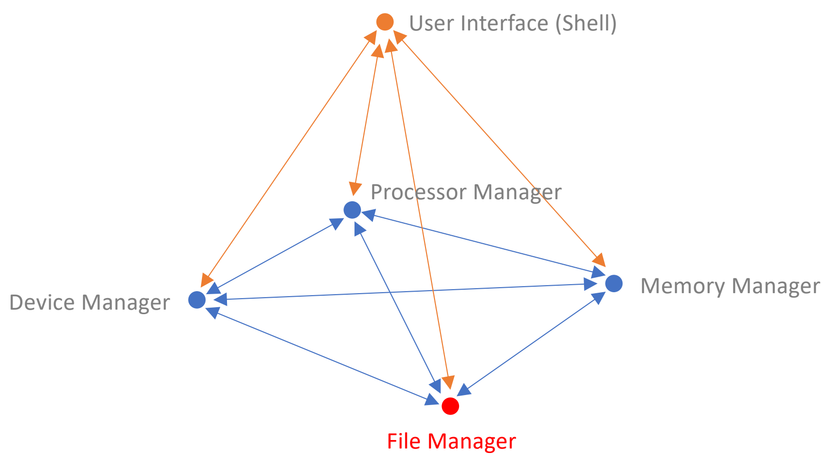
COMP124

Computer Systems

15 | File Manager | File Allocation | Disk Formats

Dr Stuart Thomason

File Manager



File Manager

- Keeps track of every file in the system
- Provides an abstracted view of files to the user (logical file system) and maps this on to the real organisation of data on disk (physical file system)

- File manager performs various tasks for the operating system…
  - Organises all files into folders (directories)
  - Manages the locations of files on disk
  - Maps logical structure on to physical locations
  - Enforces restrictions on who can access files (permission system)
  - Deals with standard file operations (open, close, read, write, delete…)
  - Provides standard system calls so other software can use files
  - Liaises with the device manager to access physical disks within the system

Disk Drives

- Mechanical drive (HDD) – magnetic platter with a read/write head that moves across it
- Solid state drive (SSD) – uses NAND flash chips to store data via electronic circuits
- The operating system provides an abstraction layer on top of the physical hardware

tinyurl.com/vhk4xbt5

HDD vs. SDD

- HDDs are mechanical devices with moving parts
  - Slower access speeds (disk needs to spin, head needs to move)
  - Shorter lifespan from wear and tear
  - Uses more power (around 8 watts)
  - Risk from vibration and damage in transit

- SSDs have no moving parts
  - Faster access speeds (electronic circuits via system bus)
  - Longer lifespan
  - More energy efficient (around 2 watts)
  - Impervious to knocks and vibrations

- However, HDDs can store much more data (many multiple terabytes) and are cheaper

Disk Blocks

- Disks are based around the idea of blocks
  - Each disk is split into blocks of a certain size
  - Operating system chooses the block size (typically 4K = 4096 bytes)
  - Files fill up one or more blocks
  - Only one file can exist in each block
  - Space is wasted when a file doesn't fully fill a block

- Each physical disk can be divided into separate partitions
  - Each partition will be formatted with a particular disk format (eg. FAT)
  - This splits the disk (partition) into blocks of equal size
  - Disk format and block usage is determined by the file allocation method

Free List & Metadata

- Parts of each disk are reserved to store data about the disk and the files on it
  - Not all blocks are used for data (so the total storage size is less than the disk size)
  - Some blocks store the free list and other housekeeping information (metadata)

- The free list is a simple bit vector that records which blocks are free (ie. not used)
  - 1 means in use; 0 means free
  - 8 blocks can be represented by one byte
  - File manager can easily see which blocks are available for new files
  - Modern disk formats use a much more complex set of metadata (taking more room)

- Some disadvantages…
  - Can become out of sync with disk if held in memory (needs to be saved regularly)
  - Could be huge (160GB disk with 4K block size needs 5MB for the free list)

File Systems and System Calls

- Every file system has four parts
  - Physical storage on disk – Formatted as FAT, XFS, EXT4, or one of many others
  - Logical naming scheme – File and directory names (and the way they are nested)
  - Permissions model – Access control such as the permissions used by Linux
  - System calls – Standardised programmatic access to files via the kernel

- System calls include fopen, fclose, fread, fwrite, fseek, fflush
  - Any program can manipulate data in files through these standard calls
  - The same calls can be used regardless of the physical media or disk format
  - System calls provide an abstraction of the underlying disk hardware
    (might not even be a real disk at all - but looks and behaves like one)

- User programs (and code) see the logical view of the file system via these system calls

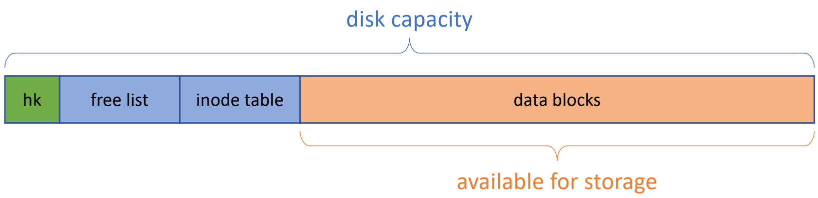
Logical and Physical Views

- Operating systems provide conveniences to make life easier for us
  - Filenames with extensions (eg. .docx, .pdf)
  - Folder hierarchies with nested folders and files
  - Shell with drag and drop to move or copy files
  - This is the logical view of a file system

- But the operating system doesn't follow (or need) these human conveniences
  - Directories and files are just bit patterns stored in blocks of the disk
  - OS needs to know which blocks belong to which files
  - Metadata about files and block usage is also stored on the disk
  - Arrangement and usage of blocks depends on the file allocation method
  - This is the physical view of a file system

Inodes and Metadata

- There must be some way to map the logical view on to physical locations on disk
  - We use inodes to point to physical blocks on disk
  - Each file has an inode that stores metadata about it
  - Also stores block information (eg. start block) according to allocation method
  - The inode table needs to be stored on the disk somewhere
  - Each disk format has its own housekeeping data that needs to be stored
  - The actual capacity of a disk will be reduced by all the space needed for metadata



File Allocation and Access

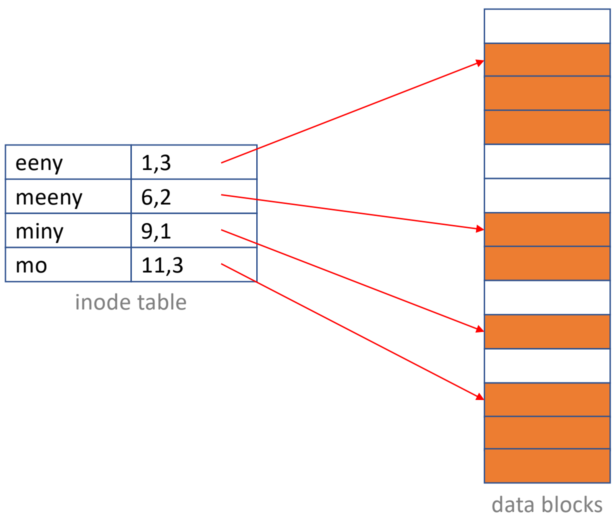
- There are several ways to allocate files to blocks on a disk
  - Contiguous
  - Linked
  - Indexed

- The disk format (file allocation method) dictates how easy it is to access blocks in a file

- Sequential file access…
  - Starts with the first block in a file
  - Reads each block in turn until the required data is found
- Direct file access…
  - Goes directly to the block that contains the required data
  - Skips over earlier blocks in the file

Contiguous Allocation

- Each file is allocated across contiguous blocks on the disk (ie. next to each other)



Contiguous Allocation

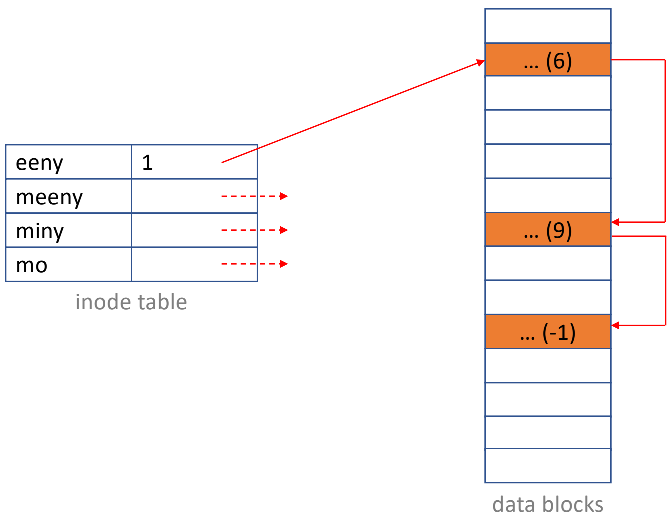
- The inode stores the start block number and number of blocks

- Advantages…
  - Fast for sequential access
  - Fast for direct access

- Disadvantages…
  - Fragmentation of free blocks
  - Might need regular compaction (defragmentation)
  - Needs policy for choosing which free blocks to allocate
  - Files don't (necessarily) have room to grow after initial allocation

Linked Allocation

- Each block contains a pointer to the next block (similar to a linked list)



Linked Allocation

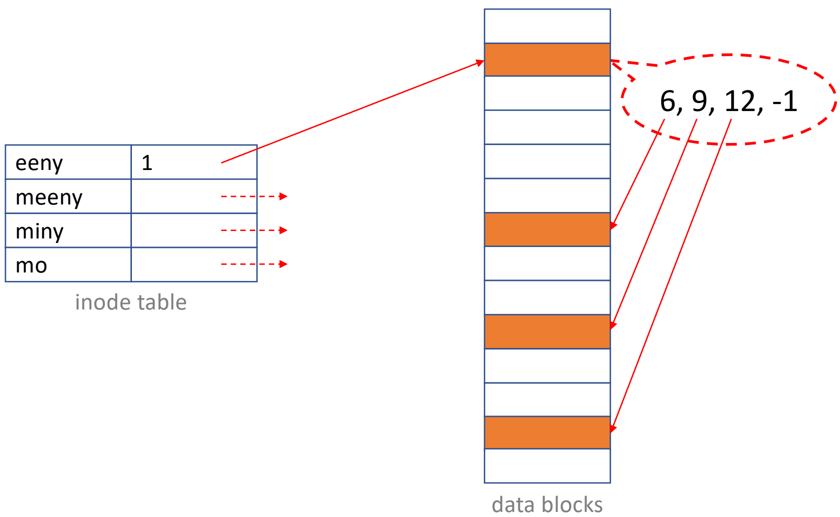
- The inode stores the start block and each block ends with the number of next block

- Advantages…
  - Easy to grow and shrink files
  - No danger of fragmentation

- Disadvantages…
  - Blocks are widely dispersed (worse in HDDs)
  - Sequential access less efficient
  - Direct access even worse (requires N reads to get to block N in the chain)
  - Danger of pointer corruption

Indexed Allocation

- First block holds index to all other blocks in a file



Indexed Allocation

- The inode stores number of the index block, which contains a list of all blocks for the file

- Advantages…
  - Each file's block information is held in one place
  - Very efficient for both sequential and direct access

- Disadvantages…
  - Blocks can still be widely dispersed across disk
  - Wastes a block for very small files
  - Can run out of pointers for large files (might need to chain index blocks)
  - Danger of index block becoming corrupted

File Allocation Table (FAT)

- Many Windows installations use the FAT disk format
  - Splits a physical disk into distinct sections
  - Uses indexed allocation but the index blocks are all stored in one place (the FAT)

- The disk is formatted as follows…
  - Boot sector – Contains housekeeping data (and bootstrap code on primary disk)
  - FAT – Master copy of the index blocks for each file (inode table)
  - FAT (copy) – Backup copy of the FAT to help recover from corrupted disks
  - Data region – Contains blocks that store the actual content of files
- Many USB sticks (flash drives) use FAT because it can be understood by all major operating systems (Windows, MacOS, Linux)

File Allocation Table (FAT)

- Indexed allocation method where the inode table is stored in its own part of the disk

- Advantages…
  - All pointer information held in one place
  - Easier to protect
  - No need for a separate free list
  - Direct access much more efficient

- Disadvantages…
  - Requires HDD head to move constantly between FAT area and file area
  - Space for FAT must be reserved when the disk is formatted
  - FAT will become huge for large disks

New Technology File System (NTFS)

- Default disk format for new Windows installations since Windows 3.1 (1993)
- Uses a master file table (MFT) but the general principle is the same as FAT
- Disk is formatted differently to provide many useful new features

  - Access control lists (similar to Linux permissions)
  - Encryption
  - Journaling
  - Hard links
  - File compression
  - System compression
  - Sparse files
  - Shadow copies
  - Disk quotas

Popular File Systems

- Windows…
  - FAT32
  - NTFS

- MacOS…
  - APFS – Apple File System
  - Can also read and write FAT disks
  - Can read NTFS disks but writing to them requires non-trivial system changes

- Linux…
  - Most installations use EXT4 (EXT2 and EXT3 are earlier versions)
  - EXT4 disks can’t be read by MacOS or Windows (so use FAT for portable disks)
  - Other popular systems include btrfs and ZFS