# CIS 452 - Operating Systems Concepts

## Nathan Bowman

## Images taken from Silberschatz book

---

# Contiguous Allocation

When moving data between disk and memory, a **block** of data is transferred

There is physical block size, which is minimum amount that can be transferred by disk controller

OS also sets logical block size for filesystem

Logical block size is smallest amount OS will transfer, must typically be greater than or equal to physical block size

When we talk about transfering a block, we mean logical block

Recall that OS provides abstraction of file so user can use human-friendly names and operations
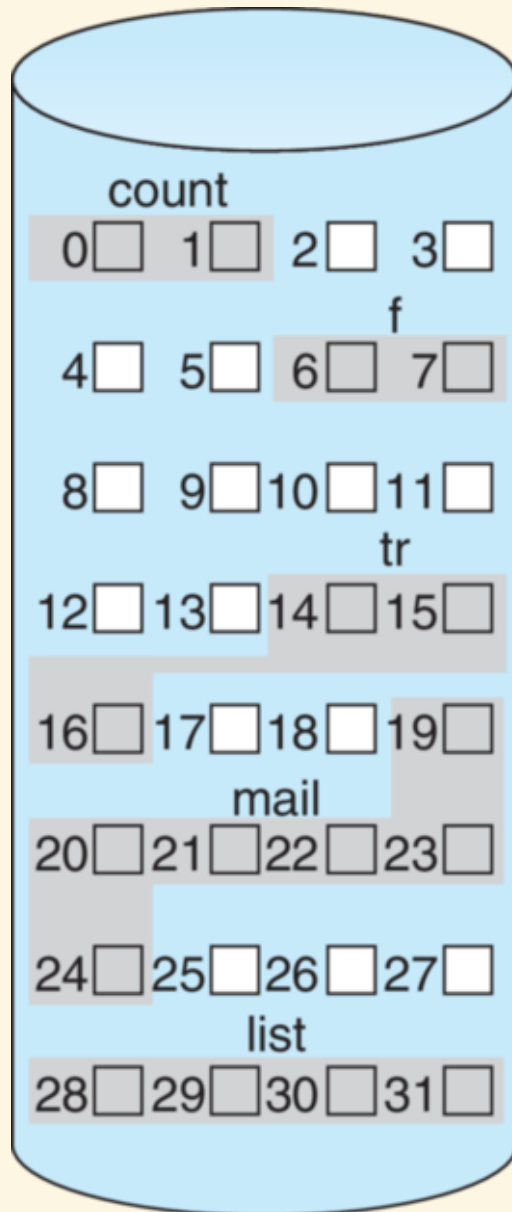
How are "files" actually stored on disk?

Just as with memory, physical storage layout may not be as simple as logical layout

Start with simplest idea: contiguous allocation

**Contiguous allocation** -- each file occupies set of contiguous blocks on disk

Keeping track of storage very simple -- record starting location and size

count

0 □  1 □  2 □  3 □

f

4 □  5 □  6 □  7 □

8 □  9 □ 10 □ 11 □

tr

12 □ 13 □ 14 □ 15 □

16 □ 17 □ 18 □ 19 □

mail

20 □ 21 □ 22 □ 23 □

24 □ 25 □ 26 □ 27 □

list

28 □ 29 □ 30 □ 31 □

directory

| file | start | length |
| --- | --- | --- |
| count | 0 | 2 |
| tr | 14 | 3 |
| mail | 19 | 6 |
| list | 28 | 4 |
| f | 6 | 2 |

Recall discussion of sequential vs direct access

Both types of access very efficient in contiguous allocation

To reach `ith` block, simply access `(start block) + i`

Because of how hard disks work, sequential access also efficient when successive blocks stored next to one another

Same downsides exist that were seen in contiguous memory allocation

Both are examples of general dynamic storage-allocation problem

Keep track of free holes of disk space, and choose strategy such as first-fit or best-fit to allocate space for files

Leads to external fragmentation

Recall that in external fragmentation, free space may be split up into small chunks and cannot be used efficiently

One solution is to rearrange files regularly to low or high addresses on disk, thereby bringing all holes together into large chunk of free space

Disks are both much larger and much slower than memory, so this is very expensive

If you own computer with hard disk, may occasionally "defragment" hard drive

Another issue: how to know ahead of time how large file will be?

If file was placed into small hole and grows over time, may run out of space

Need to either notify user or transparently move file to larger hole, which is expensive

Some OSes allow extents to alleviate this problem

**Extent** is second (or third, or...) chunk of contiguous space allocated to same file

If file grows too large for allocated space, extent from somewhere else can be added

Downsides are that file no longer contiguous and bookkeeping increased -- also need to keep track of size and location of extents