University of BRISTOL

Computer Systems B

COMS20012

Introduction to Operating Systems and Security

bristol.ac.uk

University of BRISTOL

Files on disk

bristol.ac.uk

It is all about abstractions

- OS see storage as a large addressable array of bytes
- User space wants better abstraction
  - Naming: /pics/meme.jpg instead of bytes between 24,048 to 28,156
  - Performance optimization
    - Caching
    - Pre-fetching
  - Transparent sector/block management
  - Reliability in case of crash/power failure

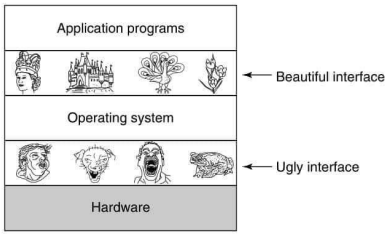


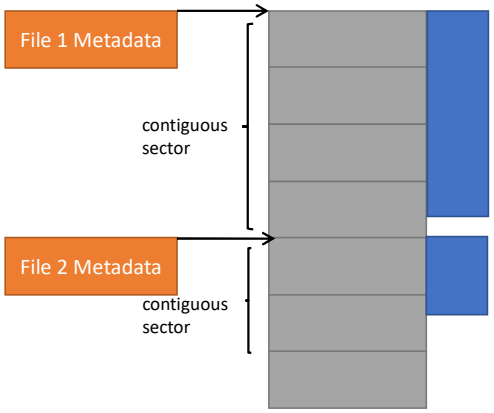
Figure 1-2. Operating systems turn ugly hardware into beautiful abstractions.

Modern Operating Systems, by Andrew Tanenbaum, Herbert Bos, Pearson

Files and Directories

- File: linear region of bytes that can grow and shrink
  - Associated with metadata
    - A name (e.g., meme.jpg)
    - Size in bytes
    - Access permissions (read/write/execute)
    - Statistics (e.g., creation and access dates)
  - OS is agnostic to the content of the file (userspace is to interpret it)
- Directory: container for files and other directories
  - Associate with a name + metadata
  - Nested directories can create a hierarchy (e.g., /home/bob/pictures/meme.jpg)

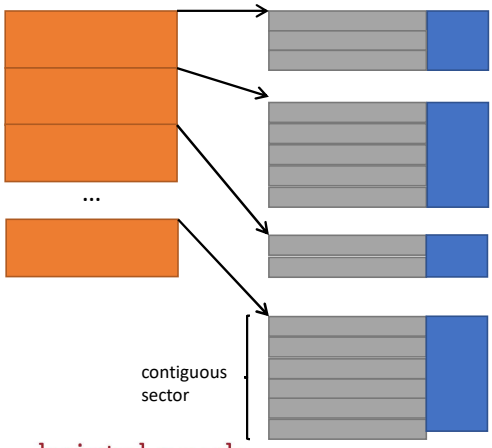
File as a single extent (1960's file systems)



- File metadata
  - Starting sector
  - Length in byte (last sector may not be fully used)
- Advantages
  - Simple
  - Small metadata
  - Good sequential and random I/O
- Problems
  - How much space to allocate to new files
  - What to do if a file grow beyond its allocation? Or Shrink?
  - External fragmentation

bristol.ac.uk

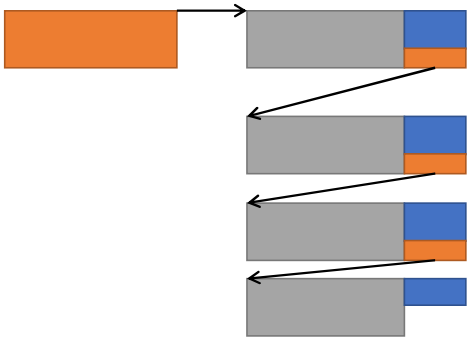
File as a collection of extents (IBM 360, ext4)



- Advantage
  - Metadata remain relatively small
  - Almost as good sequential I/O
    - Sequential offset calculation a bit tricky
  - Almost as good random I/O
- Challenges
  - How large the initial extent should be?
  - What to do if a file grows or shrink?
  - Improve on fragmentation!

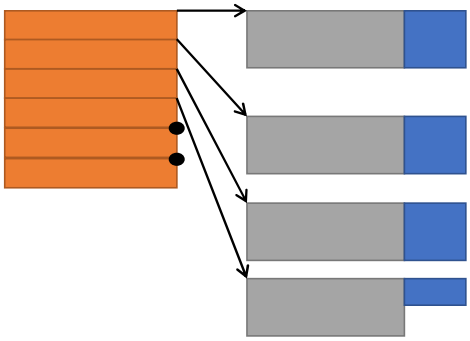
bristol.ac.uk

Files as linked list: (FAT)



- Advantages
  - Easy to shrink and grow
  - Low internal and external fragmentation
  - Sequential offset calculation is easy
- Disadvantages
  - Need to go through the list to find the part ones need
  - Some metadata at the end of each data block
  - Sequential I/O requires lots of seeks (on hard drive mechanical movement)

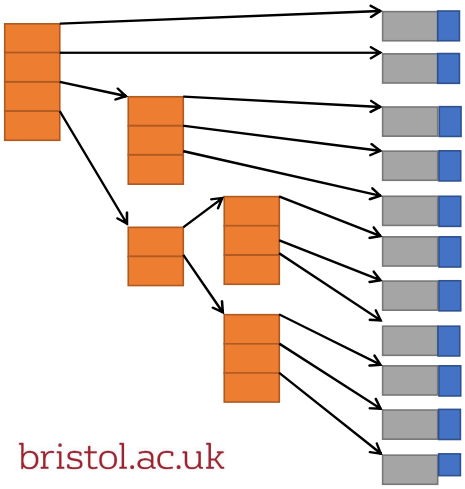
Files as flat indices



- Array mapping ranges to a block
- Advantages
  - Offset is easy to calculate
  - Low fragmentation
- Disadvantages
  - Maximum file size is fixed by number of entries in an index
  - Sequential I/O requires lots of seeks (on hard drive mechanical movement)

bristol.ac.uk

Files as hybrid indices (FFS, ext2, ext3)



- Top level index contains: direct pointers, indirect pointers, doubly indirect pointers etc.
- Advantages
  - Efficient for small files (do not materialize unused indirect list)
  - Big maximum file size (function of depth and index size)
  - Low fragmentation
- Disadvantages
  - Sometimes multiple disk access for a single read/write (need to fetch indirections)
  - Still require a large number of seek

Managing free space

Extent

- Break the disk
  - Fixed sized extents (4kb, 8kb, 12kb, …., 4MB)
  - Sorted by size arbitrary sized extents
- Maintain a list of unallocated extent
- Allocating N bytes of free space
  - e.g., best fit, worst fit, first fit etc.
- Trade-offs
  - Internal fragmentation
  - External fragmentation
  - Speed of finding match

Fixed size block

- Typically use bitmap to indicate which blocks are in use
- Allocation metadata is very compact
- Finding a single empty block is straightforward
- … finding region of N free blocks is more tedious

University of BRISTOL

Thank you

bristol.ac.uk