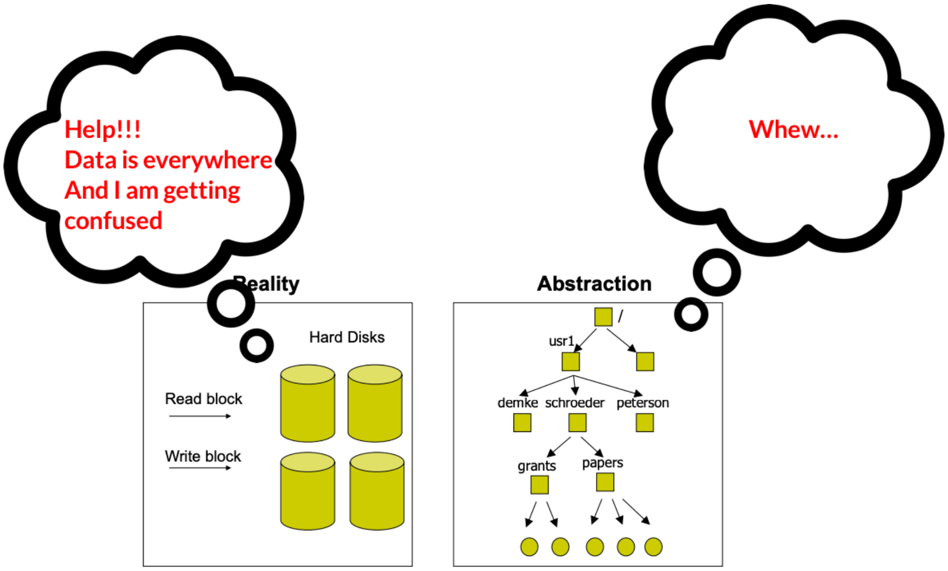
CSC369 Week 8 Notes

Hyungmo Gu

June 4, 2020

- File Systems
  - Is the part of operating system dealing with files [2]
  - Controls how data is stored and retrieved. [1]
    - Without a file system, data placed in a storage medium is one large body of data with no way to tell where it stops and the next begins



Refernces:

1) Wikipedia: File Systems, link
2) Tanebaum AS, Boss H. 2015. Modern Operating Systems. 4th Edition. New Jersy: Pearson Education, Inc.

- File Concept
  - Files

- ∗ Are logical units of information created by processes [1]
- ∗ Is named collection of data with some attributes
  1. Name
  2. Owner
  3. Location
  4. Size
  5. Protection
  6. Creation Time
  7. Time of Last Access

Refernces:

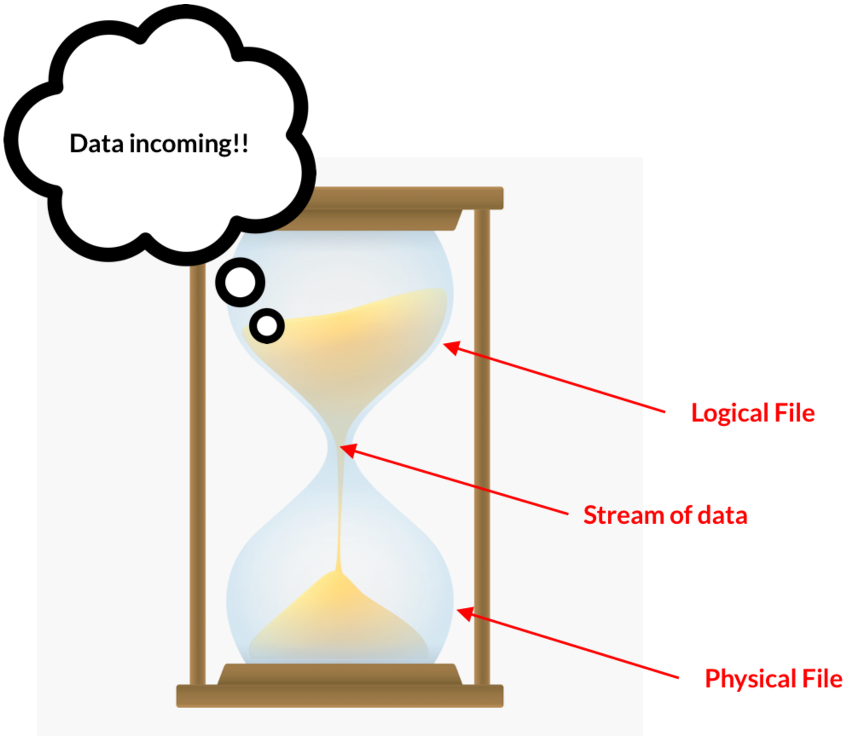
1) Tanebaum AS, Boss H. 2015. Modern Operating Systems. 4th Edition. New Jersy: Pearson Education, Inc.

- Directories
  - Are file system files for maintaining the structure of the file system [1]
  - Serves multiple purposes
    - ∗ *All* → Stores information about files (owner, permission, etc)
    - ∗ *Users* → provides a structured way to organize files
    - ∗ *File System* → provides a convinent naming interface that allows the implementation to separate **logical file** organization from **physical file** placement on the disk
      - · **Logical files:** Is a channel that connects the program to the physical file (Stream) [2]
      - · **Physical files:** A collection of bits stored in the secondary storage [2]

      Example:

      FILE* output;
      output = fopen(”sample.txt”, ”w”);

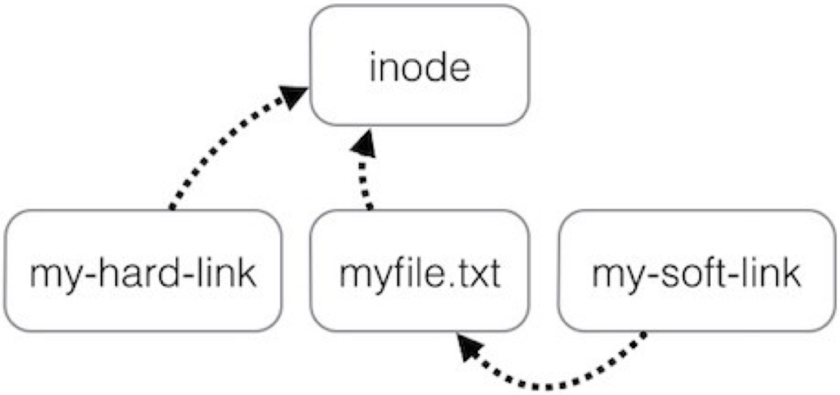
      Here, output is the logical file and sample.txt is the physical file



Refernces:

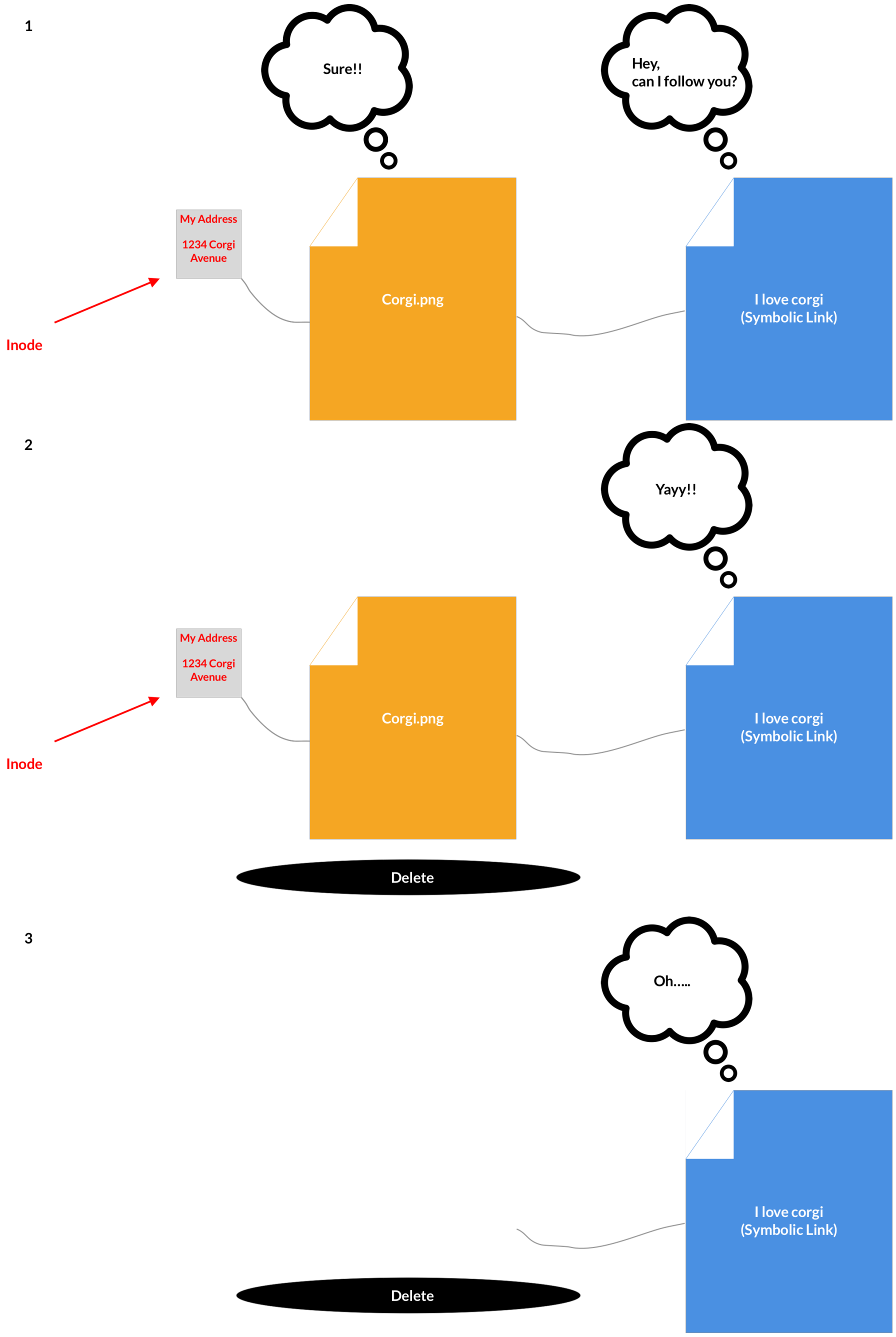
1) Tanebaum AS, Boss H. 2015. Modern Operating Systems. 4th Edition. New Jersy: Pearson Education, Inc.
2) Kumar, S. (2010). *File structures* [PowerPoint Slides]. Slide Share link

- What is a Directory at the OS Level?
- Operations on Directories
- Example Directory Operations
- Symbolic vs Hard Links

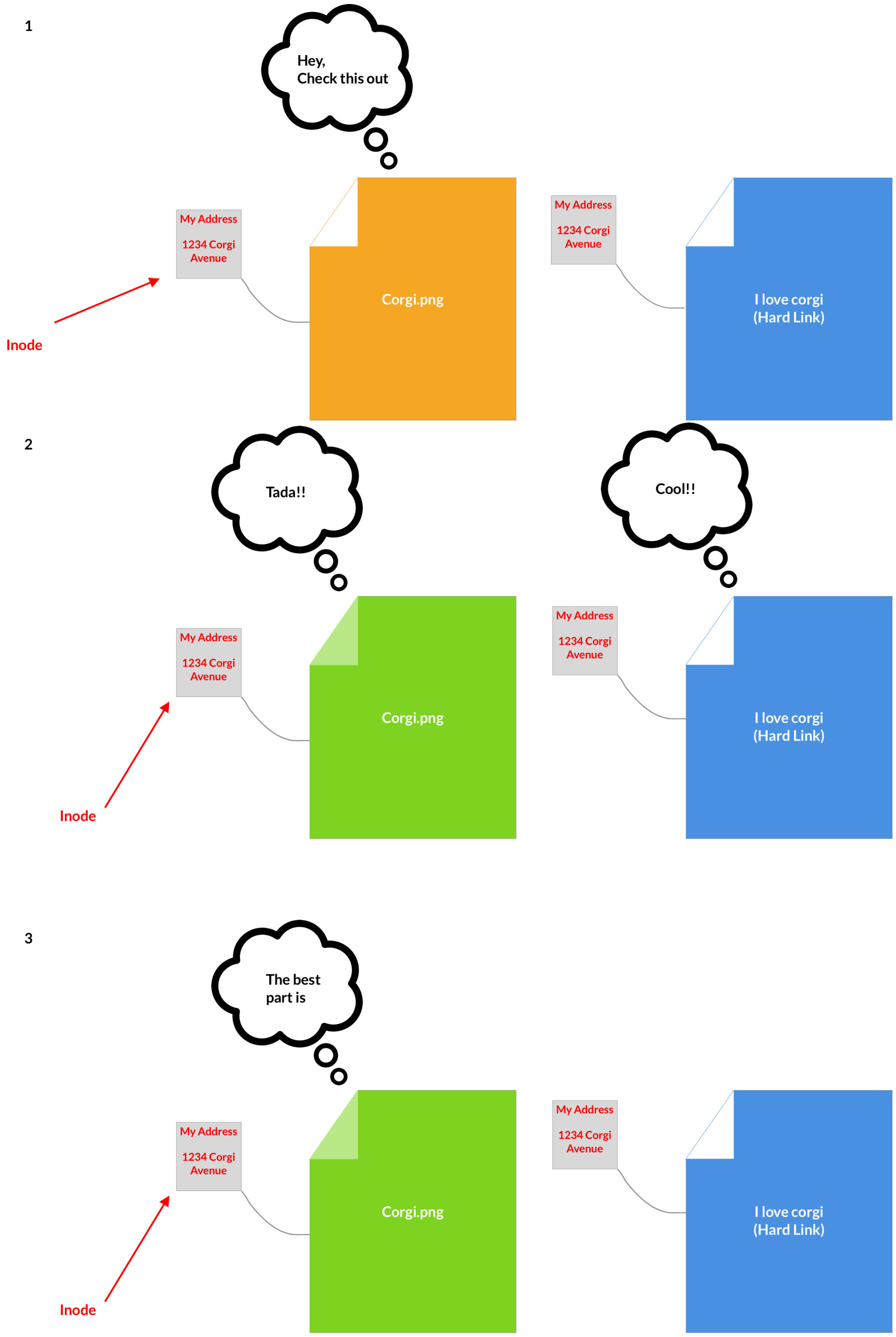


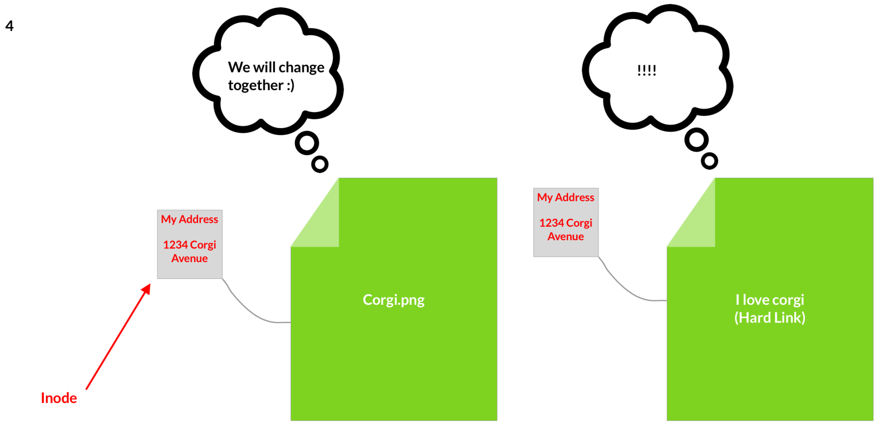
Andrew, Medium

- **Inode**
  - Is a database structure in a UNIX-style file system that describes a file system object such as a file or a directory [1]
  - Contains disk block location of the object's data [1]
  - Is a numerical equivalent of a full address [2]
- **Symbolic Link:**
  - Is directory entry containing "true" path to the file
  - Is a shortcut that reference to a file instead of inode value [2]



- **Hard Link:**
  - ∗ Is a direct reference to a file via its inode [2]
  - ∗ Is second directory entry identical to first

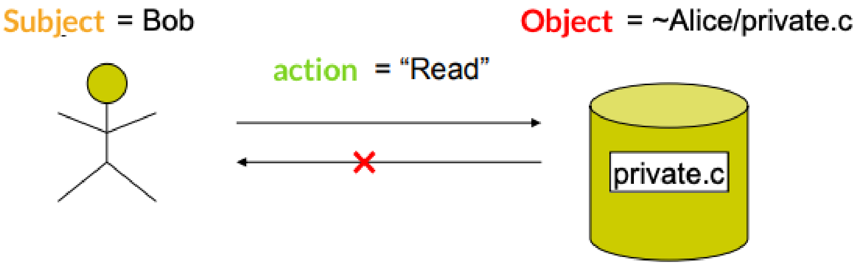

1
Hey, Check this out
My Address 1234 Corgi Avenue
Inode
Corgi.png
My Address 1234 Corgi Avenue
I love corgi (Hard Link)
2
Tada!!
Cool!!
My Address 1234 Corgi Avenue
Inode
Corgi.png
My Address 1234 Corgi Avenue
I love corgi (Hard Link)
3
The best part is
My Address 1234 Corgi Avenue
Inode
Corgi.png
My Address 1234 Corgi Avenue
I love corgi (Hard Link)



Referncces:

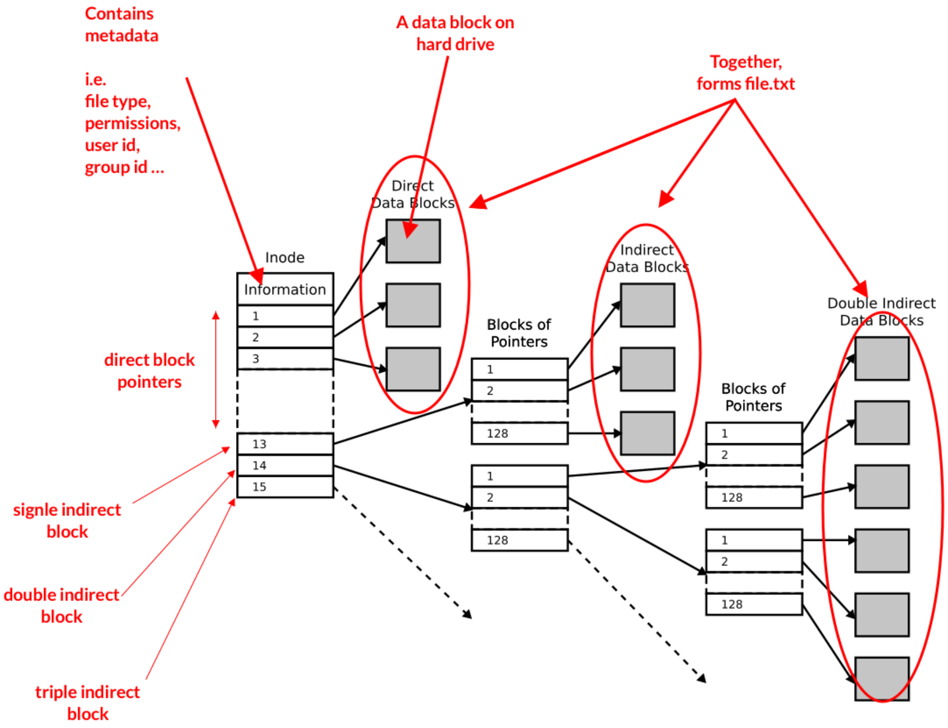
1) Wikipedia: inode, link
2) Andrew. (2018, January 16). *Hard links and Symbolic links — A comparison.* Medium. link

- File Sharing
- Protection
  - File systems implement some kind of protection system
    - Who can access a file
    - How they can access it
  - Protection system dictates whether given **action** by a given **subject** on a given **object** should be allowed
    - You can read and/or write your files, but others cannot
    - You can read "etc/motd", but you cannot write it



- Types of Access

- Representing Protection
- ACLs and Capabilities
- File System Implementation
- Directory Implementation
- Disk Layout Strategies
- Contiguous Allocation
- Linked Allocation
- Indexed Allocation: Unix Inodes
  - Each inode contains 15 block pointers
    - First 12 are direct block pointers
      - Stops here if files are small
    - Then single, double and triple indirect

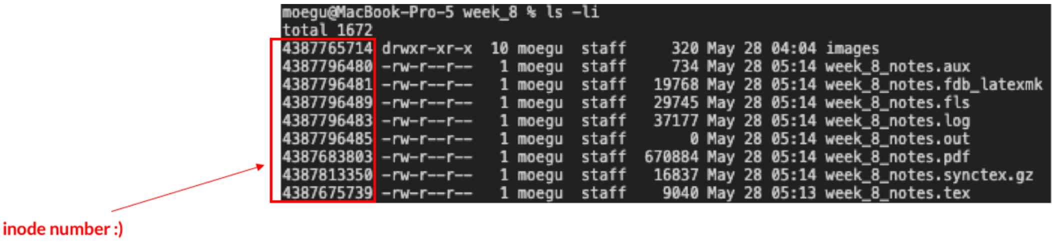


Refernces:

1) Wikipedia: Inode Pointer Structure, link

2) Udacity (2015). *Inode Structure* [online]. Available at: link (Accessed May 28th, 2020)

- Unix Inodes and Path Search
  - Unix Inodes
    - Is what we see on typing ‘ls -li’ command in terminal



```
moegu@MacBook-Pro-5 week_8 % ls -li
total 1672
4387765714 drwxr-xr-x  10 moegu  staff     320 May 28 04:04 images
4387796480 -rw-r--r--   1 moegu  staff     734 May 28 05:14 week_8_notes.aux
4387796481 -rw-r--r--   1 moegu  staff   19768 May 28 05:14 week_8_notes.fdb_latexmk
4387796489 -rw-r--r--   1 moegu  staff   29745 May 28 05:14 week_8_notes.fls
4387796483 -rw-r--r--   1 moegu  staff   37177 May 28 05:14 week_8_notes.log
4387796485 -rw-r--r--   1 moegu  staff       0 May 28 05:14 week_8_notes.out
4387683803 -rw-r--r--   1 moegu  staff  670884 May 28 05:14 week_8_notes.pdf
4387813350 -rw-r--r--   1 moegu  staff   16837 May 28 05:14 week_8_notes.synctex.gz
4387675739 -rw-r--r--   1 moegu  staff    9040 May 28 05:13 week_8_notes.tex
```

    - Describes where on the disk the blocks for a file are placed
    - inode information is loaded to main memory [1]
      - Only for the corresponding files that are open
      - NOT all are loaded

Refernces:

1) Tanebaum AS, Boss H. 2015. Modern Operating Systems. 4th Edition. New Jersy: Pearson Education, Inc.

- File Buffer Cache
  - Reads information from disk only once and then stores retrieved file blocks in memory until no longer needed []
    - Because reading from disk is slow
    - Is common to read same part of disk multiple times

Example:

1. Reading email message, read the message for an edit, and read the message again when copying to folder

Refernces:

1) Linux System Administrators Guide: Chapter 6. Memory Management, link