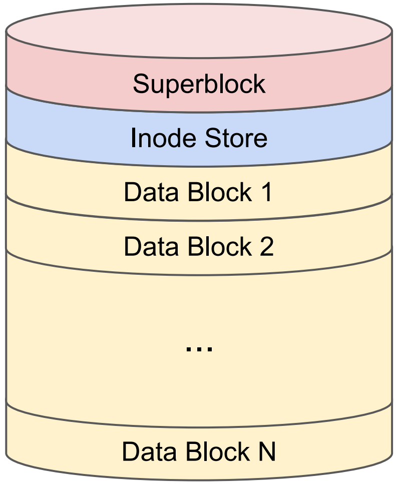
PantryFS Specification

Mitchell Gouzenko

PantryFS Layout

- Each slice shown to the right is one block.
- Each block is 4096 bytes.

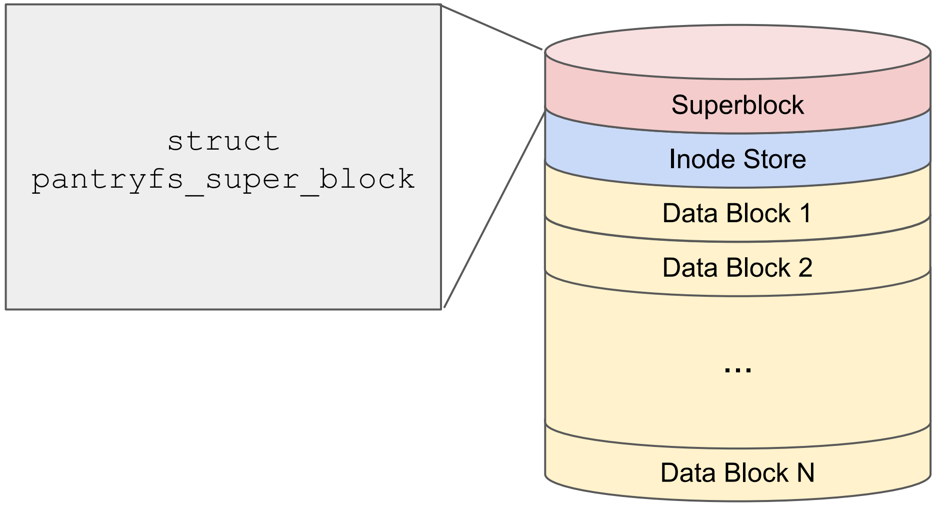


The Superblock

- The `pantryfs_super_block` is padded to be 4096 bytes, so it fits directly in the first block.
- Contains bit vectors representing which inodes and data blocks are free.
- Bit vectors declared by macros:

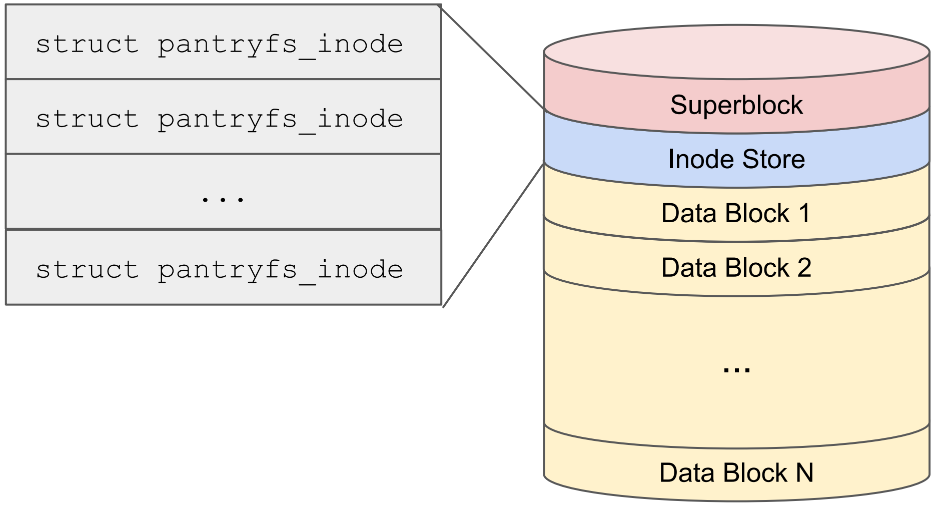
```
DECLARE_BIT_VECTOR(free_inodes, PFS_MAX_INODES);

DECLARE_BIT_VECTOR(free_data_blocks, PFS_MAX_INODES);
```

- We indicate that an inode or data block is **occupied** by setting the corresponding location in the bit vector to 1



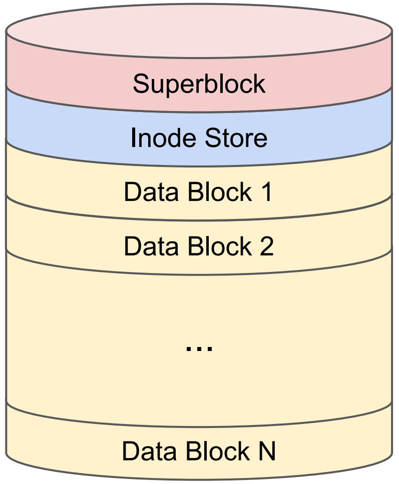
The Inode Store

- The inode store is filled with contiguous `pantryfs_inodes`. The maximum possible number of inodes is crammed into the inode store:
```
#define PFS_MAX_INODES (PFS_BLOCK_SIZE /
sizeof(struct pantryfs_inode))
```
- `pantryfs_inode` stores metadata about a file or directory (size, access times, mode, etc).
- Each `pantryfs_inode` is associated with a single data block via `data_block_number`. The data block, ***not the inode*** stores file or directory contents.
- Each inode is indexed by its position. The 1st inode is inode 1; the 10th inode is inode 10. Note that inodes are indexed from 1, not 0. The reason is that functions in VFS that return inodes return an unsigned int. On error, they return 0.



The Data Blocks

- Case 1: Inode corresponds to a directory
- Case 2: Inode corresponds to regular file



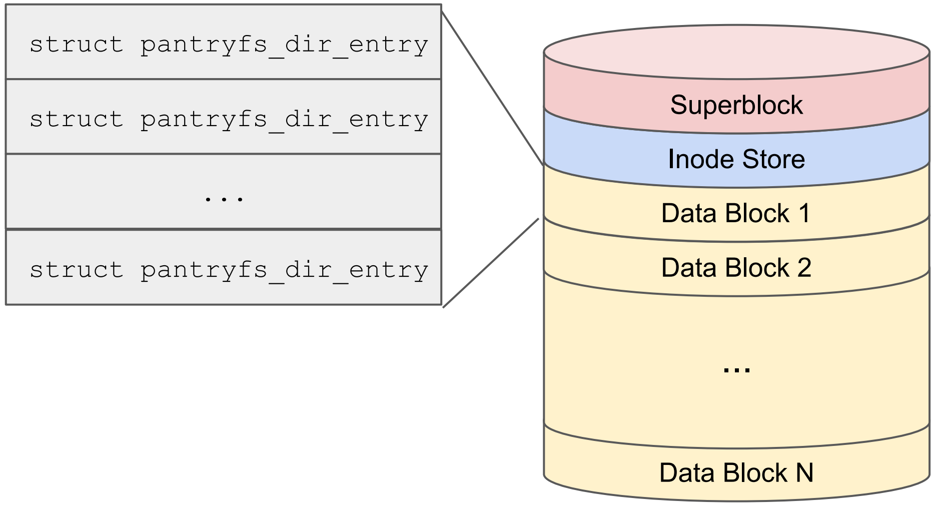
The Data Blocks: Case 1 (directory)

- Data block is a series of contiguous `pantryfs_dir_entry`. The maximum number of entries that can be crammed into the data block is given by:

```
#define PFS_MAX_CHILDREN (PFS_BLOCK_SIZE /
sizeof(struct pantryfs_dir_entry))
```

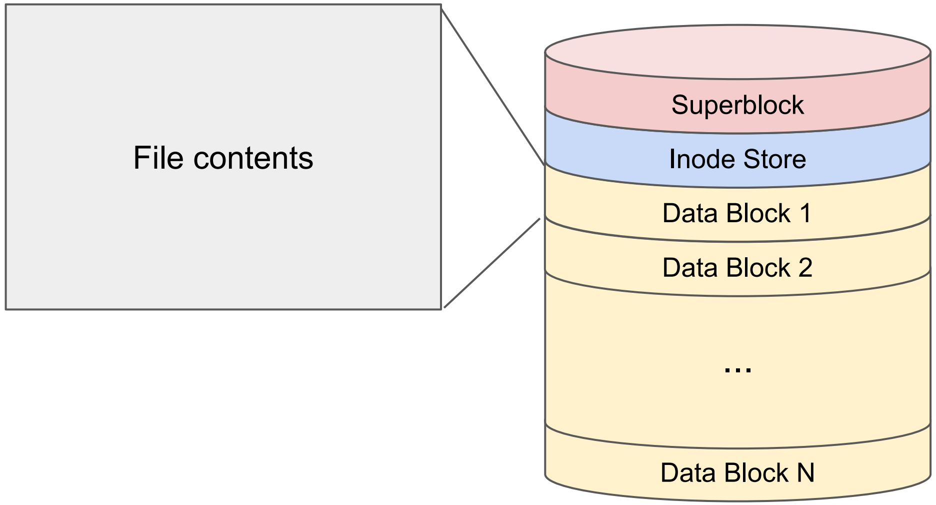
- `pantryfs_dir_entry` maps filename to inode number.

- Each `pantryfs_dir_entry` has a flag indicating whether it's active or not. When we unlink a file (rm command), we lazily delete the file by setting flag to 0.


struct pantryfs_dir_entry
struct pantryfs_dir_entry
...
struct pantryfs_dir_entry
Superblock
Inode Store
Data Block 1
Data Block 2
...
Data Block N

The Data Blocks: Case 2 (regular file)

- Data block is a series of bytes representing file contents.
- Since each inode can have only one data block, files must be smaller than 4096 bytes.



PantryFS Limitations

- Each inode can have only one data block. Thus, a PantryFS file (or directory) cannot be more than 4096 bytes in size!
- For this reason, each directory can only have `PFS_MAX_CHILDREN` children. Recall that:

```
#define PFS_MAX_CHILDREN (PFS_BLOCK_SIZE / sizeof(struct pantryfs_dir_entry))
```

- The number of inodes is limited by the macro `PFS_MAX_INODES.` Therefore, at any given time:

# files + # directories ≤ `PFS_MAX_INODES`

recall that `PFS_MAX_INODES` is defined as:

```
#define PFS_MAX_INODES (PFS_BLOCK_SIZE / sizeof(struct pantryfs_inode))
```