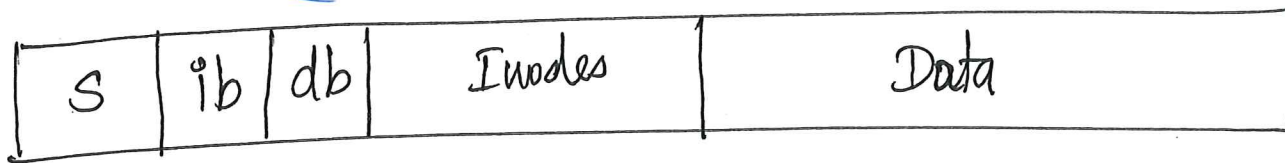


Last class:

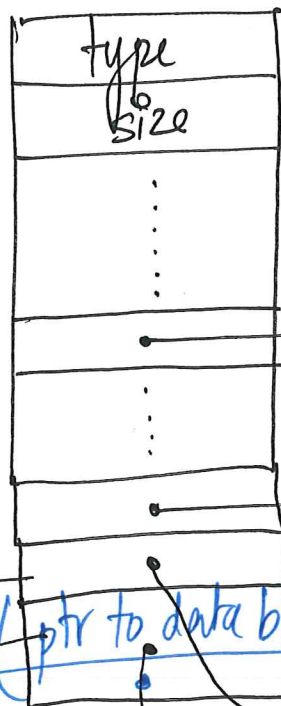
FS Implementation ++

FS Implementation
(VFS).

DS
Access Methods



Inode



1 block = 4KB

pointers to data blocks.

indirect ptr.

double indirect ptr.

ptrs in a data block

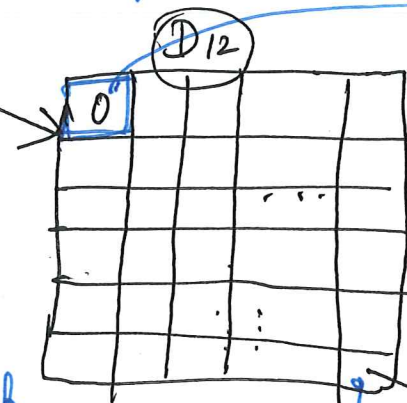
$$= \frac{4KB}{4 \text{ bytes}}$$

$$= 1024 \text{ ptrs in one data block.}$$

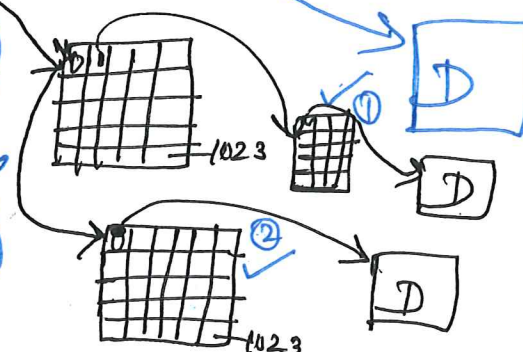
max size =

$$1024 \times 4KB = 4MB$$

$$+ 12 \times 4KB = 48KB$$



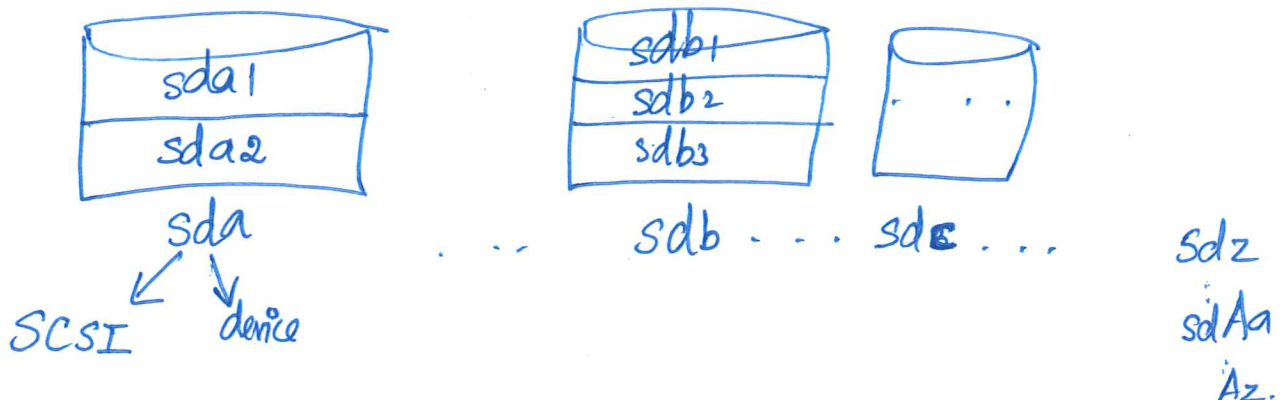
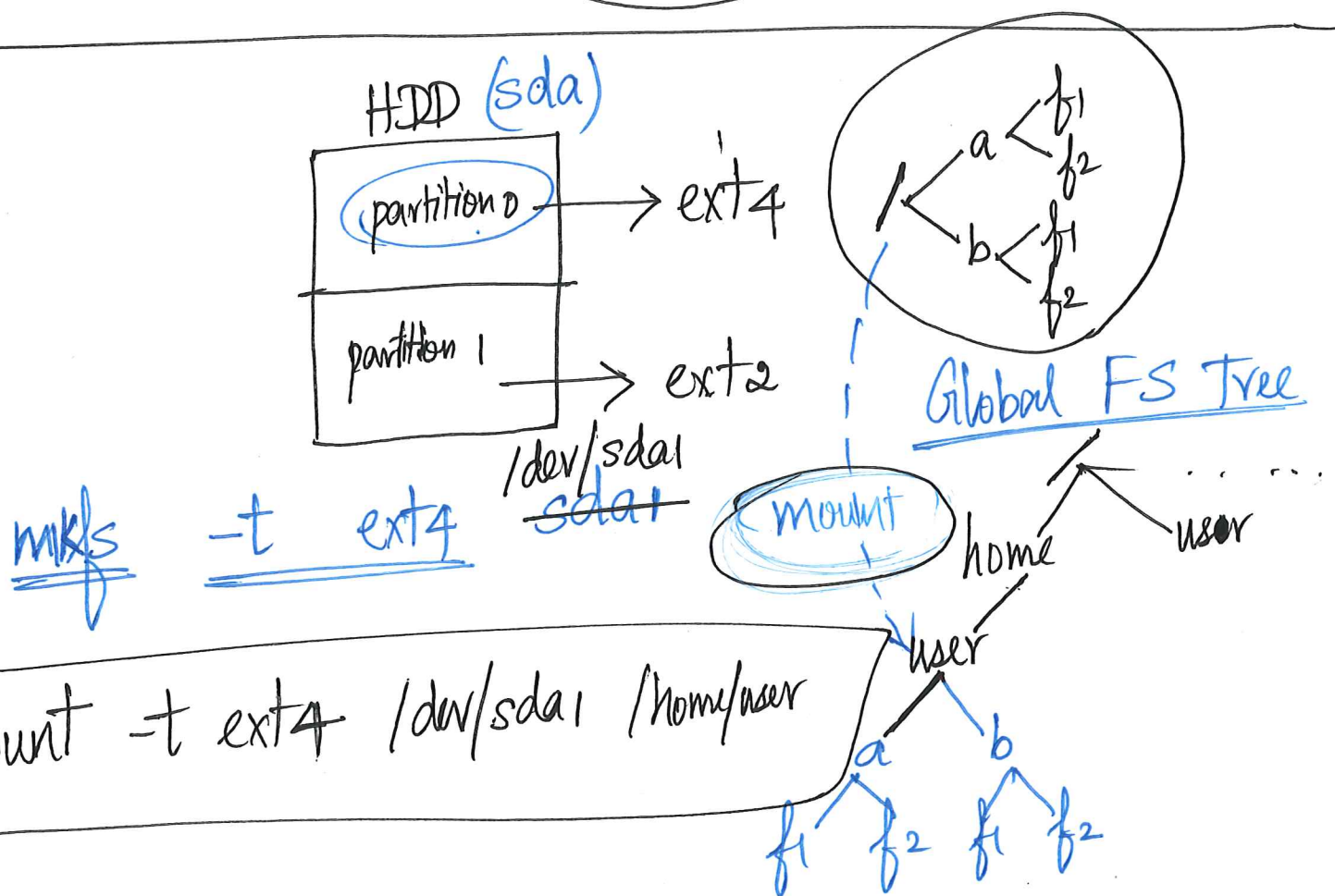
? 1023



with double indirect ptr.

$$\begin{aligned} \text{max file size} &= 1024 \times 1024 \times 4 \text{ KB} \\ &= 2^{10} \times 2^{10} \times 4 \times 2^{10} \text{ B} \end{aligned}$$

$$= \underline{4 \text{ GB}} + 4 \text{ MB} + 48 \text{ KB}$$



Access Methods: file creation + writing

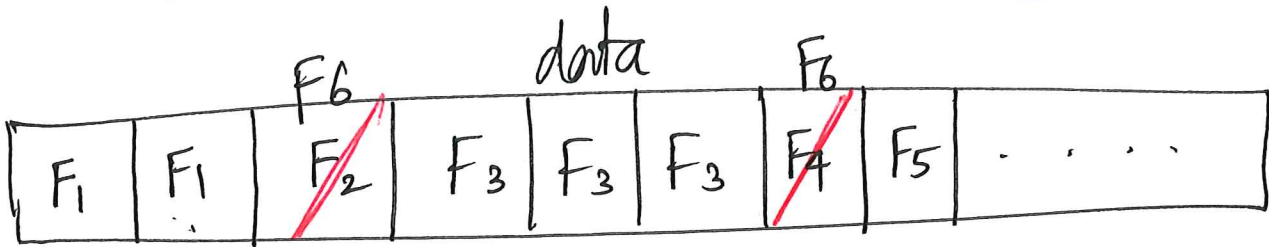
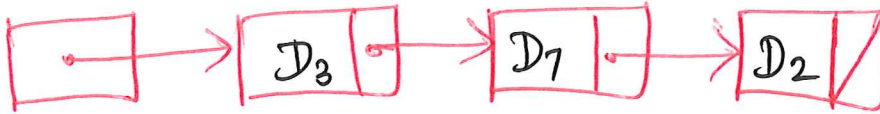
	inode data bitmap	inode data bitmap	root inode	foo inode	bar inode	root data	<u>foo</u> <u>data</u>	bar data[0]	bar data[i]		
create /foo/bar	R W		R	R		R	R	R			
write()		R W			R						
					W						

Problems?

UNIX FS - Ken Thompson

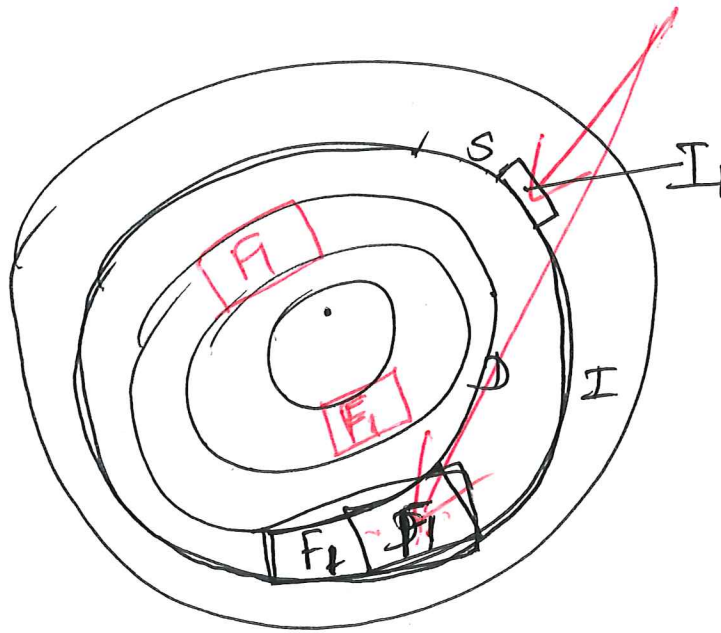


Freelist



block size = 512 bytes

F6
(2 blocks)

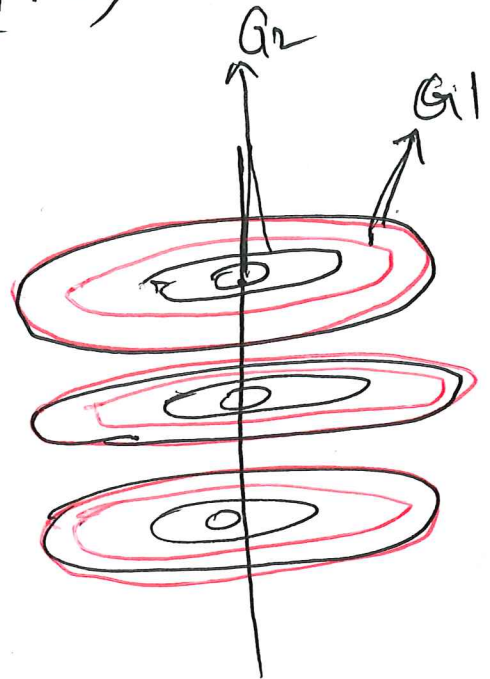
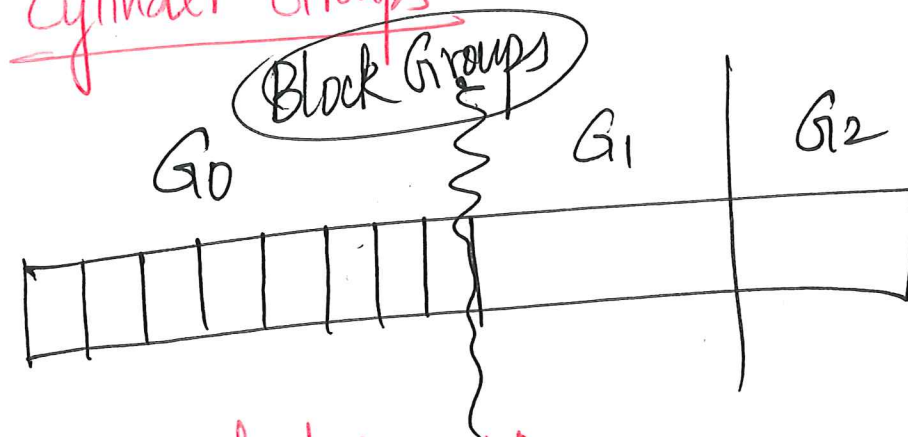


seek
+
rotation

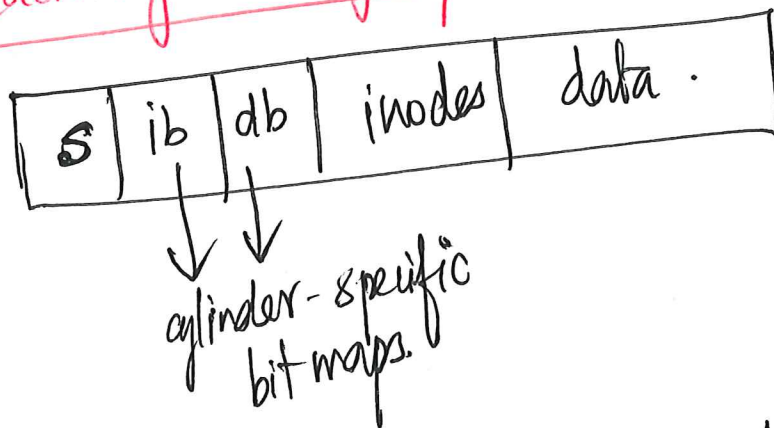
Fast File System (FFS)

"disk-aware"

Cylinder Groups



in each cylinder group



Policy: keep related data together!

How to place files?

inode + data $\rightarrow$ same cylinder group.

all the files in a dir $\rightarrow$ "

$1/d_1/f_1$ $1/d_1/f_2$ $-1/d_1/f_3$ $1/d_2/f_4$
 Group # Inodes Data

0	1 - - - -	1 - - - - -
1	<u>d1</u> f1 f2 f3 f5	<u>d1</u> f1 f2 f3 f5 f5 f5 f5 f5 f5
2	<u>d2</u> f4 - - - -	<u>d2</u> f4 f5 f5 f5 f5 - - - -

$1/d_1/f_3$ → ~~3~~¹⁰ data blocks

0	1 - - - -	
1	<u>d1</u> f1 - - - -	f f f - - - - f
2		f f . . . f
		. . .
10		f f f f . . . f

Positioning Time + Transfer Time

10 ms 10 ms

$\frac{1}{2}T$ $\frac{1}{2}T$

Disk transfer rate = 40 MB/s.

1 s $\rightarrow$ 40 MB

10 ms $\rightarrow$ $\frac{40 \text{ MB} \times 10 \text{ ms}}{1 \text{ s}}$

$\approx 40 \times 10^6 \text{ B} \times 10 \times 10^{-3}$

$\approx \underline{\underline{400 \text{ KB}}}$

Amortization.

90% time for Data Transfer.

P.T
10 ms

D.T
90 ms

1 s $\rightarrow$ 40 MB

90 ms $\rightarrow$ $\frac{40 \text{ MB} \times 90 \text{ ms}}{1 \text{ s}}$

$= 360 \times 10^6 \times 10^{-3}$

$\approx \underline{\underline{3.6 \text{ MB}}} \text{ (chunk size).}$

(6)

FFS - DID NOT use Amortization

Instead:

block size = 4KB
Disk addr = 32-bit

