

OSTEP Chapter 41

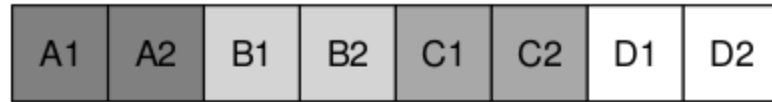
ECE 3600, Fall 2022

Table of Contents

- [1. Fragmentation](#)
- [2. Cylinder Groups](#)
- [3. Block Groups](#)
- [4. Examples](#)
- [5. Large Files](#)

1. Fragmentation

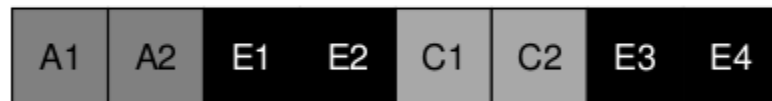
For example, imagine the following data block region, which contains four files (A, B, C, and D), each of size 2 blocks:



If B and D are deleted, the resulting layout is:

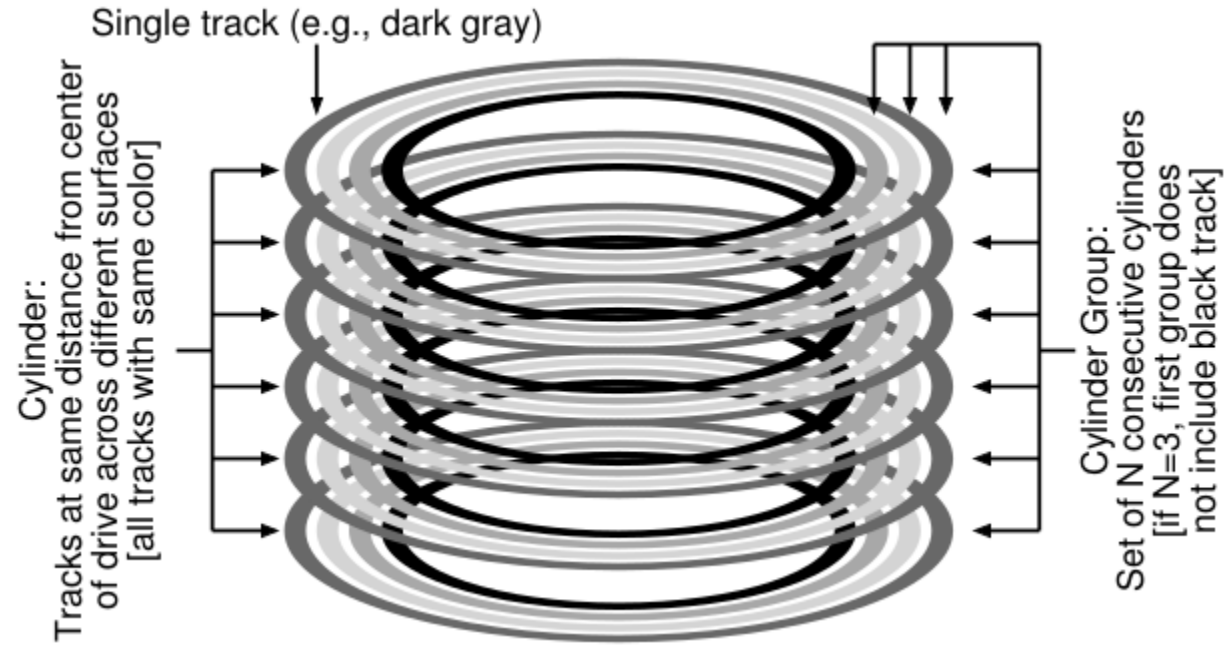


As you can see, the free space is fragmented into two chunks of two blocks, instead of one nice contiguous chunk of four. Let's say you now wish to allocate a file E, of size four blocks:

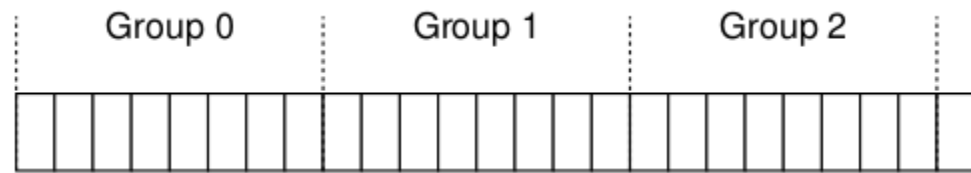


You can see what happens: E gets spread across the disk, and as a result, when accessing E, you don't get peak (sequential) performance from the disk. Rather, you first read E1 and E2, then seek, then read E3 and E4. This fragmentation problem happened all the time in the old UNIX file system, and it hurt performance. A side note: this problem is exactly what disk **defragmentation** tools help with; they reorganize on-disk data to place files contiguously and make free space for one or a few contiguous regions, moving data around and then rewriting inodes and such to reflect the changes.

2. Cylinder Groups



3. Block Groups



in each group



keep related stuff together, unrelated stuff far apart

balance allocation of directories

allocate file data in same group as inode and directory

4. Examples

Using [ffs.py](#):

```
$ cat in.example1
```

```
dir /a
dir /b
file /a/c 2
file /a/d 2
file /a/e 2
file /b/f 2
```

```
$ python ./ffs.py -f in.example1 -c -d 10
```

group	inodes	data
0	/-----	/-----
1	acde-----	acdde-----
2	bf-----	bff-----
3	-----	-----
4	-----	-----
5	-----	-----
6	-----	-----
7	-----	-----
8	-----	-----
9	-----	-----

```
$ python ./ffs.py -f in.example1 -c -d 10 -I
```

group	inodes	data
0	/-----	/-----
1	a-----	a-----
2	b-----	b-----
3	c-----	cc-----
4	d-----	dd-----
5	e-----	ee-----
6	f-----	ff-----
7	-----	-----
8	-----	-----
9	-----	-----

5. Large Files

```
$ cat in.example2
```

```
file /a 30
```

```
$ python ./ffs.py -f in.example2 -c
```

```
group inodes      data
  0 /a----- /aaaaaaaaa aaaaaaaaaa aaaaaaaaaa
  1 ----- a----- -----
  2 ----- -----
  3 ----- -----
  4 ----- -----
  5 ----- -----
  6 ----- -----
  7 ----- -----
  8 ----- -----
  9 ----- -----
```

```
$ python ./ffs.py -f in.example2 -c -L 4
```

```
group inodes      data
  0 /a----- /aaaa----- -----
  1 ----- aaaa----- -----
  2 ----- aaaa----- -----
  3 ----- aaaa----- -----
  4 ----- aaaa----- -----
  5 ----- aaaa----- -----
  6 ----- aaaa----- -----
  7 ----- aa----- -----
  8 ----- -----
  9 ----- -----
```