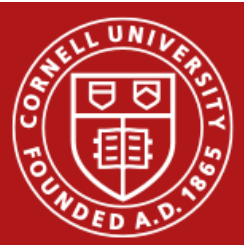
Project 6
Unix File System

Ege Mihmanli

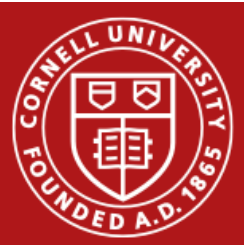
Department of Computer Science
Cornell University

November 18, 2016



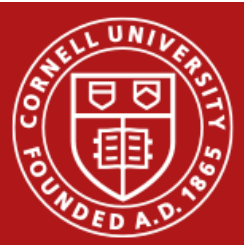
We've come a long way

- It's not me, it's you PortOS!
- Project 6 will be released soon.
- It is due 12/02, 11:59pm.
- Reminder: Project 5 is due this coming Monday.



What you will do

- Implement a Unix-like file system: ufsdisk
- Use bitmaps to keep track of free space
- Keep your file system consistent at all times
- (what does that mean?)
- Optional: implement a file system checker: fsck

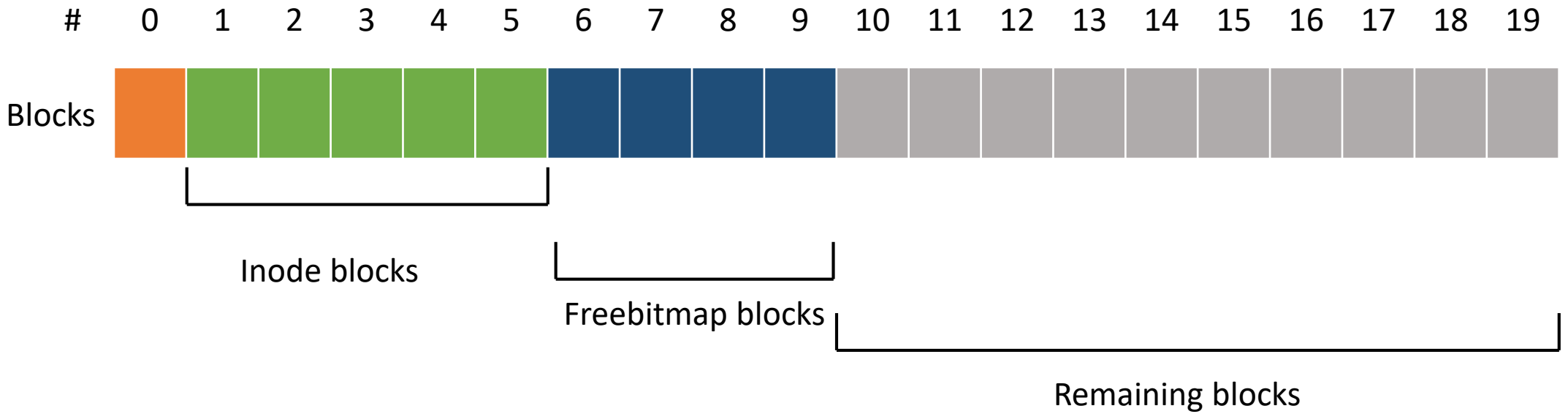


Block Store Abstraction

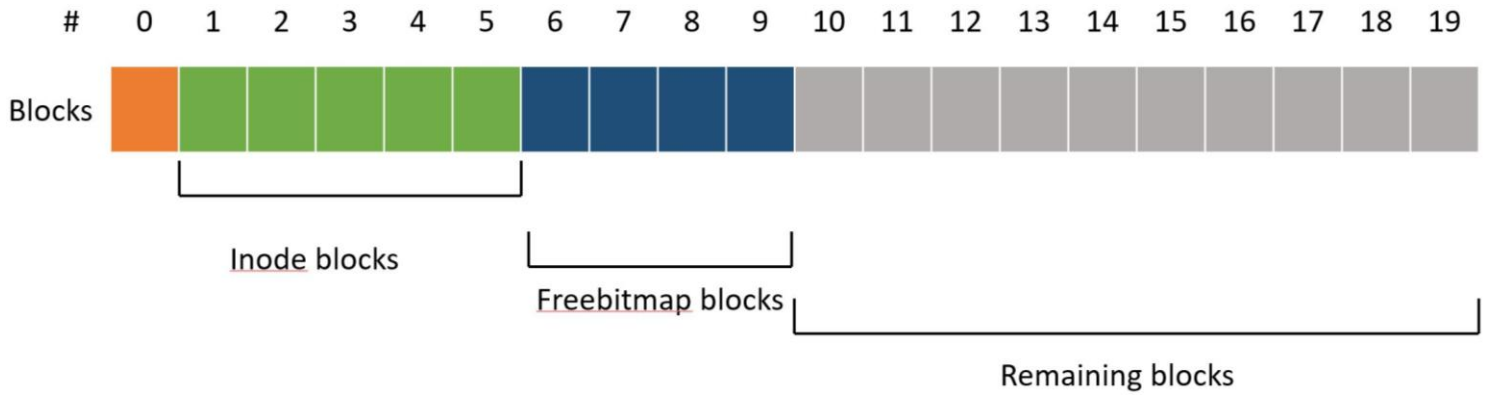
- Taken directly from block_if.h (!!!!!!READ IT!!!!!!!!)

```
struct block_if {
        void *state;
        int (*nblocks)(struct block_if *bi);
        int (*read)(struct block_if *bi, block_no offset, block_t *block);
        int (*write)(struct block_if *bi, block_no offset, block_t *block);
        int (*setsize)(struct block_if *bi, block_no size);
        void (*destroy)(struct block_if *bi);
};
```

What it looks like

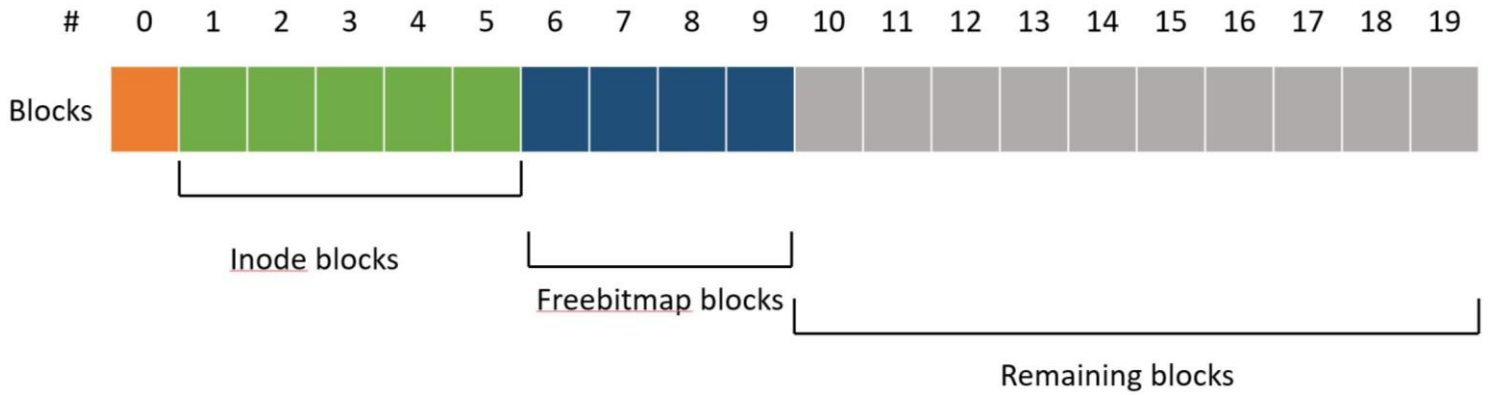


Orange Is The New Superblock




```
Struct ufs_superblock {
        unsigned int magic_number; // magic number of ufsdisk
        block_no n_inodeblocks; // number of inodeblocks in this system
        block_no n_freebitmapblocks; // number of freebitmap blocks
};
```

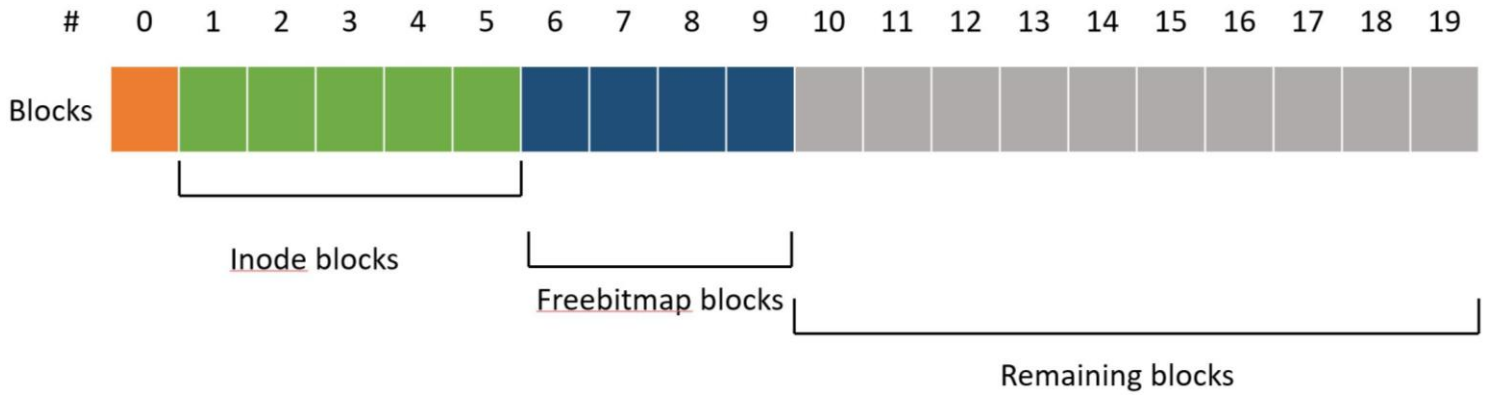
Followed by inode blocks




```
#DEFINE INODES_PER_BLOCK (BLOCK_SIZE / (sizeof(struct ufs_inode)))

Struct ufs_inodeblock {
        struct ufs_inode inodes[INODES_PER_BLOCK];
};
```

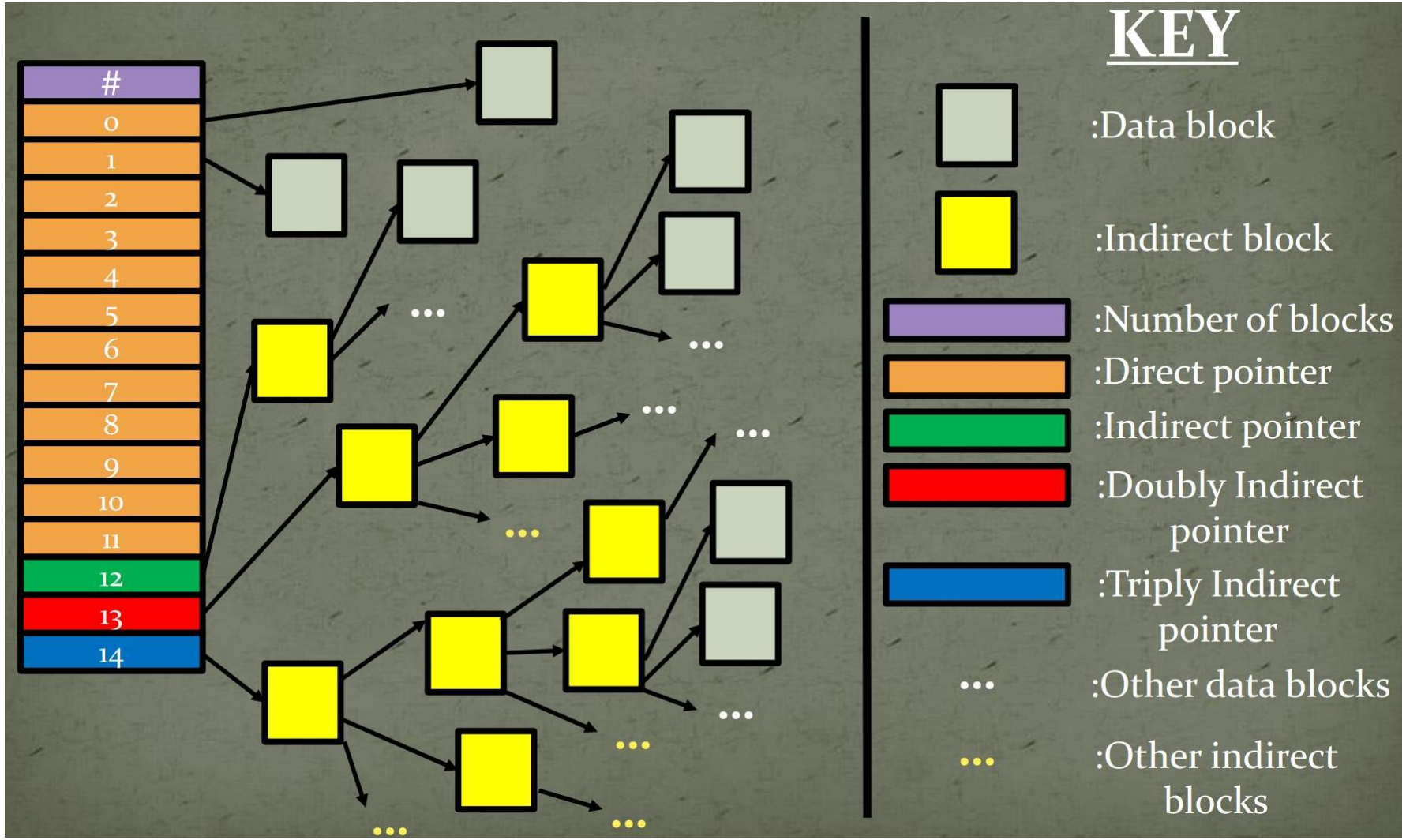
Each inodeblock stores many inodes



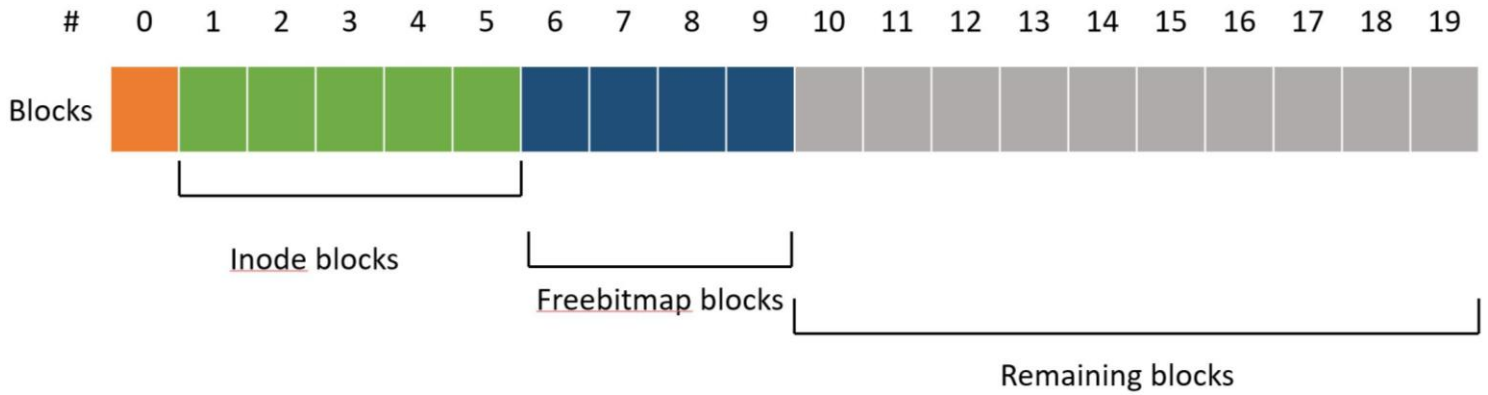

```
#define REFS_PER_INODE 15

struct ufs_inode {
        block_no nblocks; // total size of the file block_no
        refs[REFS_PER_INODE];
};
```

Inside a single inode



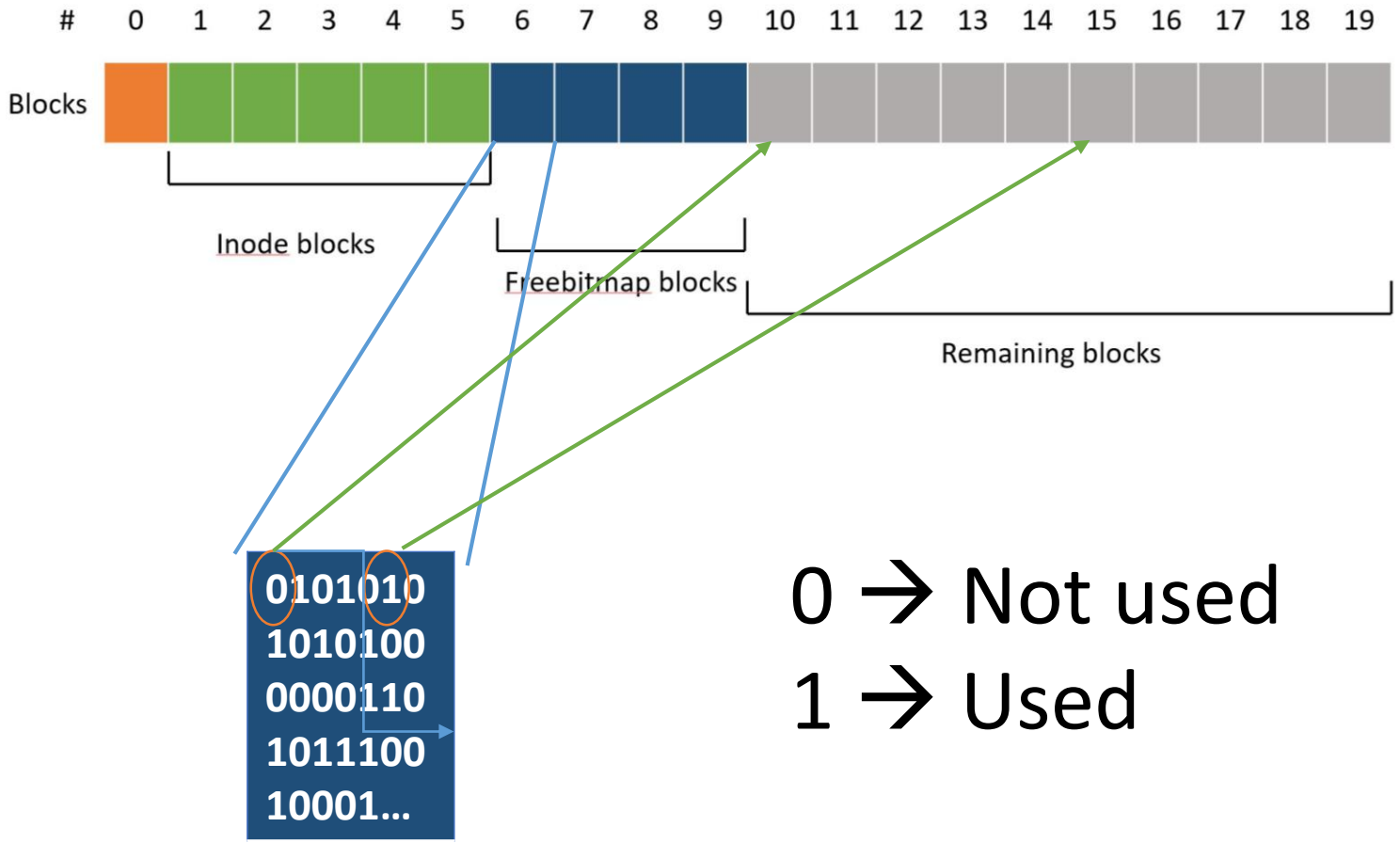
Onto freebitmaps



Each freebitmap block stores enough bits to represent all the remaining block.

```
Struct ufs_freebitmapblock {
        char status[BLOCK_SIZE];
};
```

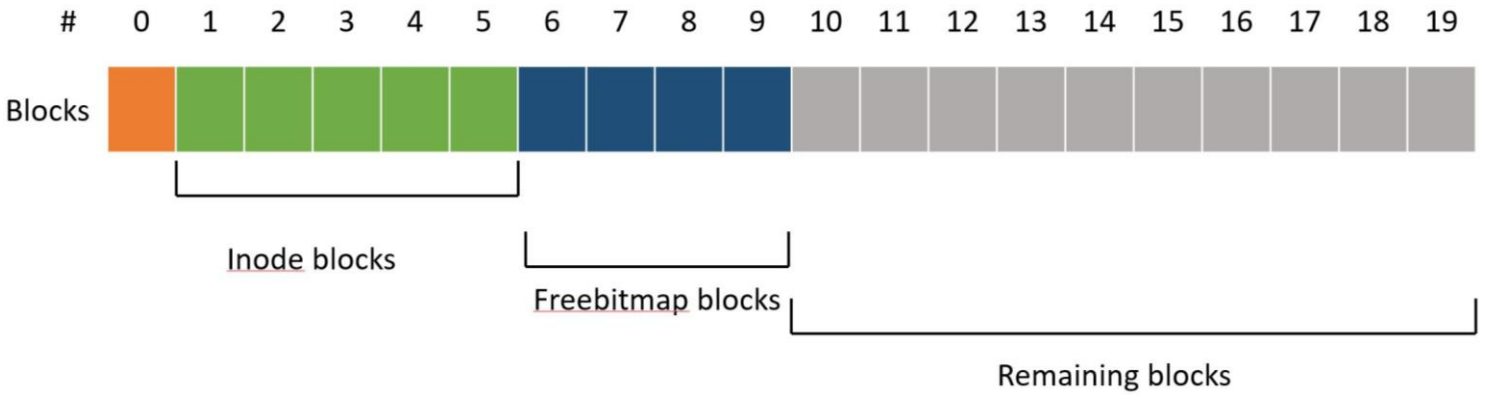
A closer look



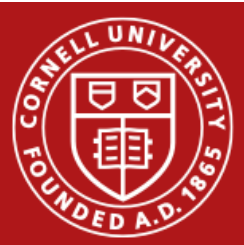
0 → Not used

1 → Used

How many freebitmapblocks?

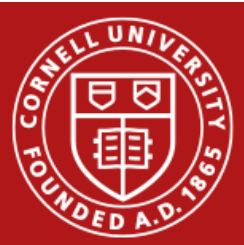


$$\frac{numblocks - 1 - \left(\frac{num\_inodes}{INODES\_PER\_BLOCK}\right)}{BLOCK\_SIZE * 8 + 1}$$



Filesystem checker (fsck)

- Checks if the filesystem is consistent.
- A filesystem is inconsistent if:
  - A datablock is in use but marked as free.
  - A particular block is both direct block and indirect block.
  - A particular datablock is pointed to more than once by one more more inodes.



Specifics

- You should only need to change ufsdisk.c/h
- Add ufsdisk.o to the list of OBJECTS in the makefile
- treedisk_chk.c could be inspiration for the ufsdisk_chk.c
- Keep an eye out for updates and FAQ on Piazza
- Start early, thank yourself later.