CS 3502
Operating Systems

File system

Kun Suo

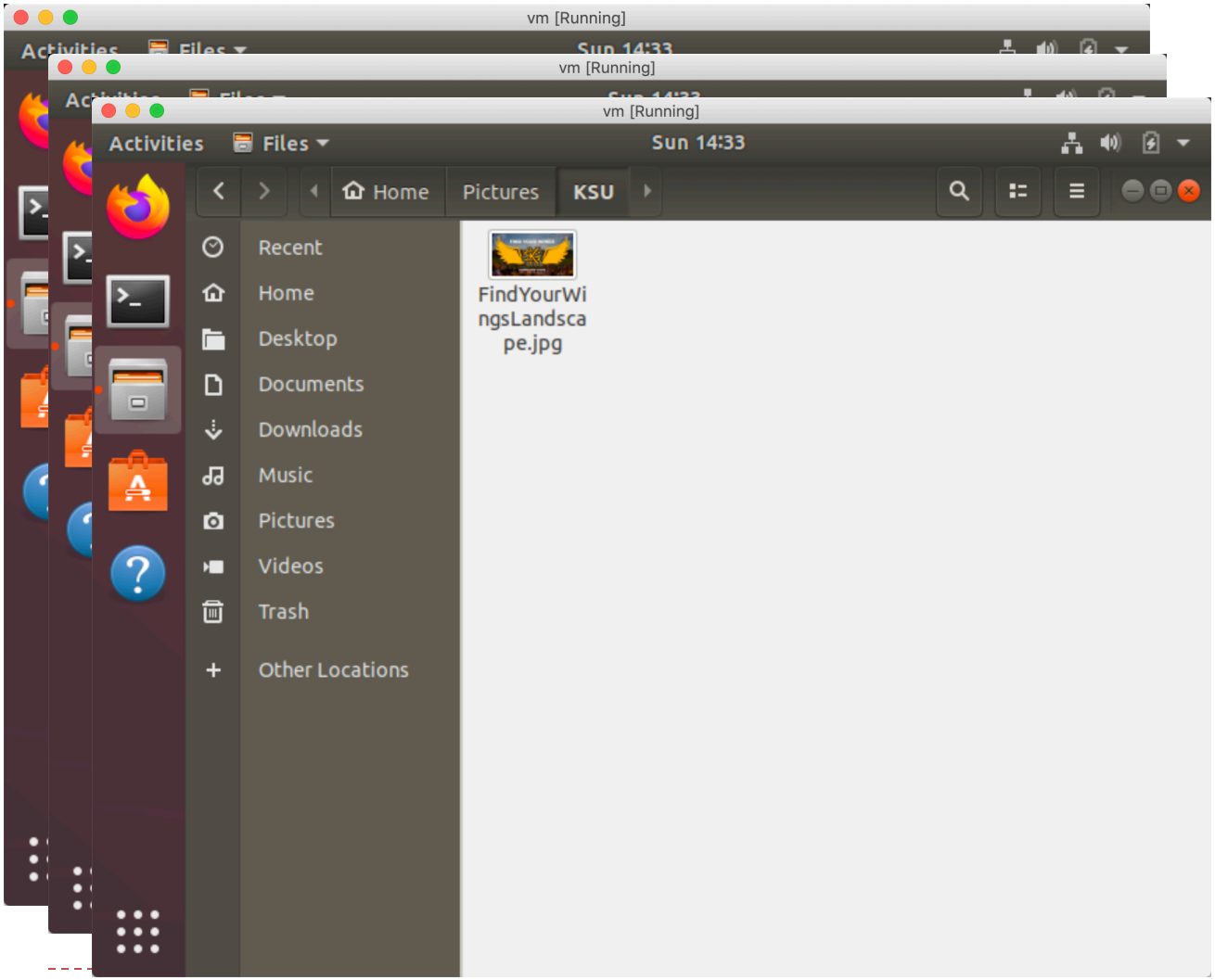
Computer Science, Kennesaw State University

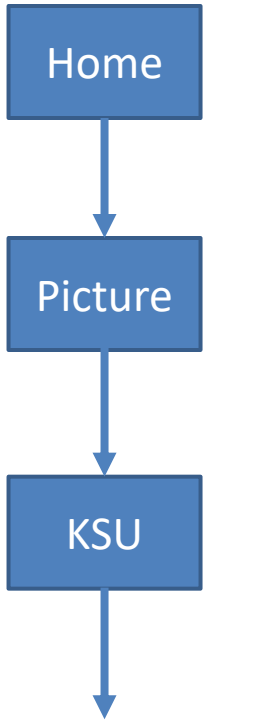
https://kevinsuo.github.io/

Outline

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system

What is a File System?

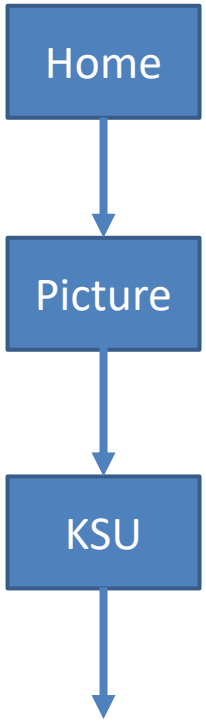




Your image.jpg

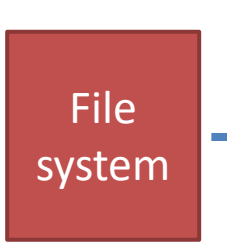
What is a File System?

What you see…

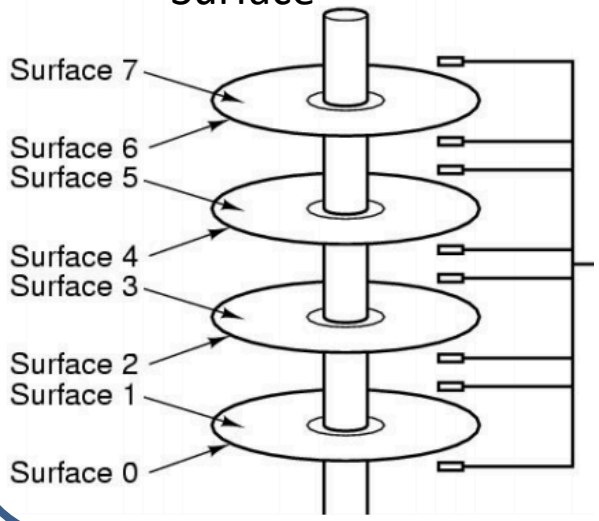


Your image.jpg

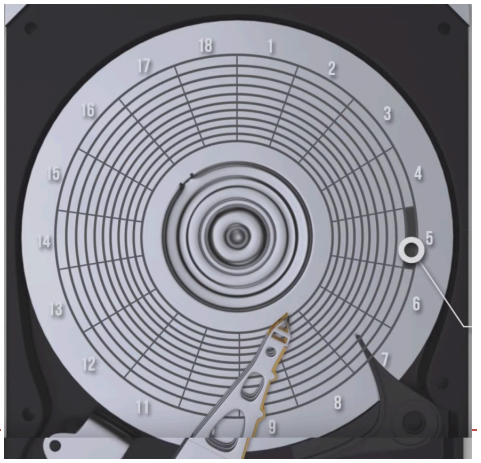
Reality



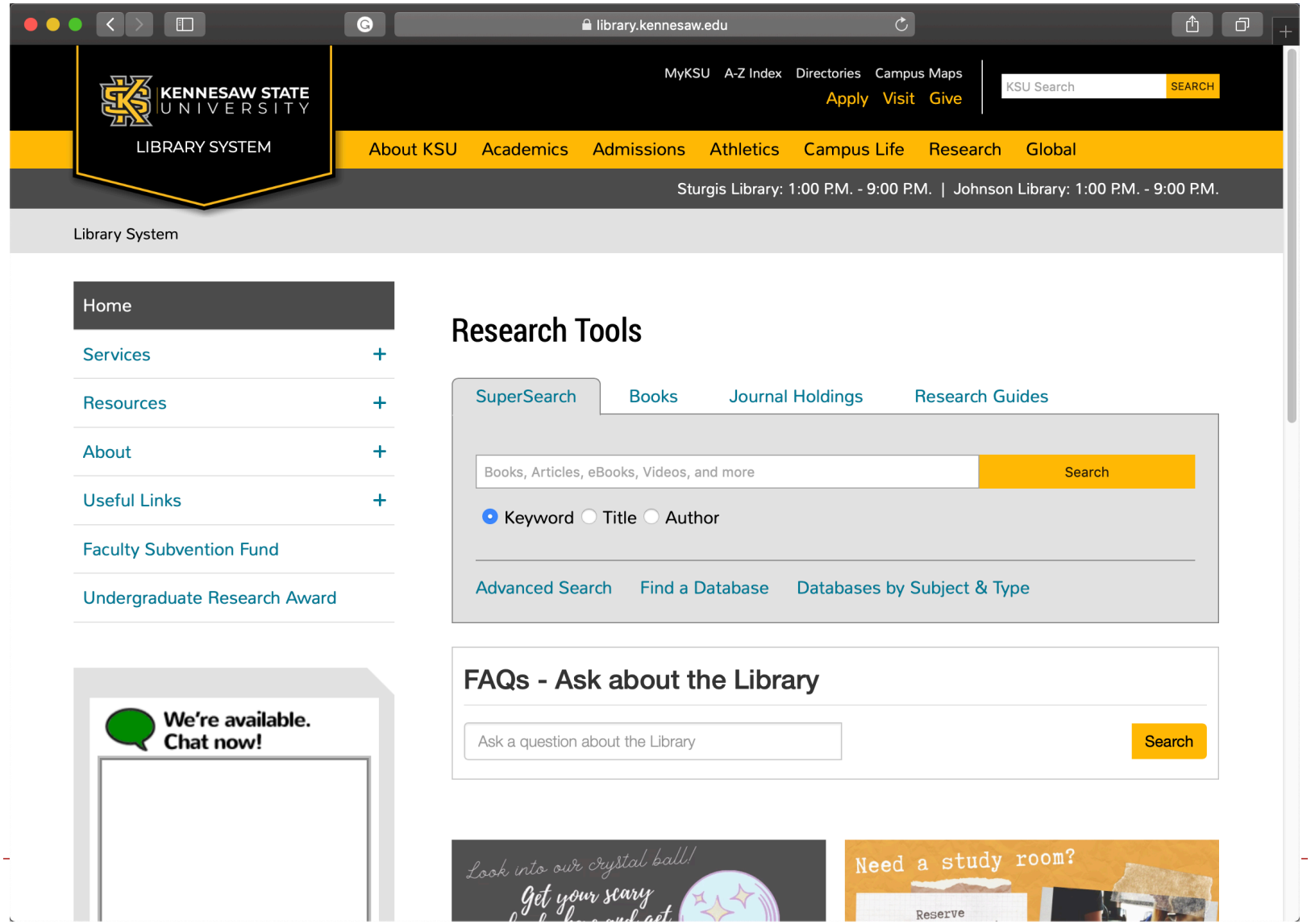
Surface



Track and sector



Library system: manages how and where book is stored, managed and accessed



What is a File System?

- File system is the OS component that manages how and where data is stored, managed and accessed on the raw storage.

- Data in File system:
  - Raw data, by itself, is just a meaningless sequence of bits and bytes.
  - Metadata is the information that describes the data.
    - Also called attributes.
    - Without meta-data, data itself will be incomprehensible.

Raw Data and Metadata

- Metadata is the data about data
- Metadata is usually structured data
  - Metadata of books such as title, version, published data, related descriptions, language, review, etc.
- Metadata is extracted from information resources to describe its characteristics and content,
- Meta is used for organization, description, preservation, management, search, etc.

KENNESAW STATE UNIVERSITY

Searching KSU SuperSearch

Keyword | world history | Search

Basic Search | Advanced Search | Search History

Ken

Detailed Record

PlumX Metrics

See in GIL-Find

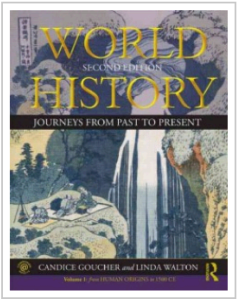
Related Information

- Other Formats and Editions
- Similar Books
- Other Books by this Author

Find Similar Results
using SmartText Searching.

Result List | Refine Search | 1 of 42,152,256

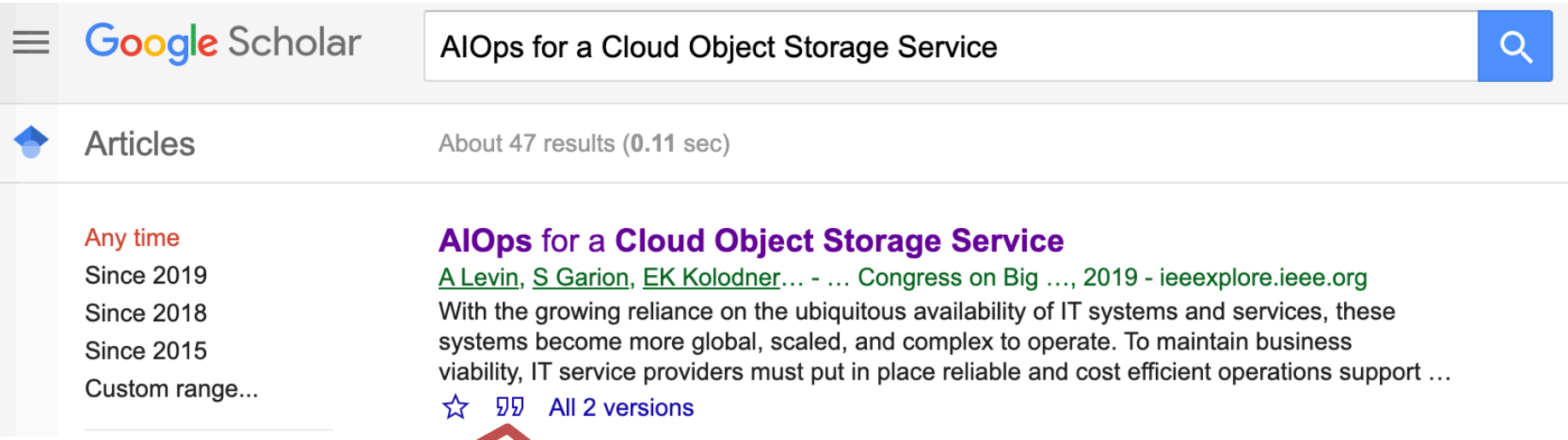
World history [electronic resource] Volume 1, From human origins to 1600 CE journeys from past to present / Candice Goucher and Linda Walton.



Language:	English
Authors:	Goucher, Candice
Publication Information:	Oxfordshire, England ; New York : Routledge, c2013.
Edition:	2nd ed.
Publication Date:	2013
Physical Description:	1 online resource (2 v. (xx, 736 p.)) ill. (chiefly col.), col. maps
Publication Type:	Book; eBook
ISBN:	0-415-67002-0 1-135-08821-7 0-203-06972-2 1-135-08822-5
OCLC:	857067534
Online Access:	Online Access
Accession Number:	ken.9919771319502931
Database:	Kennesaw Alma Catalog

Metadata

Raw Data and Metadata




```
@inproceedings{levin2019aiops,
  title={AIOps for a Cloud Object Storage Service},
  author={Levin, Anna and Garion, Shelly and Kolodner, Elliot K and Lorenz, Dean H and Barabash, Katherine and Kugler, Mike and McShane, Niall},
  booktitle={2019 IEEE International Congress on Big Data (BigDataCongress)},
  pages={165--169},
  year={2019},
  organization={IEEE}
}
```

Raw Data and Metadata

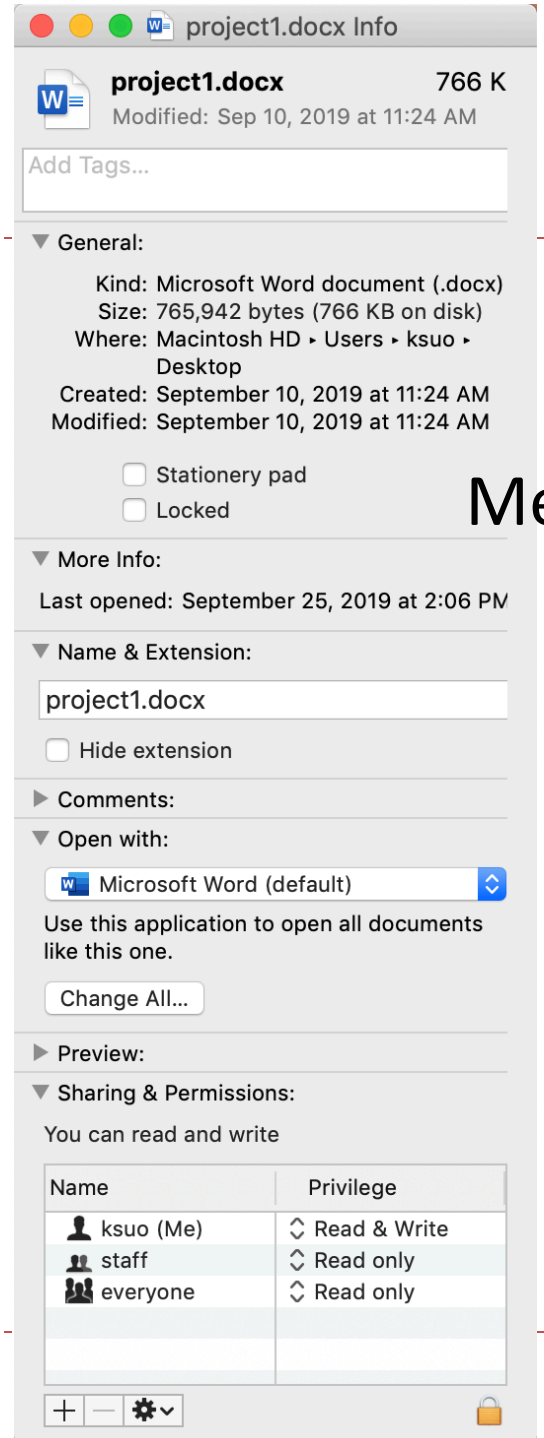
BASIS FOR COMPARISON	DATA	METADATA
Basic	Data is a set of facts and statistics can that be operated, referred or analyzed.	Metadata describes relevant information about the data.
Information	Data can or can not be informative (e.g., raw data like 0101).	Metadata is always informative.
Structure	Data may or may not have been processed or structured.	Metadata is always a processed and structured data.

What is a File System?

Data

project1.docx

Metadata



What is a File System?

- A File System defines
  - Format of the data objects and the format/meaning of meta-data associated with each data object. (store)
    - E.g. File name, permissions, and size of a file.
  - Location of the individual data blocks for each data object. (access)
  - A framework to manage free space on the raw storage. (manage)

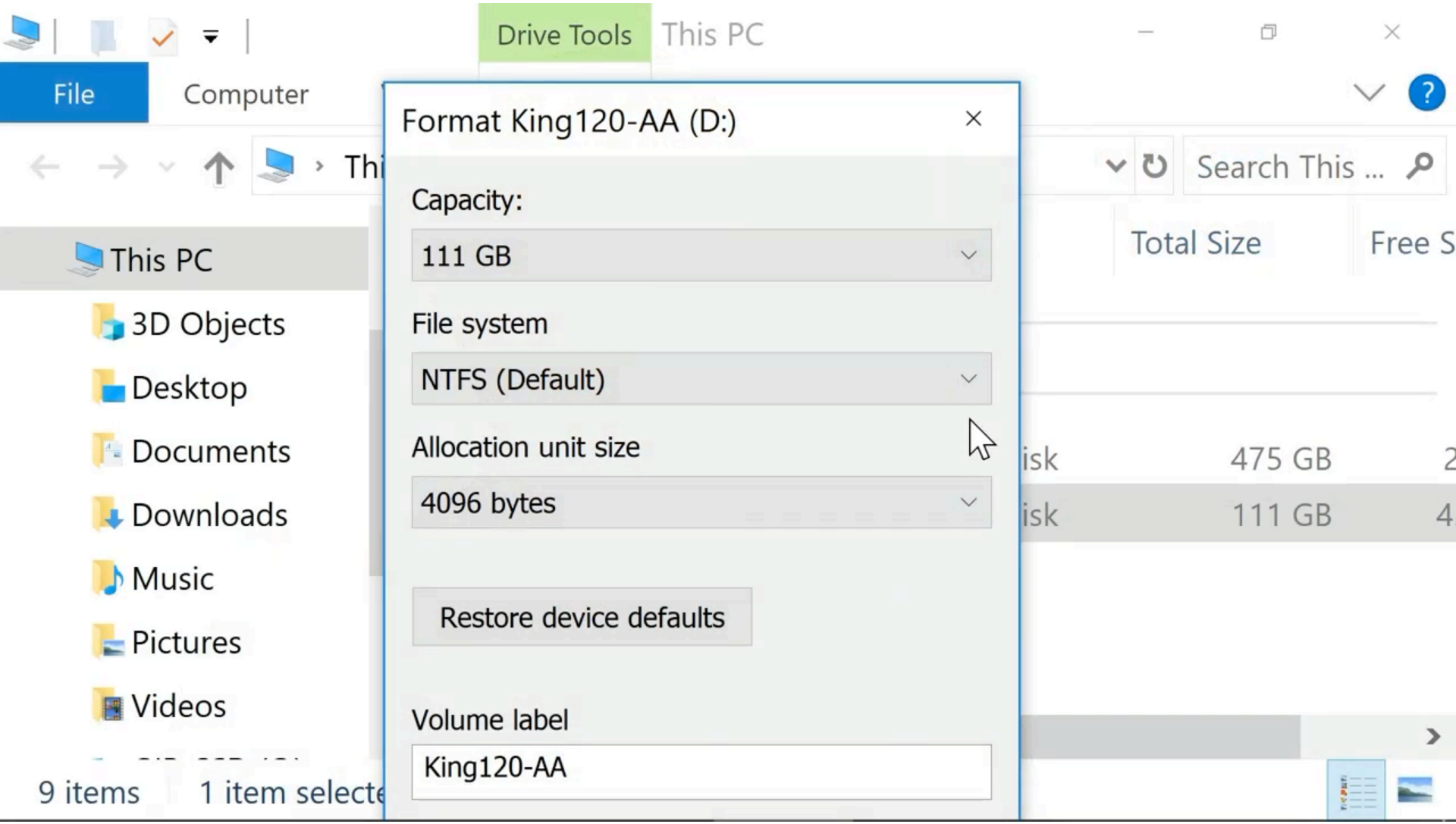
Common file system: ext3, ext4, btrfs, etc.

- Linux
  - ~~Ext: extended file system (1992)~~
  - ~~Ext2: second extended file system (1993)~~
  - Ext3: third extended filesystem (2001)
  - Ext4: fourth extended filesystem (2008)
  - Btrfs: file system based on the copy-on-write (2009)

Not used

- Windows
  - FAT32, exFAT
  - Ntfs: New technology file system developed by Microsoft (1993)
- Apple
  - HFS: Hierarchical File System (1985)
  - Apfs: Apple file system for (2017)

Common file system: ext3, ext4, btrfs, etc.

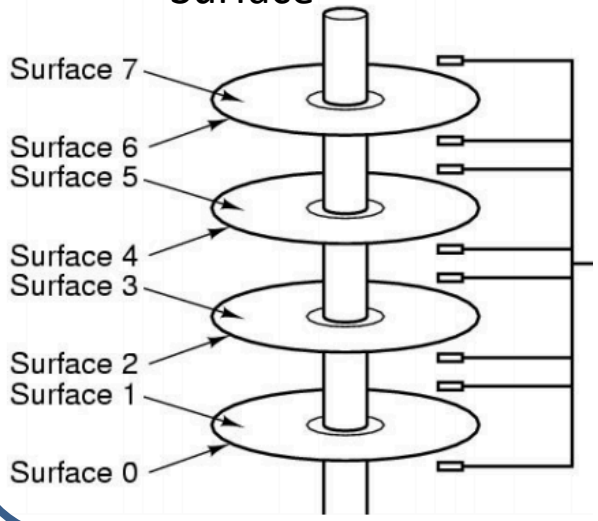


https://www.youtube.com/watch?v=_h30HBYxtws&t=52s

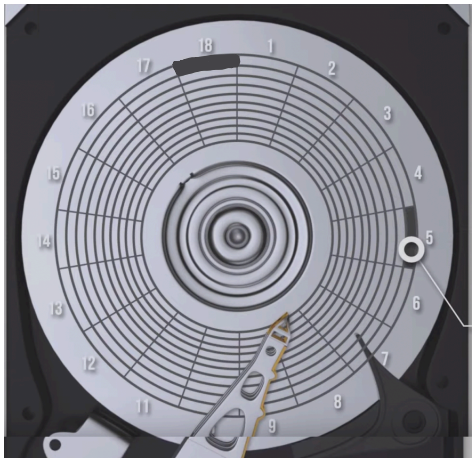
Unit of File System: Blocks

- Files and directories are allocated in blocks
- Block size is usually a multiple of the sector size

Surface

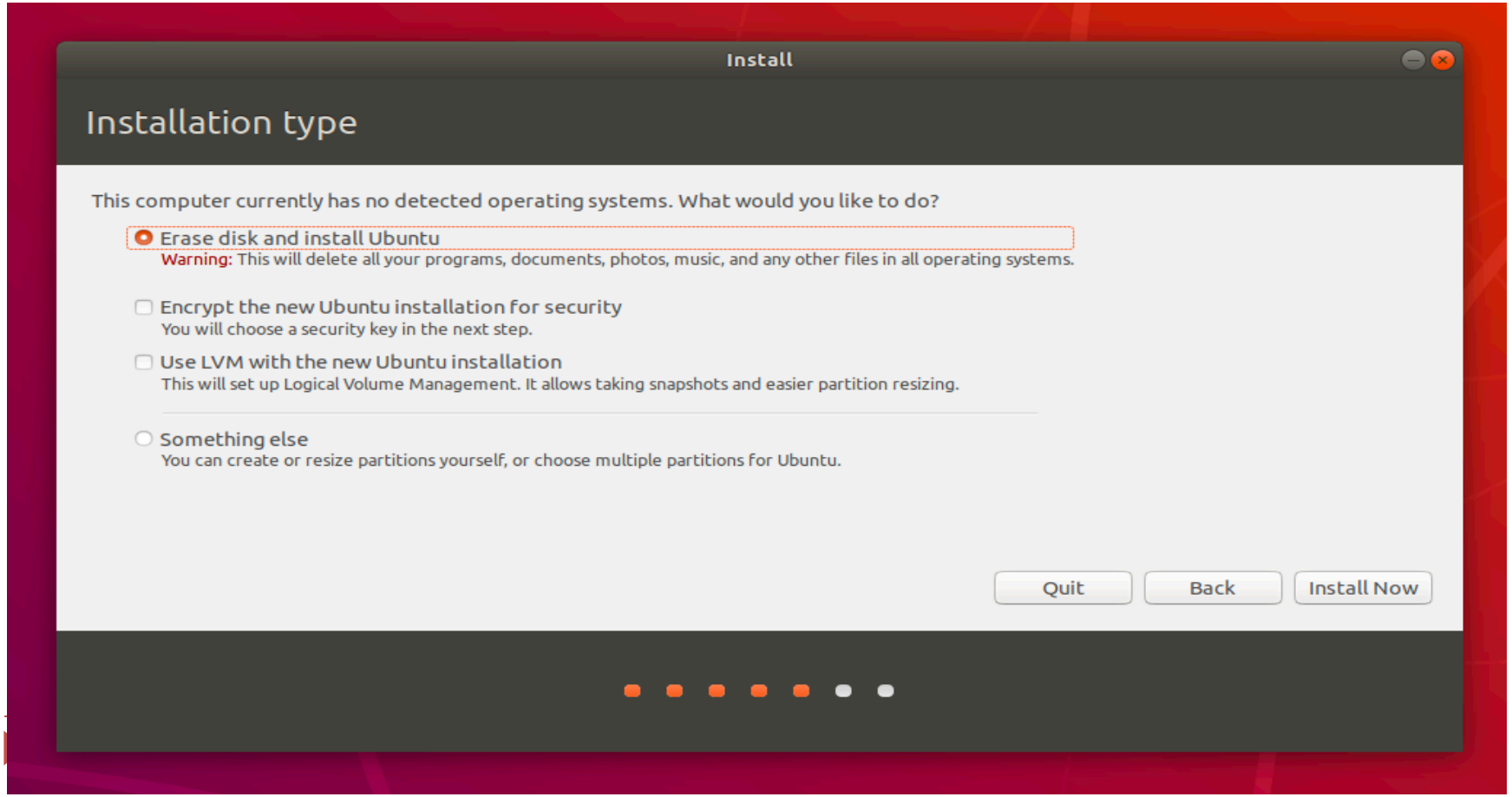


Track and sector



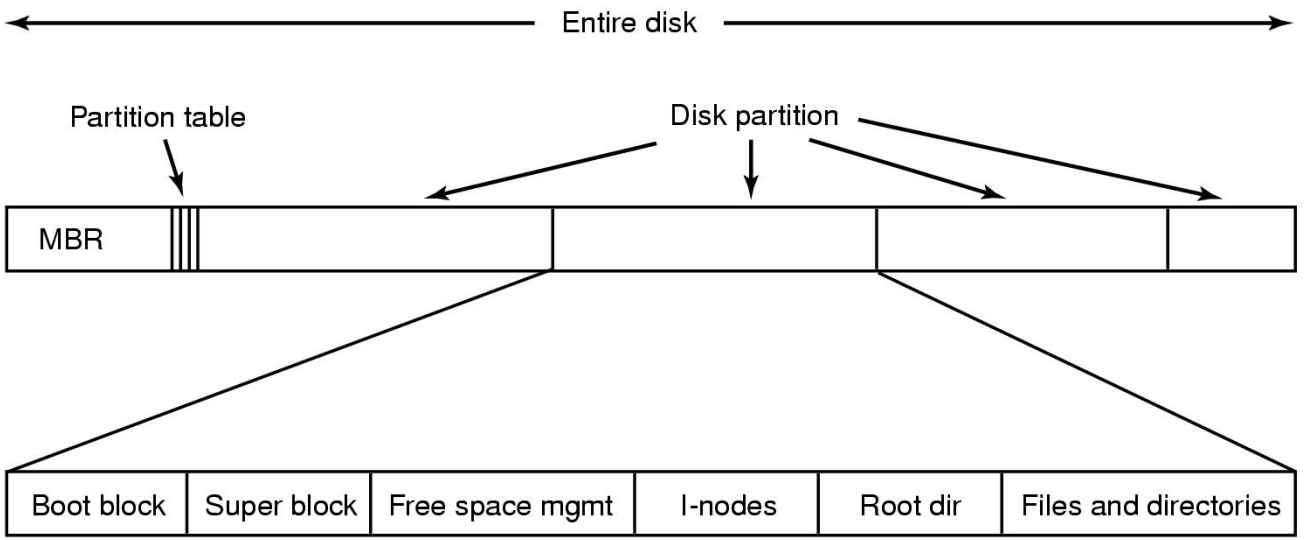
What will happen to the disk when you have file system?

Free space



File System layout

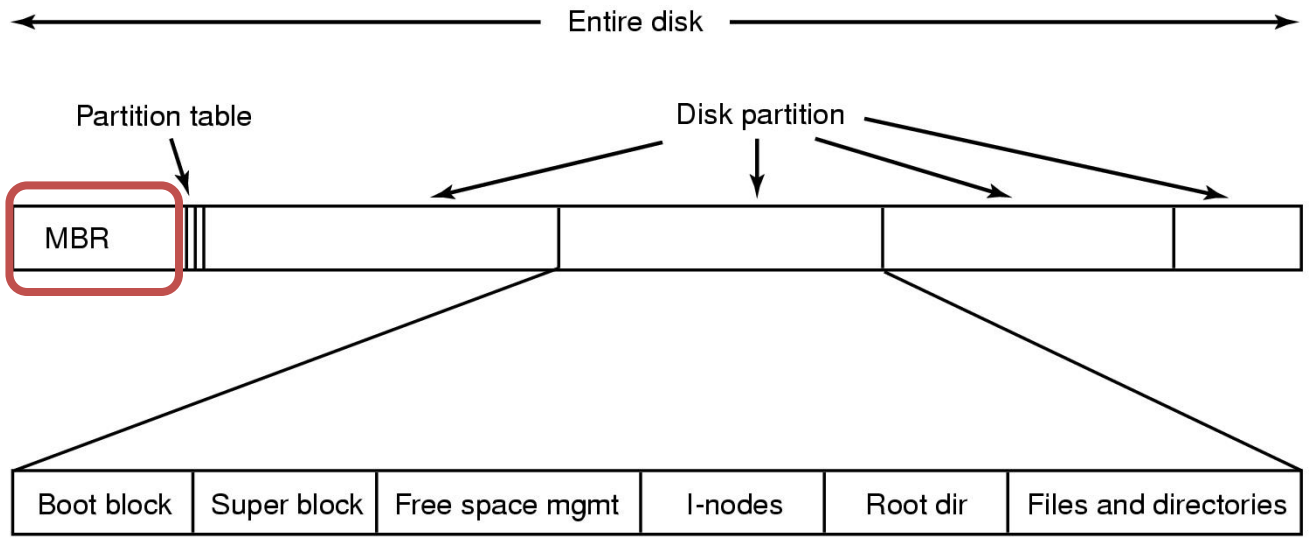
- Most disks can be divided into one or more partitions



A possible file system layout

File system layout

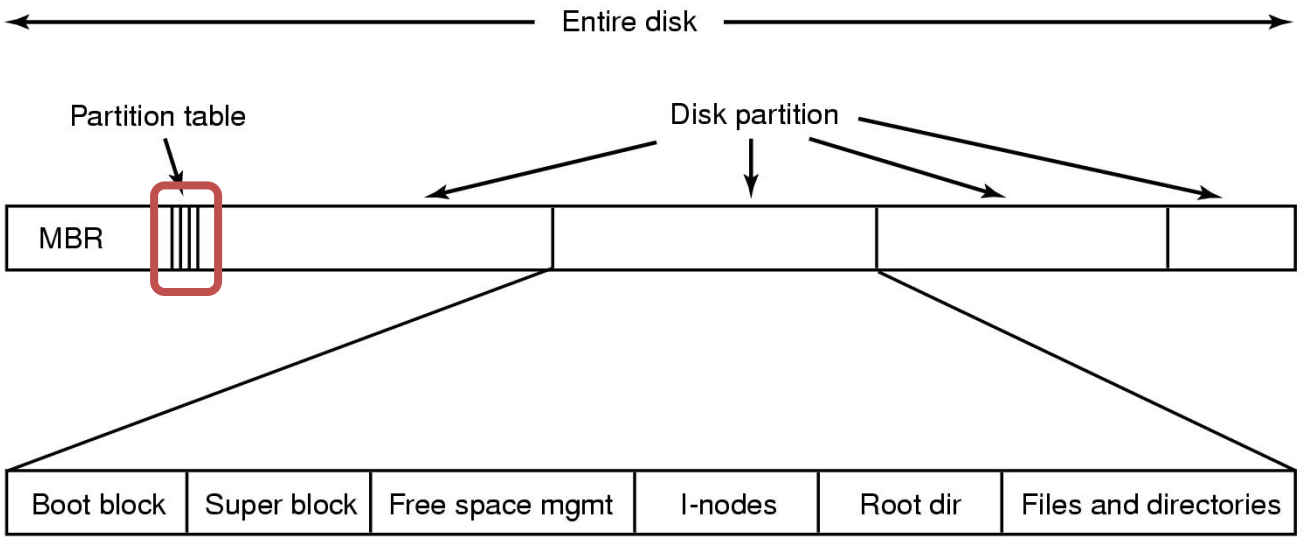
- *Master Boot Record* is used to boot the computer



A possible file system layout

File system layout

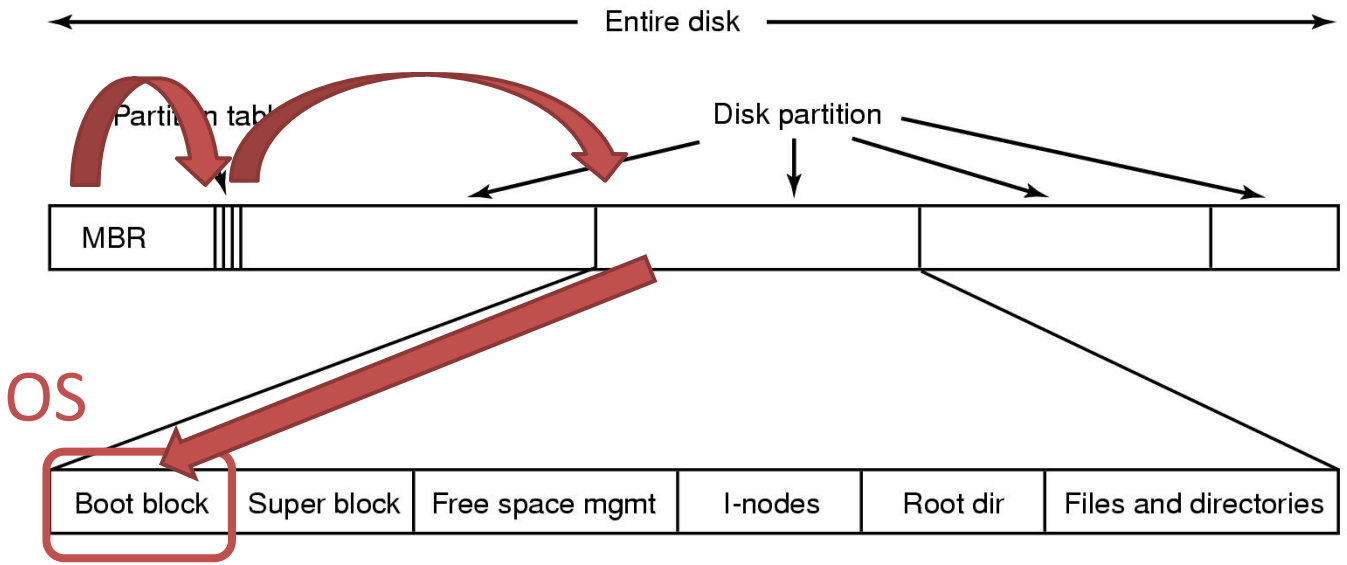
- *Partition table* is present at the end of MBR.
- This table gives the starting and ending addresses of each partition.



A possible file system layout

File system layout

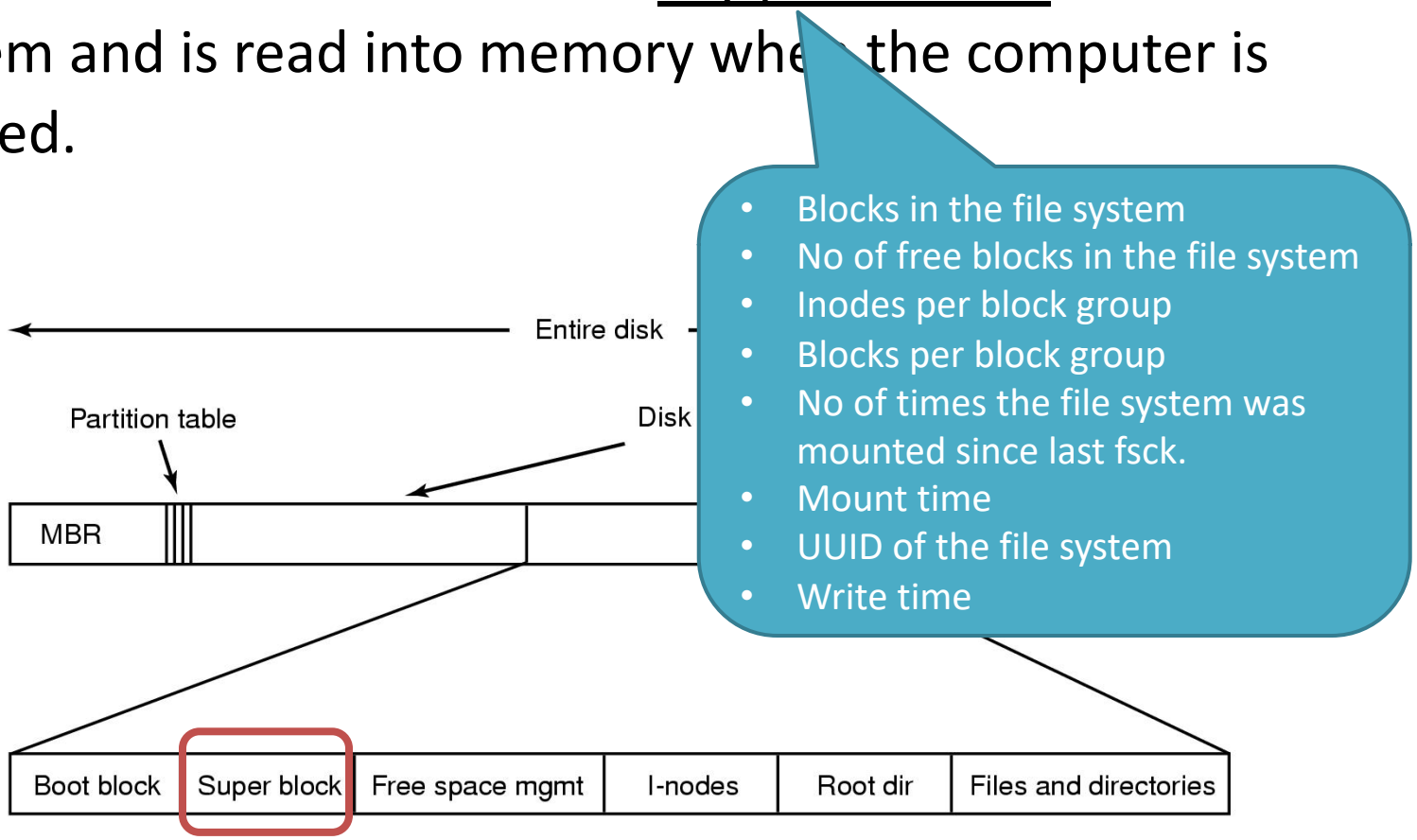
- ***Boot Block***: When the computer is booted, the BIOS reads in and executes the MBR.
- The first thing the MBR program does is locate the active partition, read in its first block, which is called the ***boot block***, and execute it.
- The program in the boot block loads the operating system contained in that partition.



A possible file system layout

File system layout

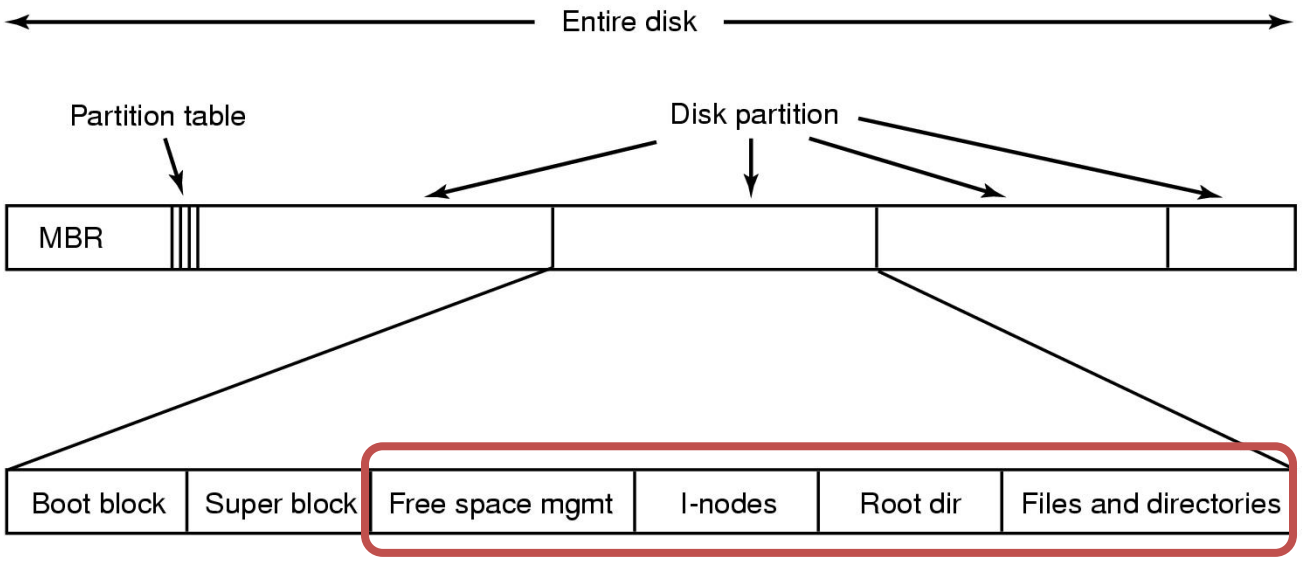
- *Super Block*: It contains all the key parameters about the file system and is read into memory when the computer is booted.



A possible file system layout

File system layout

- *Rest*: free space management, files and directories



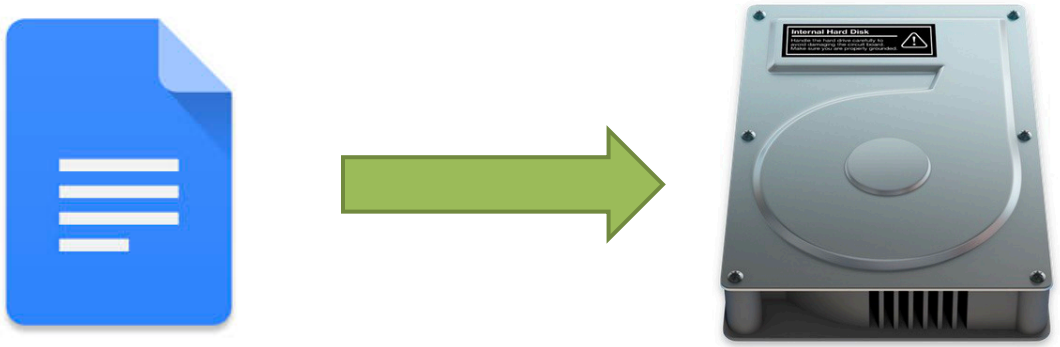
A possible file system layout

Outline

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system

Implementing Files on Disks

- Allocation: how to store the files on the disk?
  - Contiguous allocation
  - Linked list allocation
  - File allocation table (FAT)
  - Index allocation

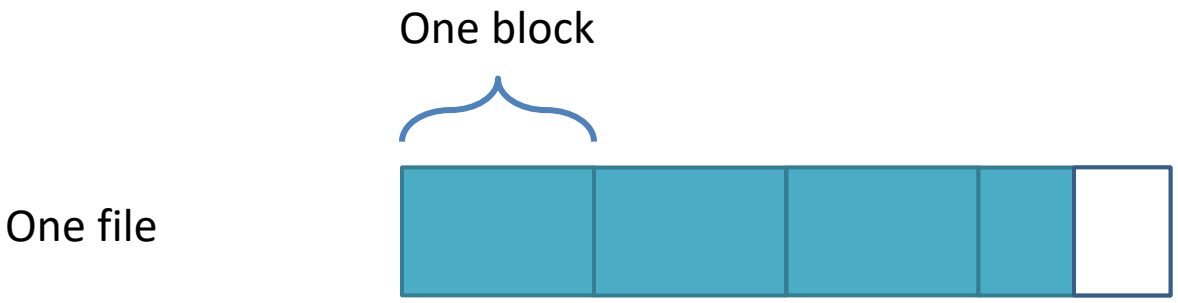


Implementing Files (1) – Contiguous Allocation

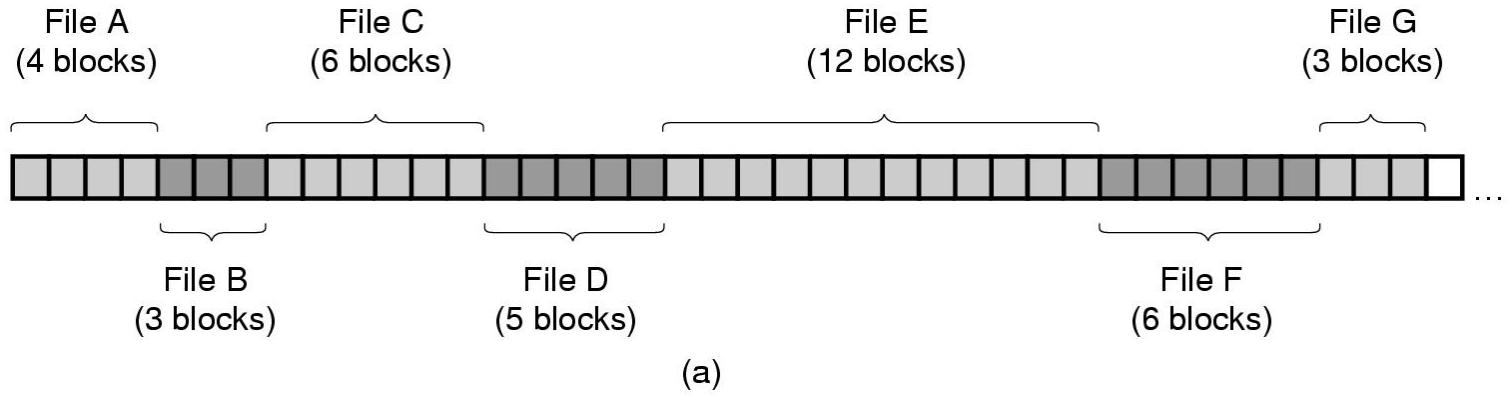
- **Contiguous Allocation**: each file is stored as a contiguous run of disk blocks.

- **Example:** On a disk with 1KB blocks, a 50KB file would be allocated 50 consecutive blocks. With 2KB blocks it would be 25 consecutive blocks.

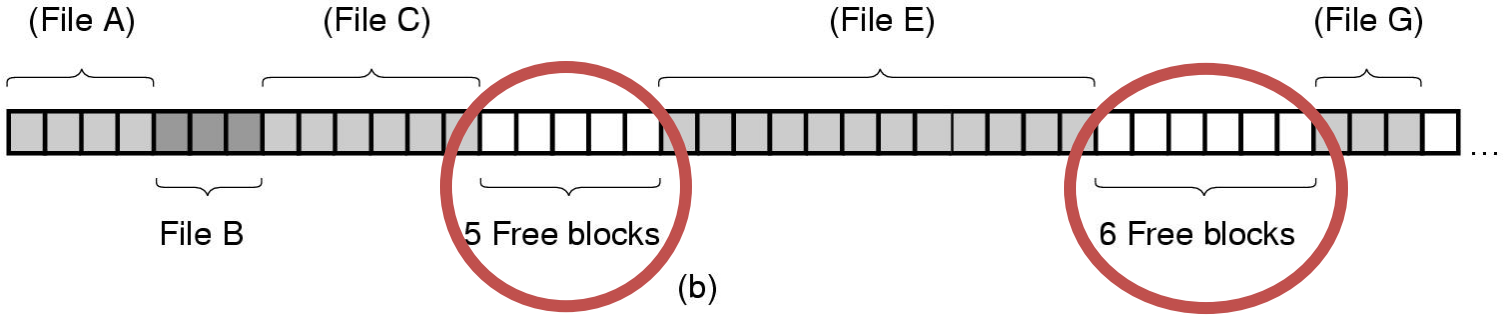
- **Problem:** Each file begins at the start of a new block, so that if file A is occupying 3½ blocks, some space is wasted at the end of the last block.



Implementing Files (1) – Contiguous Allocation



(a) Contiguous *block allocation* of disk space for 7 files

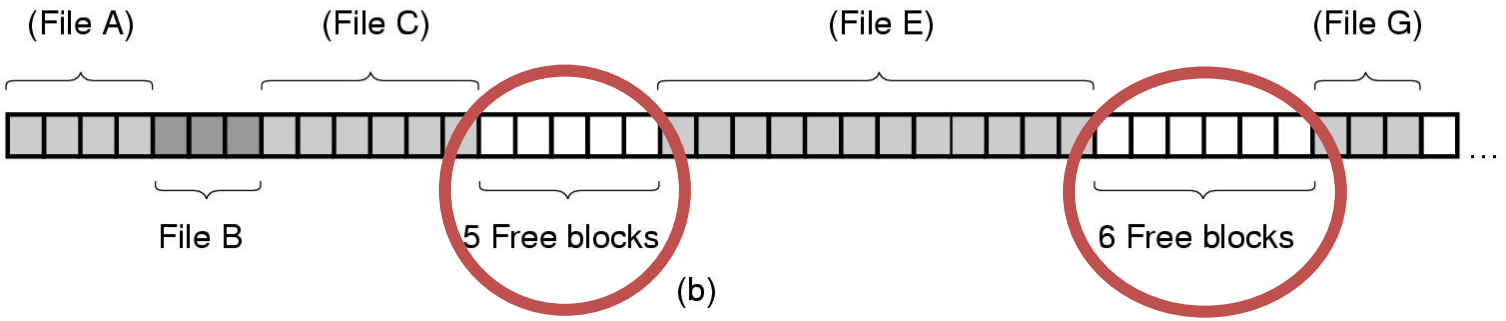


(b) State of the disk after files *D* and *F* have been dynamically removed

Implementing Files (1) – Contiguous Allocation

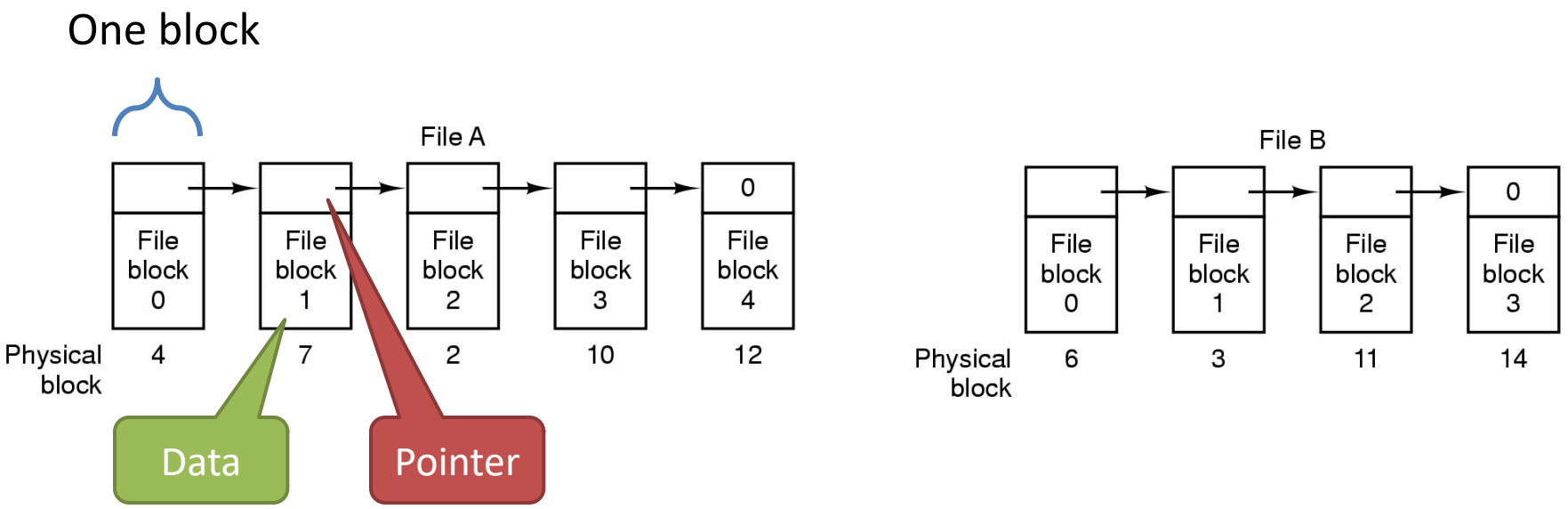
- **Advantages:**
  - Simple to implement.
  - The read performance is excellent because the entire file can be read from the disk in a single operation.

- **Drawbacks:**
  - Over the course of time the disk becomes **fragmented**.



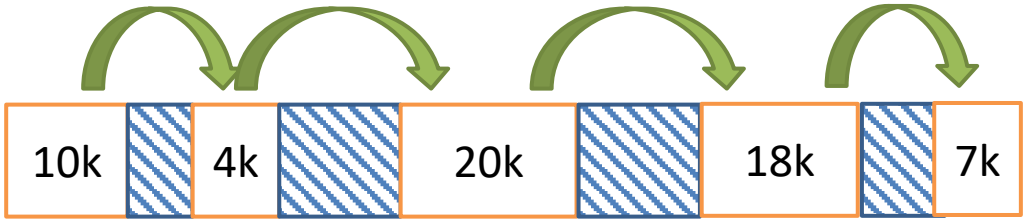
Implementing Files (2) – Linked List Allocation

- Keep each file as a linked list of disk blocks
  - The first word of each block is used as a pointer to the next one.
  - The rest of the block is for data.

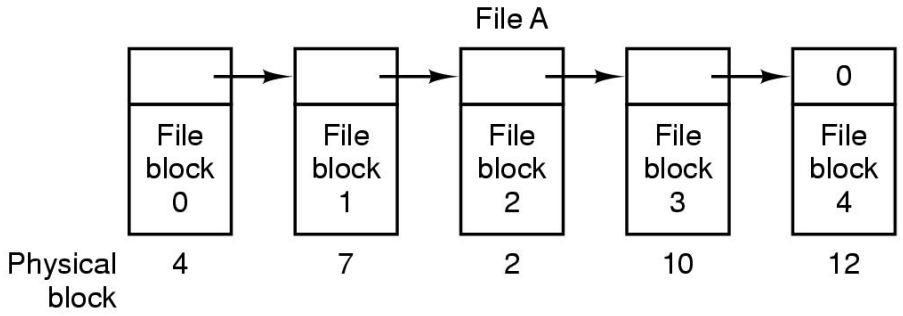


Implementing Files (2) – Linked List Allocation

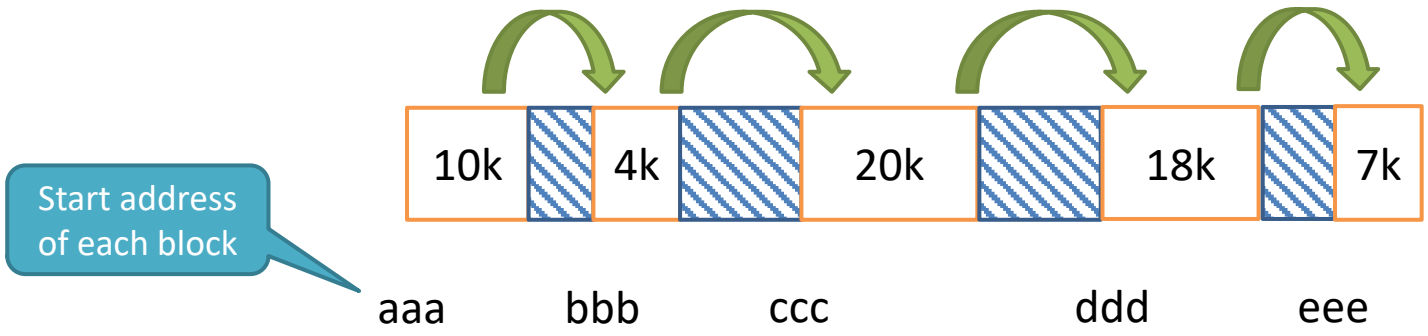
- Pros:
  - no space is lost due to disk fragmentation



- Cons:
  - random access of a file is very slow (because every time it has search from the start)



Implementing Files (3) – FAT (File Allocation Table)



Memory

Block ID	address
1	aaa
2	bbb
3	ccc
4	ddd
5	eee

- FAT (File allocation table): a table in memory with the pointer word of each disk block

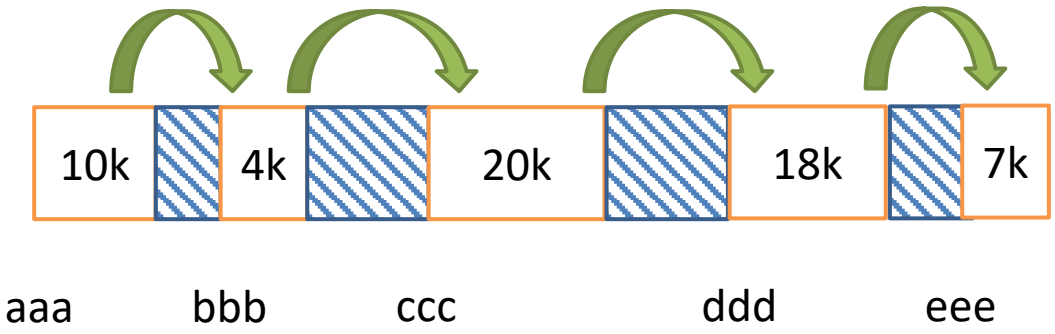
Implementing Files (3) – FAT (File Allocation Table)

- **Advantages:**
  - Using FAT random access can be made much easier.

- **Drawbacks:**
  - the entire table must be in memory all the time to make it work.

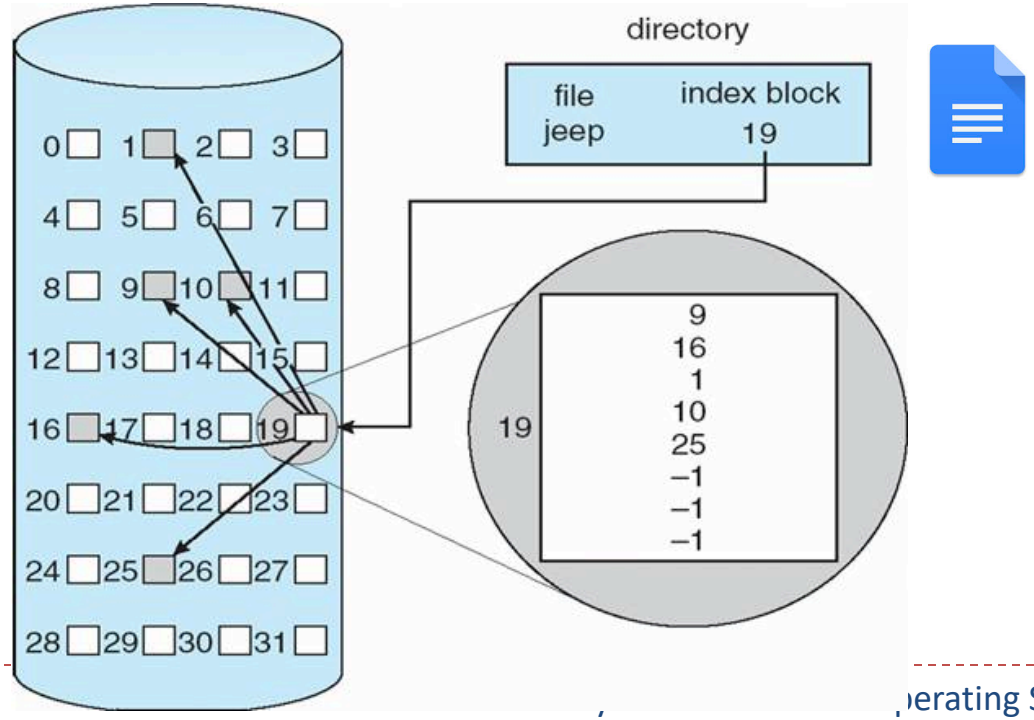
Memory

Block ID	address
1	aaa
2	bbb
3	ccc
4	ddd
5	eee

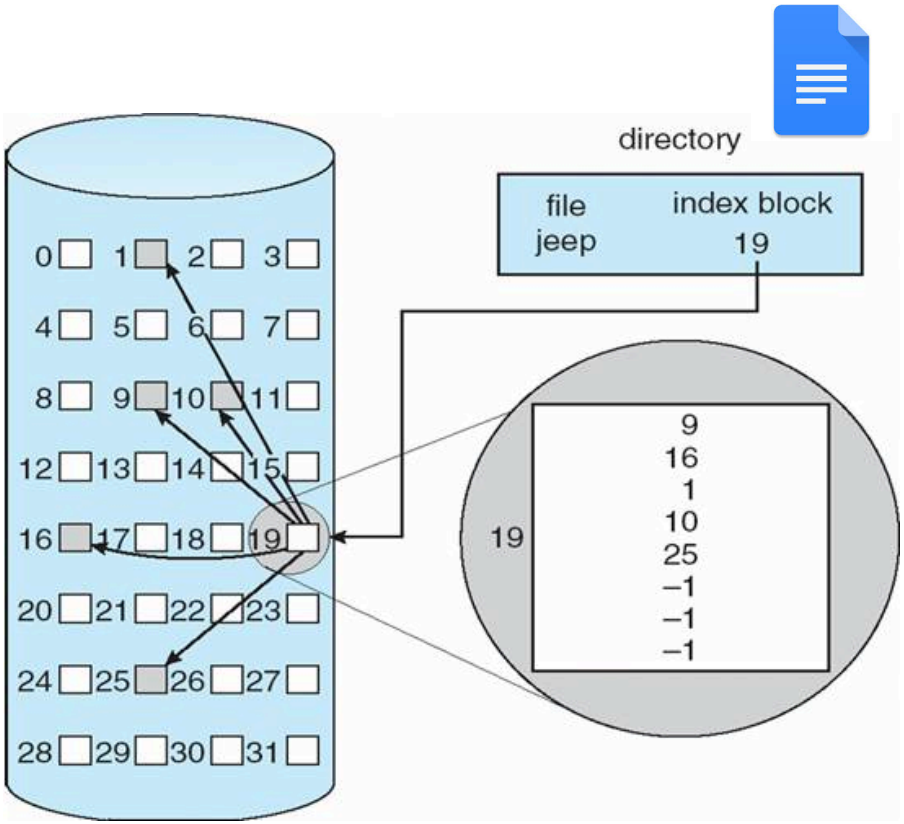


Implementing Files (4) – Indexed Allocation

- Each file has its own ***index block*** on disk
- An index block grouping all pointers together (e.g., node 19)
- It contains an array of disk address



Implementing Files (4) – Indexed Allocation

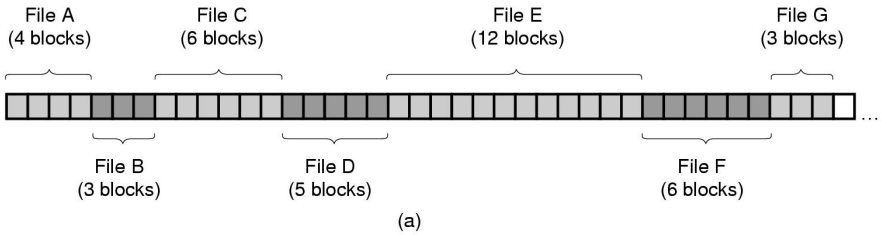


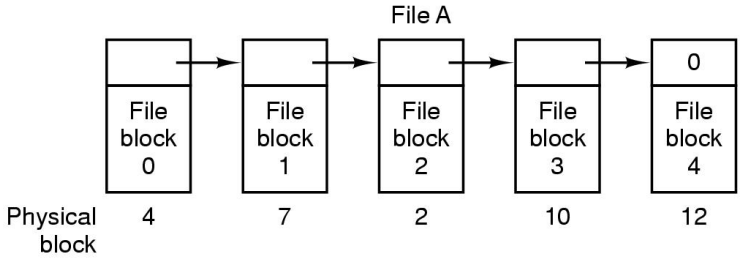
i-node: a data structure listing the attributes and disk addresses of the file's blocks

Unlike the FAT, the i-node is **in memory** only when the corresponding file is **open**.

Implementing Files – Summary

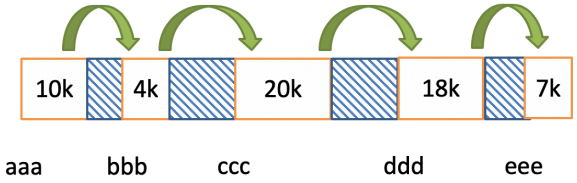
- How to store file on disk?
  - Contiguous allocation
  - Linked list allocation
  - File allocation table (FAT)
  - Index allocation

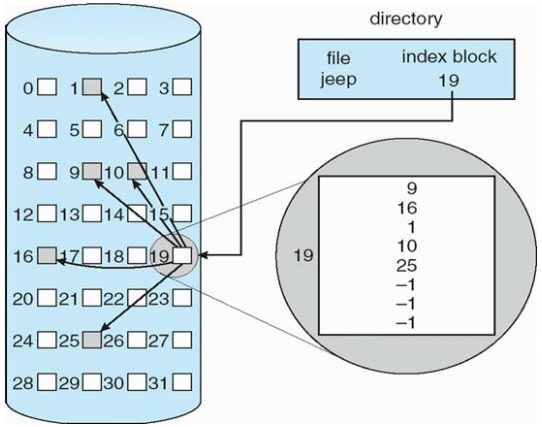




Memory

Block ID	address
1	aaa
2	bbb
3	ccc
4	ddd
5	eee

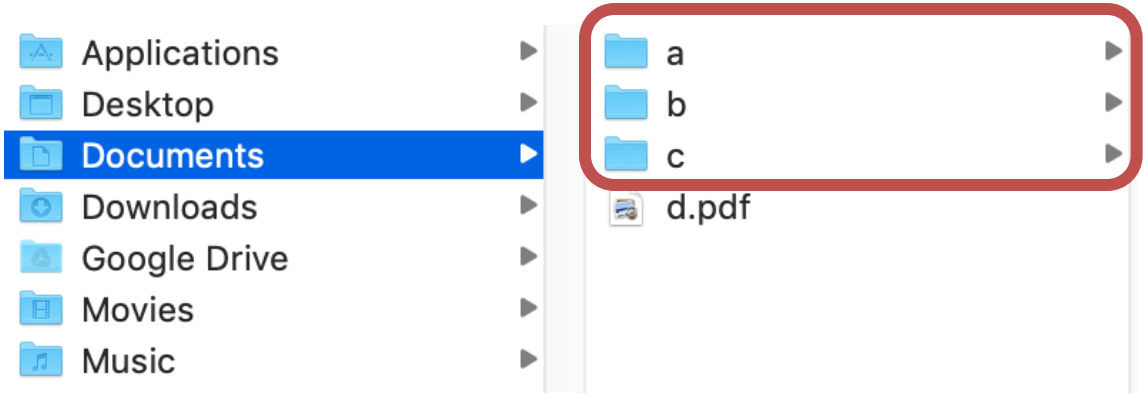




Outline

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system

Implementing Directories



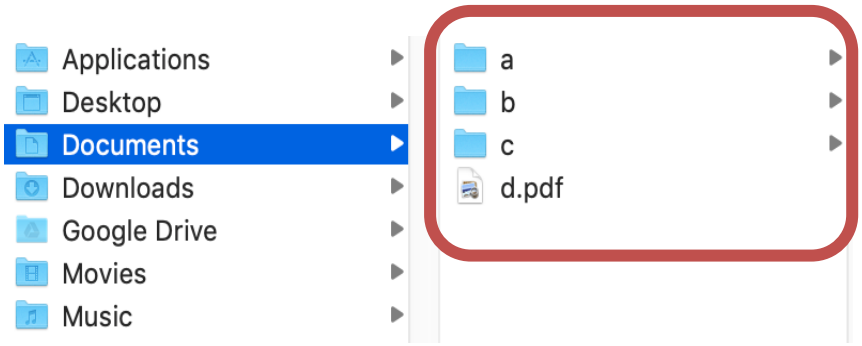

```
ksuo@ltksup50143mac ~/Documents> ls -al
total 104
drwx------+  8 ksuo  staff    256 Nov  5 16:20 .
drwxr-xr-x+ 32 ksuo  staff   1024 Nov  5 16:17 ..
-rw-r--r--@  1 ksuo  staff   6148 Nov  5 16:20 .DS_Store
-rw-------   1 ksuo  staff      0 Aug  5 09:21 .localized
drwxr-xr-x   2 ksuo  staff     64 Nov  5 16:19 a
drwxr-xr-x   2 ksuo  staff     64 Nov  5 16:20 b
drwxr-xr-x   2 ksuo  staff     64 Nov  5 16:20 c
-rw-r--r--@  1 ksuo  staff  42981 Nov  1 15:50 d.pdf
```

- The directory entry is a special file

Implementing Directories Option 1

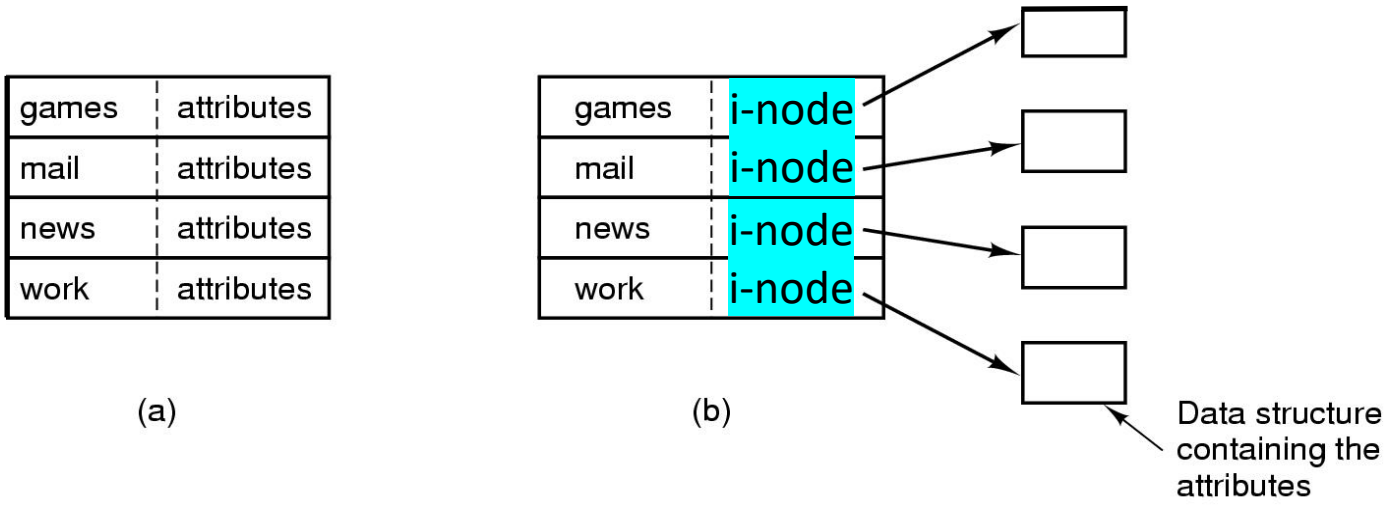
- In a simple design a directory consists of a list of fixed-size entries, one per file, containing
  - a (fixed-length) file name,
  - a structure of the file attributes, and
  - one or more disk addresses

	attribute		
name	type	address	size
a	directory	0x001010	64
b	directory	0x011010	64
c	directory	0x111010	64
d.pdf	pdf file	0x001011	40k



Implementing Directories Option 2

- For systems that use i-nodes, another possibility for storing the attributes is in the i-nodes, rather than in the directory entries



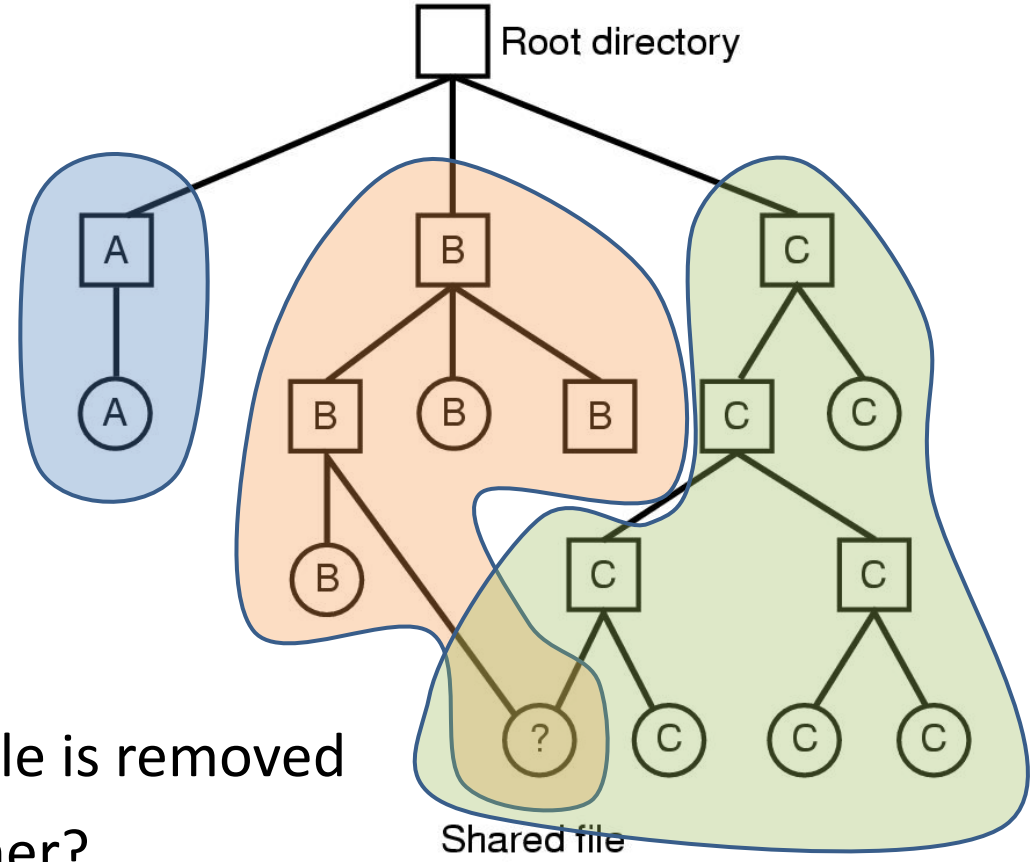
Outline

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system

Shared Files

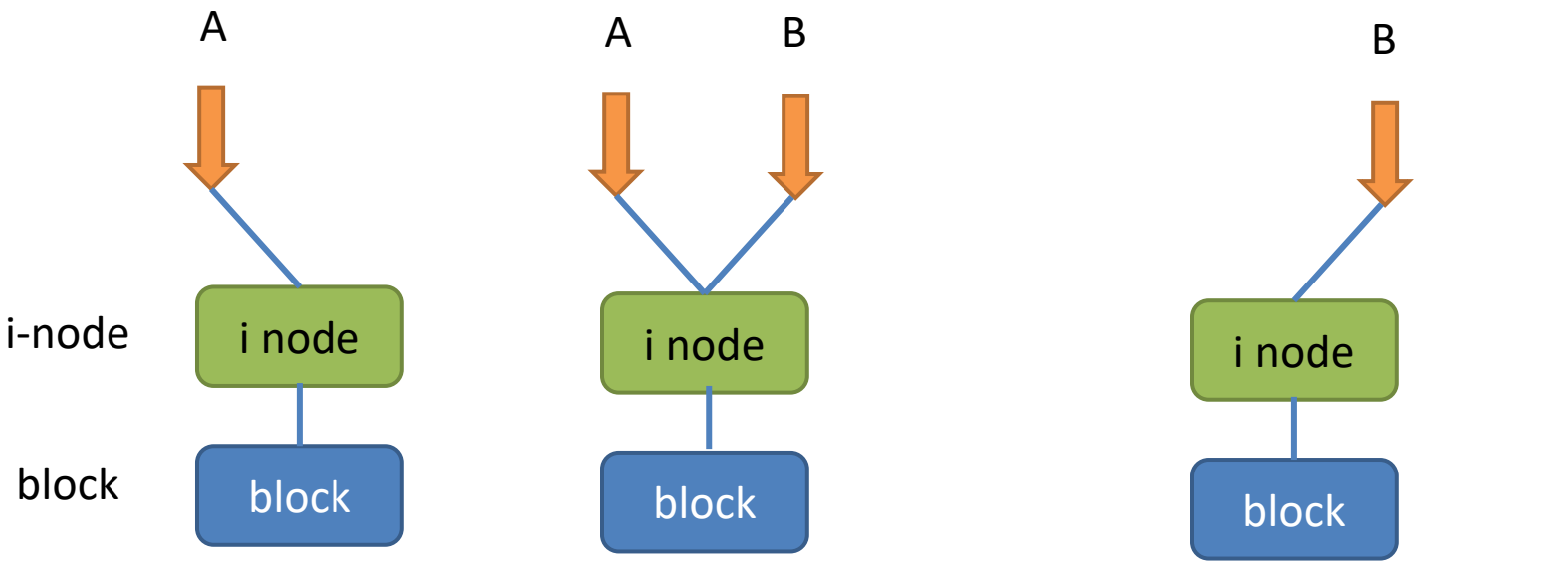
- Shared file is convenient for a file to appear simultaneously in different directories belonging to different users

File system containing a shared file



What if a file is removed by the owner?

Shared Files using i-node Hard Link

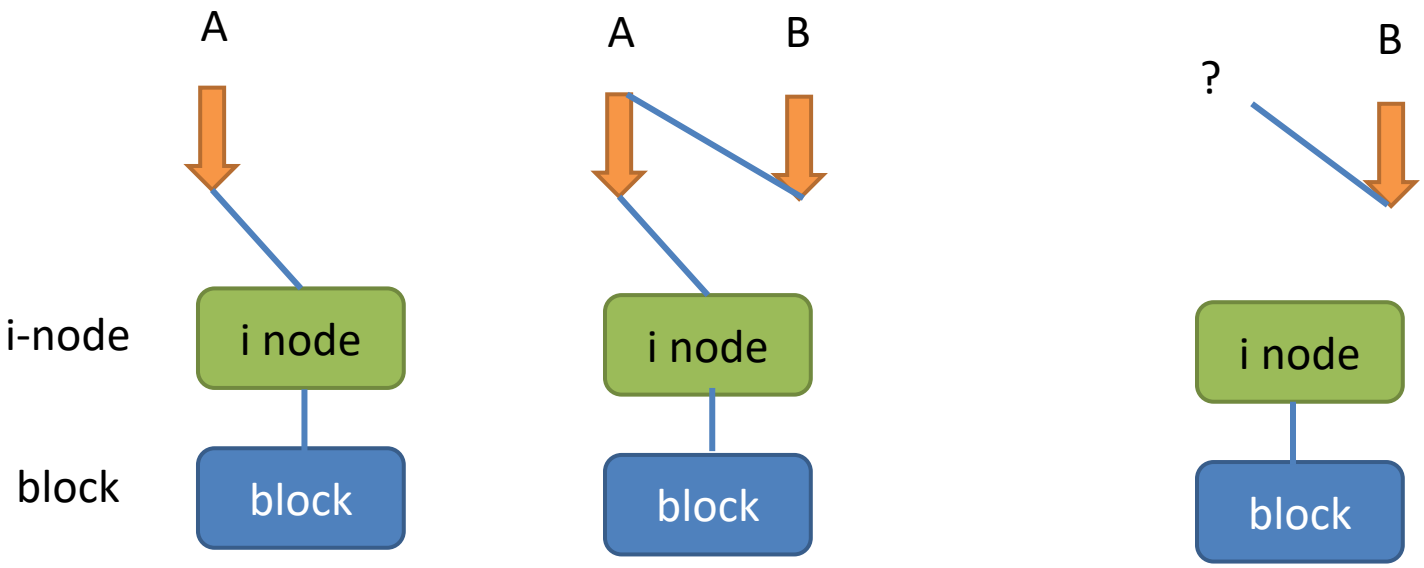


(a) Situation prior to linking;

(b) After the link is created

(c) After the original owner removes the file (the data is still there, just remove the link)

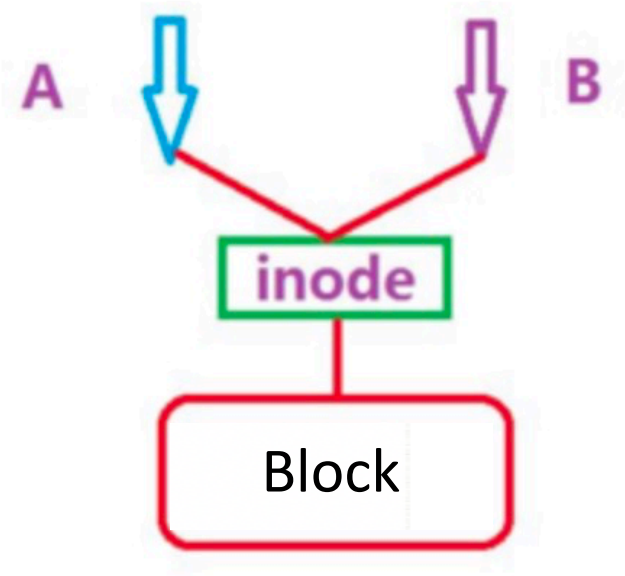
Shared Files – Symbolic Link or Soft Link



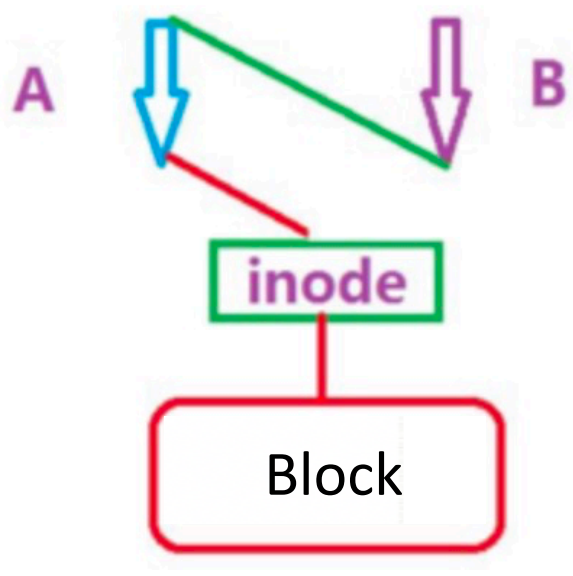
- A symbolic link is a special kind of file that points to another file. However, a symbolic link does not contain the data in the target file.

Difference between hard link and soft link

Hard link



Soft link



Hard link and soft link in Linux

- To create Hard link,
  - ln <source> <linkname>
- To create Symbolic link,
  - ln -s <source> <linkname>

Hard link and soft link comparison

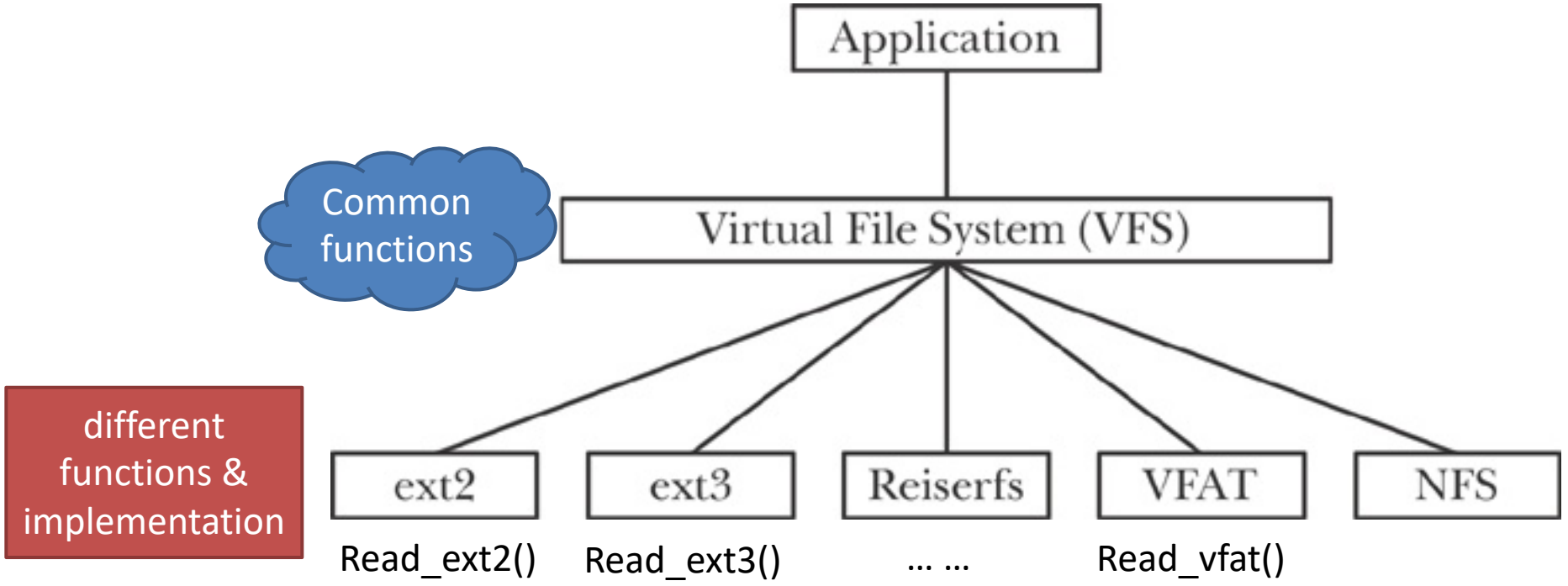
COMPARISON	HARD LINK	SOFT LINK
Definition	Hard link points to the original file	Soft link points to the hard link
I-node	Share the same inode pointing to the same file	Different inode pointing to the same file
Performance	Fast and efficient	Have some overhead due to longer path
Feature	Only link to a file	Link both to files and directory

Outline

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system

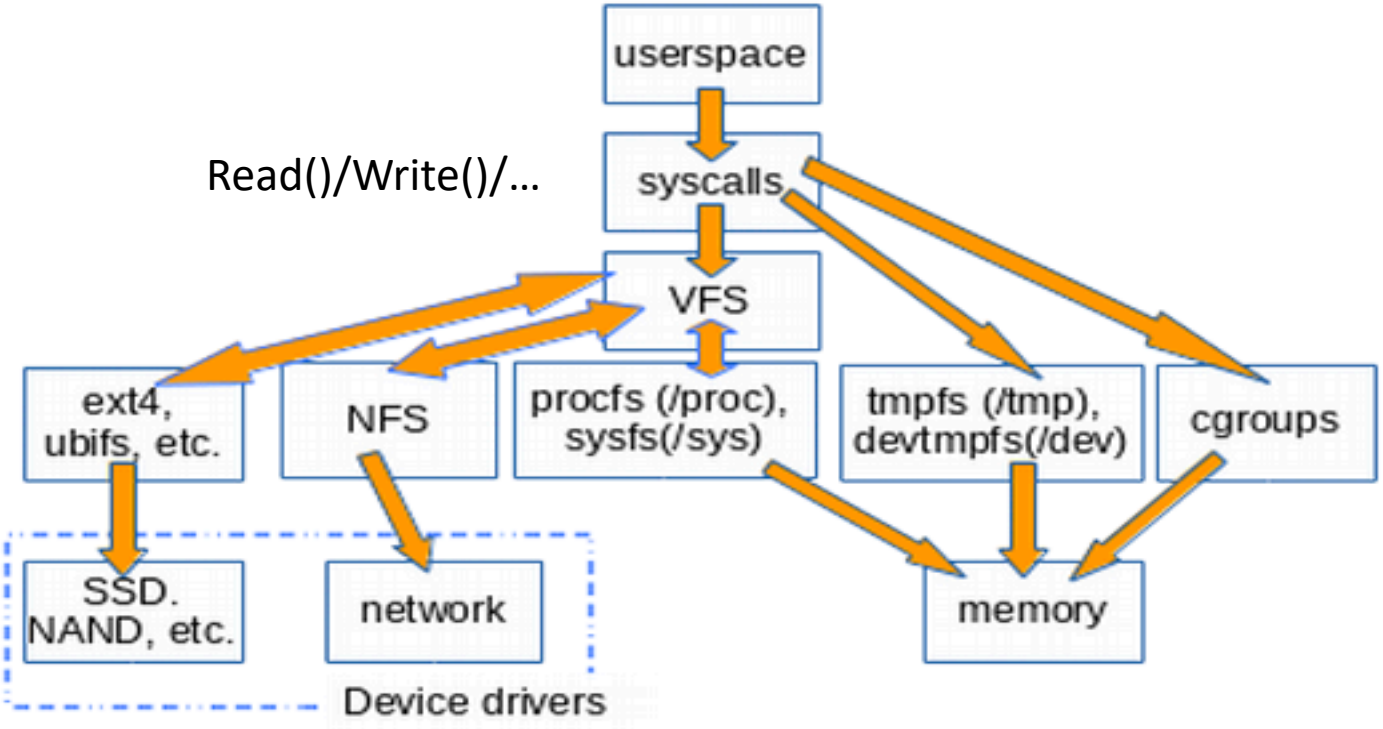
Virtual File System (VFS)

- **A virtual file system** is an abstract layer with common functionalities supported by all the underlying concrete file systems



Virtual File System (VFS)

- The key idea is to abstract out that part of the file system that is **common to all file systems** and put that code in a separate layer that calls the underlying concrete file system to actually manage the data. The apps interact with the VFS.



Conclusion

- What is File System?
  - The OS component that organizes data on raw storage device
- File implementation
  - Contiguous allocation, Linked list allocation, File allocation table (FAT), Index allocation
- Directory implementation
  - Fixed entry, i-node index
- Shared Files
  - Hard link vs soft link
- Virtual file system