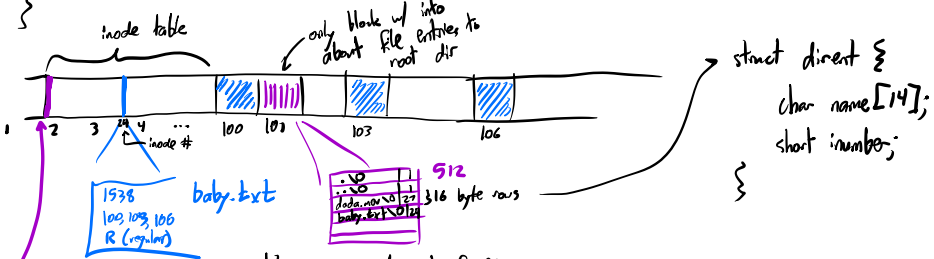
Lecture 3

From last class: 50-60% of file system implemented

```
struct inode {
  size_t size;          // limits filesize to something really small :(
  short blocknums[8];
  char type; // reg. file or directory?
  permissions, timestamps, owners, etc.
};
```

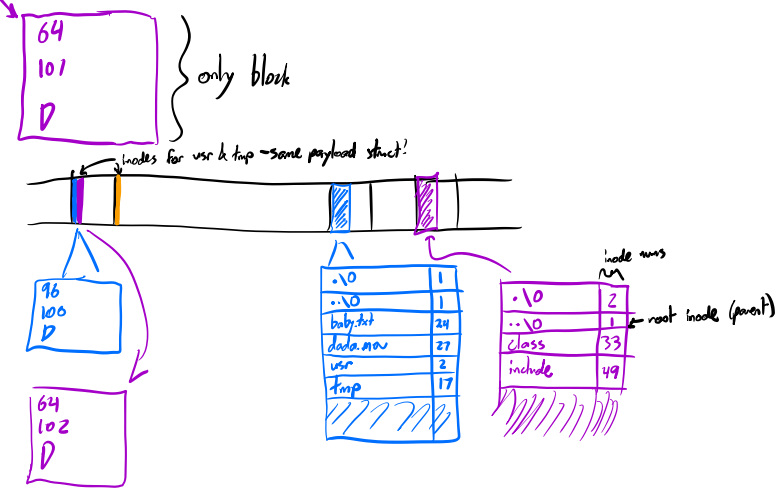



```
struct dirent {
  char name[14];
  short inumber;
};
```

haven't done directories yet! → a special sort of file

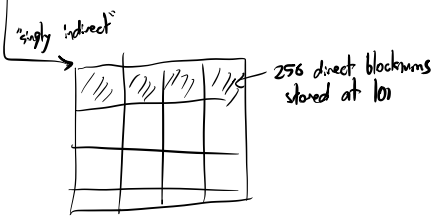
for now: support root dir! ↳ contents store filenames & associated inode nums

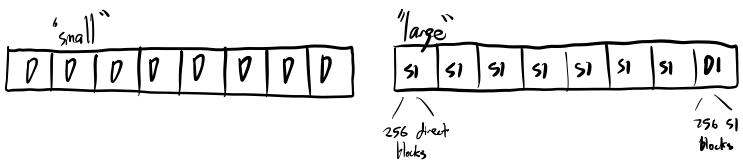
want to map baby.txt → 24 (the # inode it is)



one problem not yet addressed: files only 4096 bytes!

assuming 8 blocknums are all direct - if file is big, ...





Major design principles

Modularity & Layering

↳ special kind of modularity - org. of modules interact in some hierarchical manner

- symbolic link layer ↳ each layer only opens the interface of layer above it

- absolute path name layer
- path name layer
- file name layer
- inode number layer
- file layer (how to aggregate sequences of blocks & store payloads)
- block layer

Naming & Name Resolution