文件、目录的读取功能。

二、实现细节

inode.c

本文件中主要实现了`inode_iget`和`inode_indexlookup`两个函数。

其中，`inode_iget`通过给定的`inode`编号从文件系统中获取对应的`inode`内容。主要通过计算出一个`block`中能够存放的`inode`结构数量，来计算给定的`inode`编号应当处于连续磁盘的具体哪个`block`上，以及在该`block`上的偏移量。而后，只需要将这个`block`通过disk相关的函数读取进来，返回对应偏移位置上的`inode`即可。

至于`inode_indexlookup`则是负责通过给定的文件在`inode`上的`block`编号，获取具体文件在磁盘上的`block`编号。实现该函数的主要思路为：对文件类型进行判断，对于小文件，直接解读`inode`结构中的`i_addr`字段，获取对应`block`编号上的值即可；而对于大文件，则需要判断其是否在单次间接`block`中，还是要去二次间接`block`上获取。无论何种方式，均是通过计算偏移量来获得`block`编号所在的磁盘`block`，而后在该`block`上获取需要的`i_addr`中的具体值。

file.c

本文件中的`file_getblock`函数目标为获取给定文件`block`的内容。具体实现上，通过给定的`inode`编号调用`inode_iget`获取对应`inode`，而后调用`inode_indexlookup`获取文件所在的`block`编号，之后就能直接从disk上读取到对应`block`的内容。由于函数还需要返回读取到文件的实际大小，因此，我们可以通过`inode`中存储的文件大小信息，来判断该文件需要多少个`block`来存储；再根据我们此次查询的`block`编号来判断是否为该文件的最后一个`block`。除文件的最后一个`block`外，其余`block`的大小应当占满整个`block`，即大小为`DISKIMG_SECTOR_SIZE`。

directory.c

本文件中的`directory_findname`函数是用于查询给定目录中给定名称的文件内容并获取。在实现上，只需要读取给定的`inode`，通过该`inode`的大小来判断该`inode`中的存储占据了多少个磁盘`block`。对这些数量的`block`进行循环读取，再对读取到的`block`内容按照`direntv6`结构进行解析，查询其`d_name`是否与给定的`name`相同，即可找到该目录下名称为`name`的文件。

pathname.c

本文件中的`pathname_lookup`函数通过给定的绝对路径获取对应的`inode`编号。实现该函数的主要思路为：首先对给定的路径是否为根目录进行判断，如果是，则直接返回`ROOT_INUMBER`；否则对路径进行递归解析，不断切割出头上的父目录、获取对应`inode`、在对应`inode`上解析子路径，直至子路径不再包含分隔符，即到达目录最深处时，返回所在`entry`的`inode`编号。

三、测试

由于没有提供`sanitycheck`工具，因此我通过命令行中执行给定的测试命令，并将输出与给定的gold结果进行比对来测试程序的正确性。

最终，在三个Disk Image上执行i、p两项功能均与给定的标准输出相同，通过了测试。