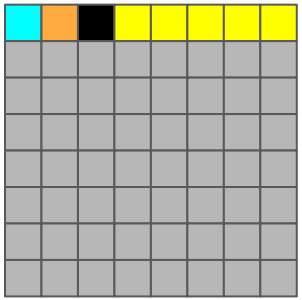
CS 537
P5b: xv6 File System

Xiangjin Wu

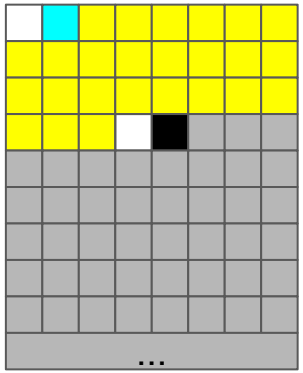
Aug 5, 2017

vsfs (OSTEP) vs xv6

- BSIZE = 4K
- Superblock (0)
- Inode bitmap (1)
- Data bitmap (2)
- Inode table (3-7)
- Data region (8-63)



- BSIZE = 512
- Reserved (0)
- Superblock (1)
- Inode table (2-26)
- Reserved (27)
- Data bitmap (28)
- Data region (29-1023)



- Smaller block
- More blocks
- Reserved blocks
- No inode bitmap
- Data bitmap after inode table

xv6 File System Structures

- include/fs.h (on-disk, p5a)

```
struct superblock {
  uint size;         // Size of file system image (blocks)
  uint nblocks;      // Number of data blocks
  uint ninodes;      // Number of inodes.
};


struct dinode {
  short type;           // File type
  short major;          // Major device number (T_DEV only)
  short minor;          // Minor device number (T_DEV only)
  short nlink;          // # of links to inode in the fs
  uint size;            // Size of file (bytes)
  uint addrs[NDIRECT+1];   // Data block addresses
};


struct dirent {
  ushort inum;
  char name[DIRSIZ];
};
```

- kernel/file.h (kernel memory, p5b)

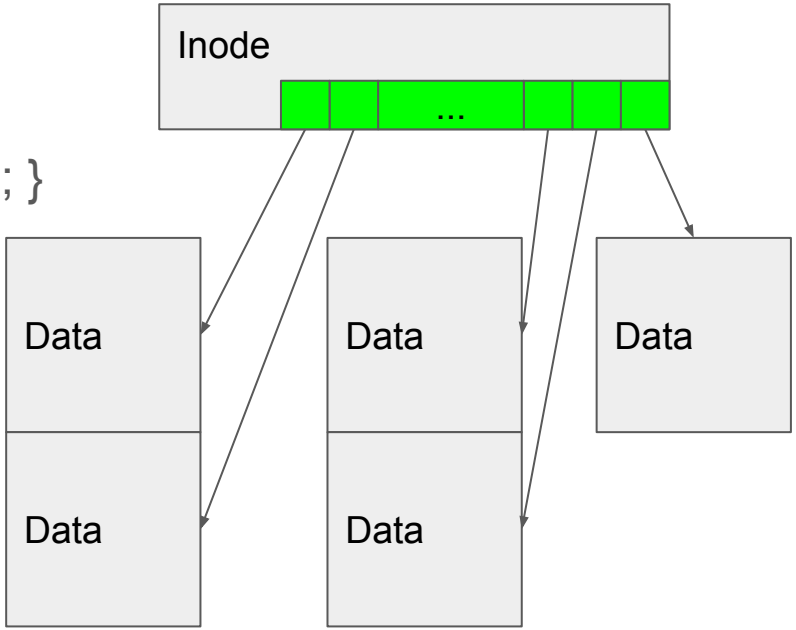
```
struct file {
  enum { FD_NONE, FD_PIPE, FD_INODE } type;
  int ref; // reference count
  char readable;
  char writable;
  struct pipe *pipe;
  struct inode *ip;
  uint off;
};


struct inode {
  uint dev;           // Device number
  uint inum;          // Inode number
  int ref;            // Reference count
  int flags;          // I_BUSY, I_VALID

  short type;         // copy of disk inode
  short major;
  short minor;
  short nlink;
  uint size;
  uint addrs[NDIRECT+1];
};
```

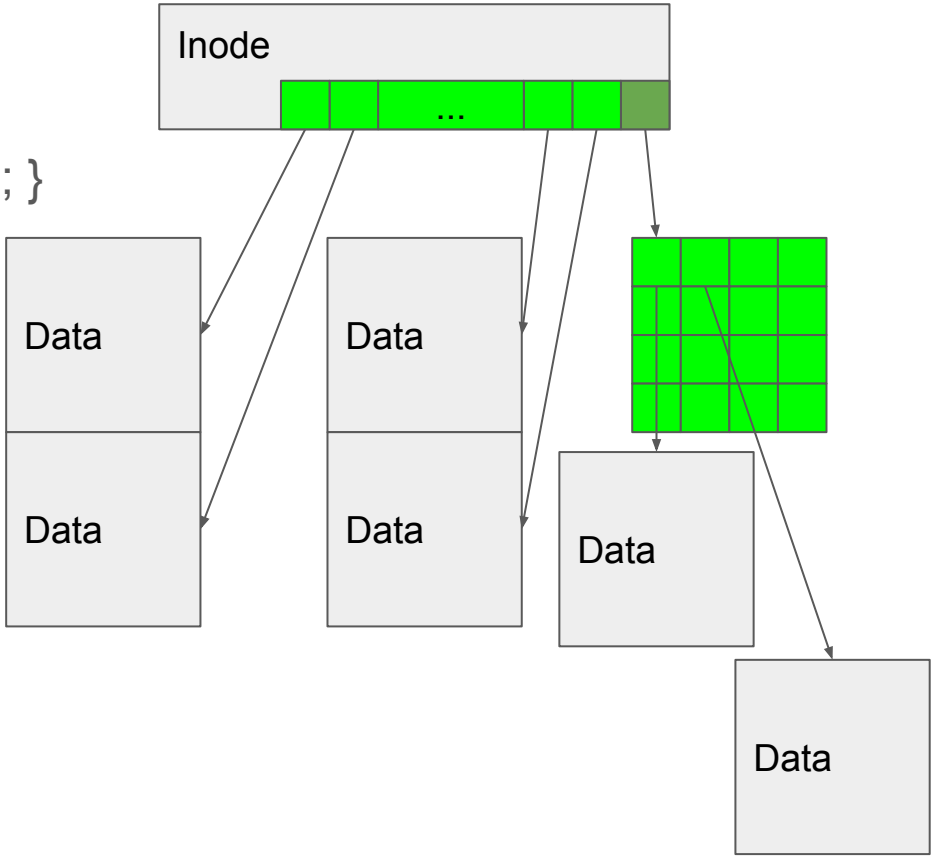
Large Files

- BSIZE = 512, NDIRECT = 12
- struct inode { uint addrs[NDIRECT+1]; }
- Maximum file size = 512 * 13 = 6.5 K



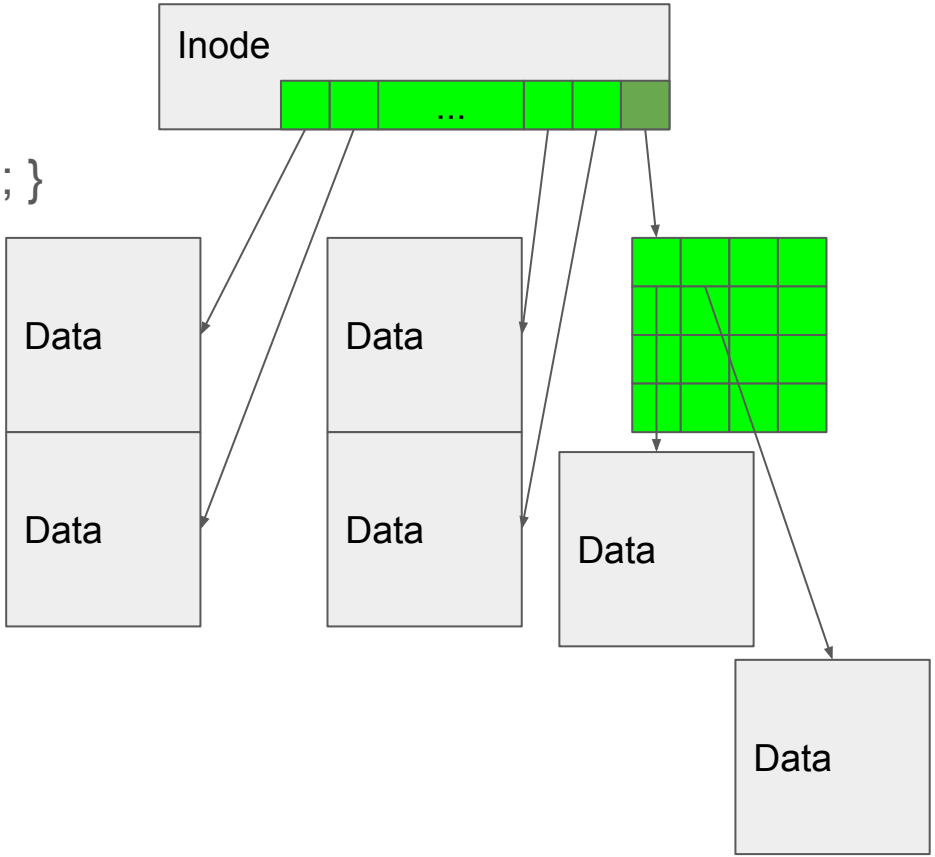
Large Files

- BSIZE = 512, NDIRECT = 12
- struct inode { uint addrs[NDIRECT+1]; }
- Maximum file size = 512 * 13 = 6.5 K
- Indirect pointer



Large Files

- BSIZE = 512, NDIRECT = 12
- struct inode { uint addrs[NDIRECT+1]; }
- Maximum file size = 512 * 13 = 6.5 K
- Indirect pointer
- Max = 512 * (12 + BSIZE/4)



xv6 File System Functions

filename	structure	function
kernel/sysfile.c	syscall args	sys_open, sys_read, sys_write, sys_close, sys_unlink
kernel/file.c	fd	fileread, filewrite, fileclose
kernel/fs.c	path/dir	dirlookup, dirlink, namei, nameiparent
kernel/fs.c	inode table	ialloc, iupdate, iget/iput, ilock/iunlock
kernel/fs.c	inode data pointer	bmap, itrunc, readi, writei
kernel/fs.c	data	balloc, bfree
kernel/bio.c	block	bread, bwrite, bget/brelse
kernel/ide.c	disk	iderw

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 0 0 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (0, “”) (0, “”) (0, “”) (0, “”) (0, “”) (0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	0
I#2: T_UNUSED I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 0 0 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (0, “”) (0, “”) (0, “”) (0, “”) (0, “”) (0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	0
I#2: T_FILE, addrs={0,0} I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 0 0 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}, sz	0
I#2: T_FILE, addrs={0,0} I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 0 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	0
I#2: T_FILE, addrs={7,0} I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 0 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,0} I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 1 0
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,8} I#3: T_UNUSED	0
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 1 1
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,8} I#3: T_UNUSED	9
0	0

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 1 1
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,8} I#3: T_UNUSED	9
0	world!

xv6 File System Example

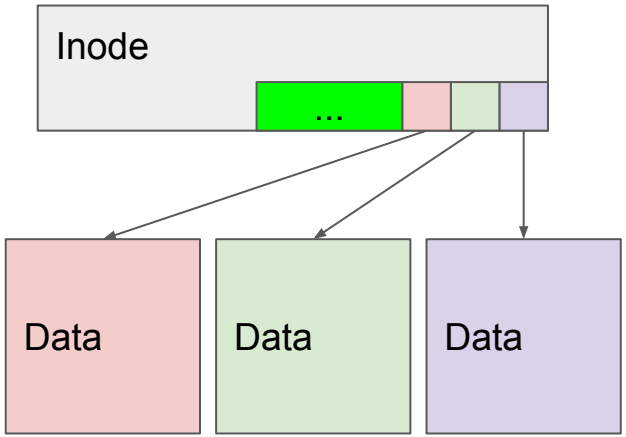
0	D-Bmap: 1 1 1 1 1 1 1 1 1 1
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,8} I#3: T_UNUSED	9
0	world!

xv6 File System Example

0	D-Bmap: 1 1 1 1 1 1 1 1 1 1
SB: size = 1024 nblocks=995, ninodes=200	(1, “.”) (1, “..”) (2, “foo.txt”) (0, “”)(0, “”)(0, “”)(0, “”)(0, “”)
I#0: T_UNUSED I#1: T_DIR, addrs={6,0}	Hello,
I#2: T_FILE, addrs={7,8} I#3: T_UNUSED	9
0	world!

Smart Files

- Use the last pointer as direct pointer when the file is small



Smart Files

- Use the last pointer as direct pointer when the file is small
- Allocate the pointer block when it grows larger
- Smart file: T_SMART
- Create / read / write / remove