OPERATING SYSTEMS

I/O Management, System Protection and Security and Case Study

Kakoli Bora

Department of Computer Science

OPERATING SYSTEMS

Case Study – Windows File System

Kakoli Bora

Department of Computer Science

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau
5. Internet source

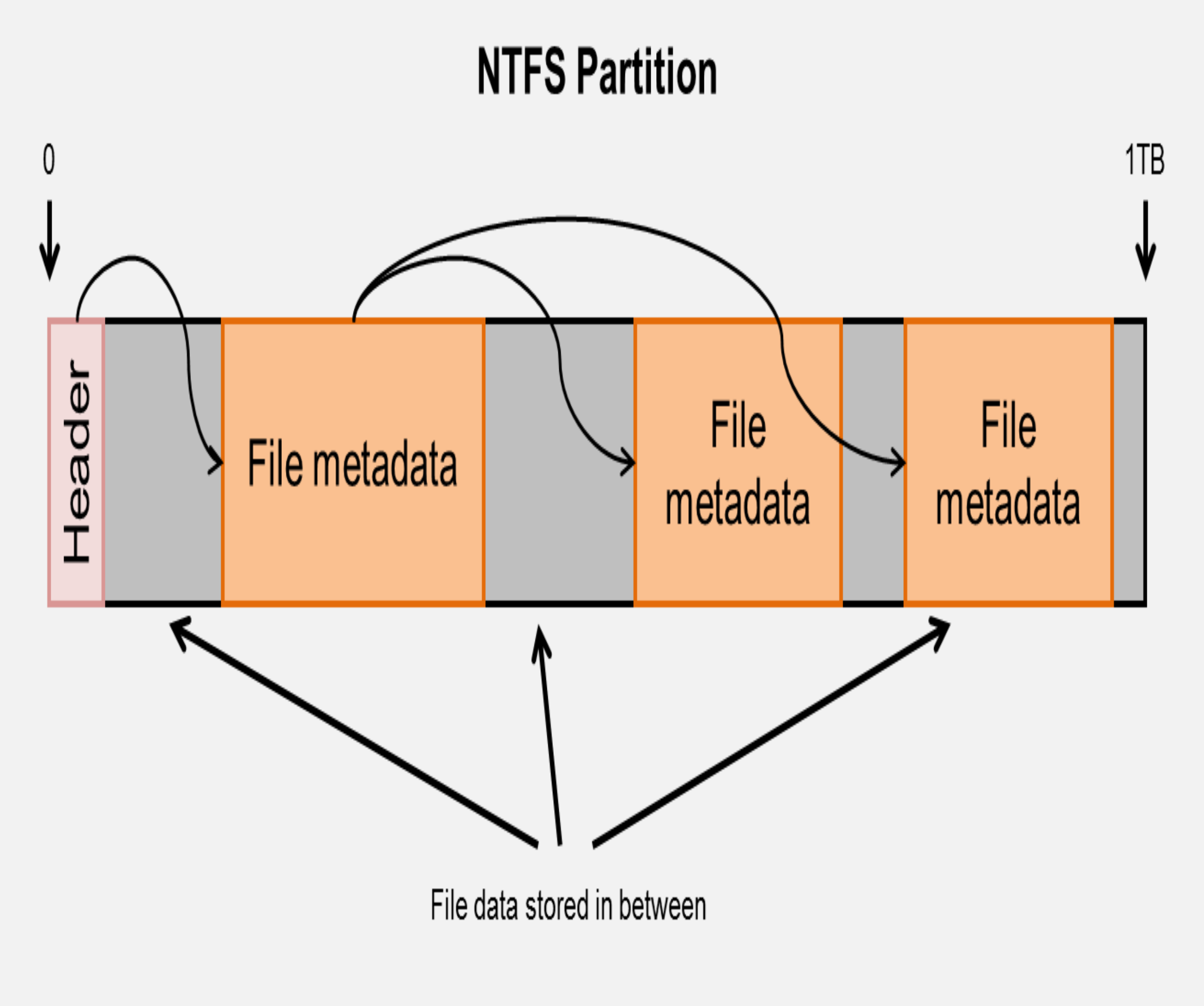
File System - Basics

- ❖FAT file system is still used for portability on other systems such as cameras, flash memory and external disks
- ❖FAT file system does not restrict file access to authorized users.
- ❖NTFS uses ACLs to control access to individual files and supports encryption.
- ❖NTFS supports data recovery, fault tolerance, very large files and file systems, journaling, file compression, etc
- ❖A file in NTFS is not a simple byte stream, as in MS-DOS or UNIX, rather, it is a structured object consisting of attributes
- ❖Attributes like file name, creation time, descriptor, ACLs, etc

File System – Basics (Cont.)

- The fundamental structure of Windows file system (NTFS) is a *volume*
  - Created by the disk administrator utility
  - Based on a logical disk partition
  - May occupy a portions of a disk, an entire disk, or span across several disks
- All *metadata*, such as information about the volume, is stored in a regular file
- NTFS uses ***clusters*** as the underlying unit of disk allocation
  - A cluster is a number of disk sectors that is a power of two
  - The default cluster size is based on the volume size - 4 KB for volumes > 2 GB
  - Because the cluster size is smaller than for the 16-bit FAT file system, the amount of internal fragmentation is reduced

File System — NTFS (Cont.)



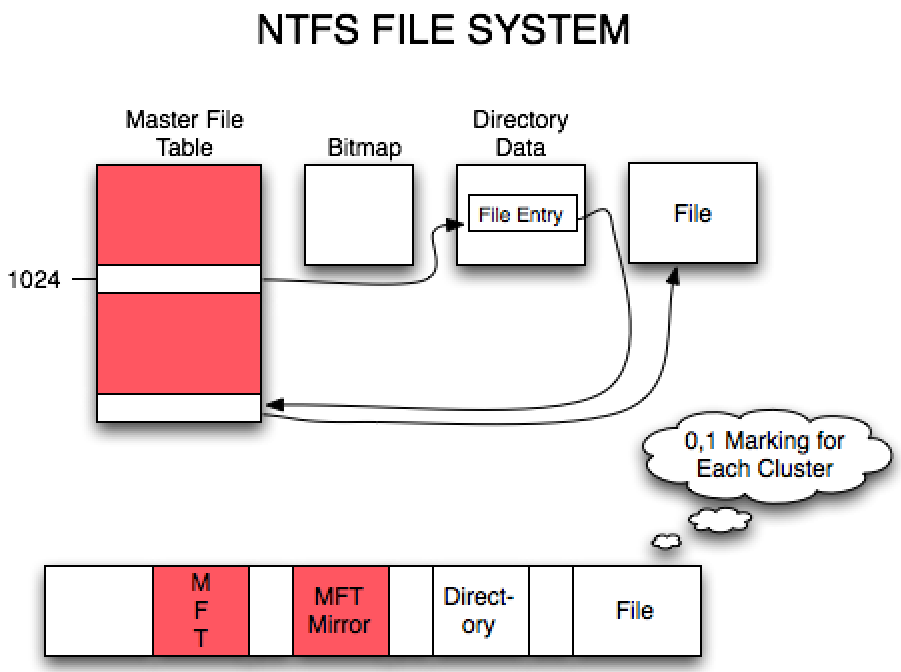
❖NTFS, the filesystem used by all modern versions of Windows

❖Instead of storing file metadata in tables scattered across the partition (usually to optimise hard disk seek times going from file metadata to actual file data), NTFS stores all the file metadata in a few large contiguous blocks.

❖These blocks are collectively known as the MFT (master file table).

❖Helps to quickly scan all the files in the partition to get each file's associated metadata. Instead of recursing through each directory manually and enumerating contents (like when computing summary statistics for directories)

File System — NTFS (Cont.)

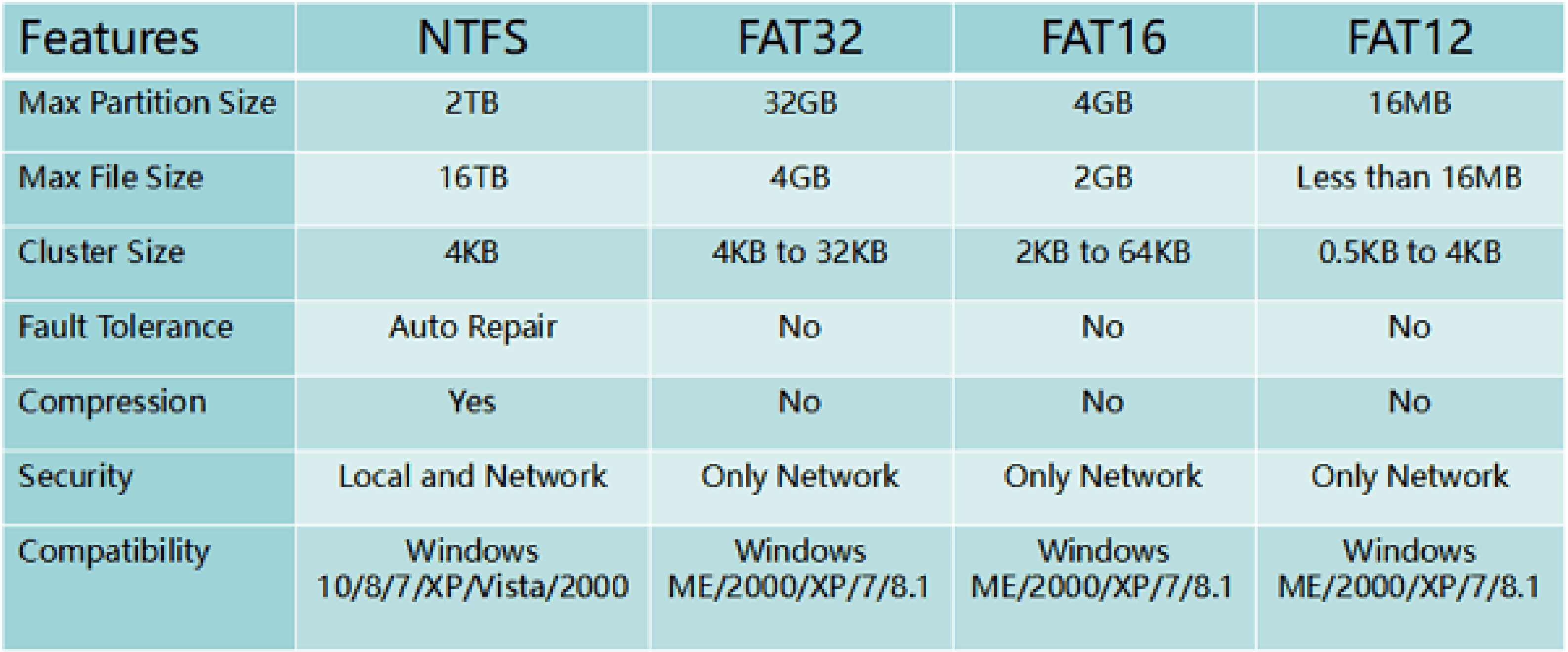


❖Every file in NTFS is described by one or more records in an array stored in a special file called the Master File Table (MFT)

❖Windows puts the Master File Table at the front of the drive, with a mirror file in the middle of the drive with only the first directory containing critical information.

❖The Bitmap is used to determine allocation/non-allocation of clusters.

File System — NTFS vs FAT

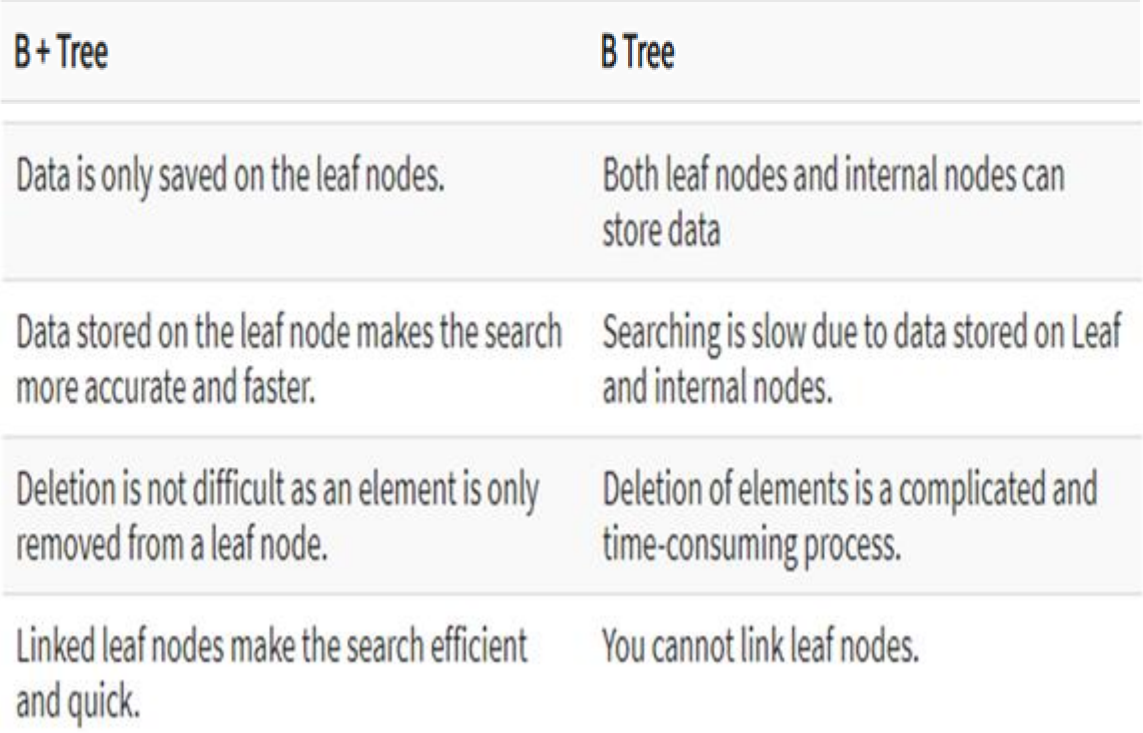


Features	NTFS	FAT32	FAT16	FAT12
Max Partition Size	2TB	32GB	4GB	16MB
Max File Size	16TB	4GB	2GB	Less than 16MB
Cluster Size	4KB	4KB to 32KB	2KB to 64KB	0.5KB to 4KB
Fault Tolerance	Auto Repair	No	No	No
Compression	Yes	No	No	No
Security	Local and Network	Only Network	Only Network	Only Network
Compatibility	Windows 10/8/7/XP/Vista/2000	Windows ME/2000/XP/7/8.1	Windows ME/2000/XP/7/8.1	Windows ME/2000/XP/7/8.1

File System — Internal Layout

- NTFS uses logical cluster numbers (LCNs) as disk addresses
- Physical disk offset in bytes = LCN x cluster size
- A file in NTFS is not a simple byte stream, as in MS-DOS or UNIX, rather, it is a structured object consisting of attributes
- Each file on an NTFS volume has a unique ID called a file reference.
  - 64-bit quantity that consists of a 48-bit file number and a 16-bit sequence number
  - Sequence number is incremented every time an MFT entry is reused, can be used to perform internal consistency checks (to catch a stale reference to a deleted file)
- The NTFS name space is organized by a hierarchy of directories; the index root contains the top level of the B+ tree
  - B+ tree is an extension of the B tree

File System—B+ Tree vs. B Tree



B + Tree	B Tree
Data is only saved on the leaf nodes.	Both leaf nodes and internal nodes can store data
Data stored on the leaf node makes the search more accurate and faster.	Searching is slow due to data stored on Leaf and internal nodes.
Deletion is not difficult as an element is only removed from a leaf node.	Deletion of elements is a complicated and time-consuming process.
Linked leaf nodes make the search efficient and quick.	You cannot link leaf nodes.

The drawback of B-tree used for indexing, however is that it stores the data pointer (a pointer to the disk file block containing the key value), corresponding to a particular key value, along with that key value in the node of a B-tree. This technique, greatly reduces the number of entries that can be packed into a node of a B-tree, thereby contributing to the increase in the number of levels in the B-tree, hence increasing the search time of a record.

B+ tree eliminates the above drawback by storing data pointers only at the leaf nodes of the tree.

File System - Recovery

- All file system data structure updates are performed inside transactions that are logged.
    - Before a data structure is altered, the transaction writes a log record that contains redo and undo information.
    - After the data structure has been changed, a commit record is written to the log to signify that the transaction succeeded.
    - After a crash, the file system data structures can be restored to a consistent state by processing the log records.

File System – Recovery (Cont.)

- This scheme does not guarantee that all the user file data can be recovered after a crash, just that the file system data structures (the metadata files) are undamaged and reflect some consistent state prior to the crash.
- The log is stored in the third metadata file at the beginning of the volume.
- The logging functionality is provided by the *log-file service which keeps track of free space in the log file*
  - *halts new I/O operations in case free space gets too low*

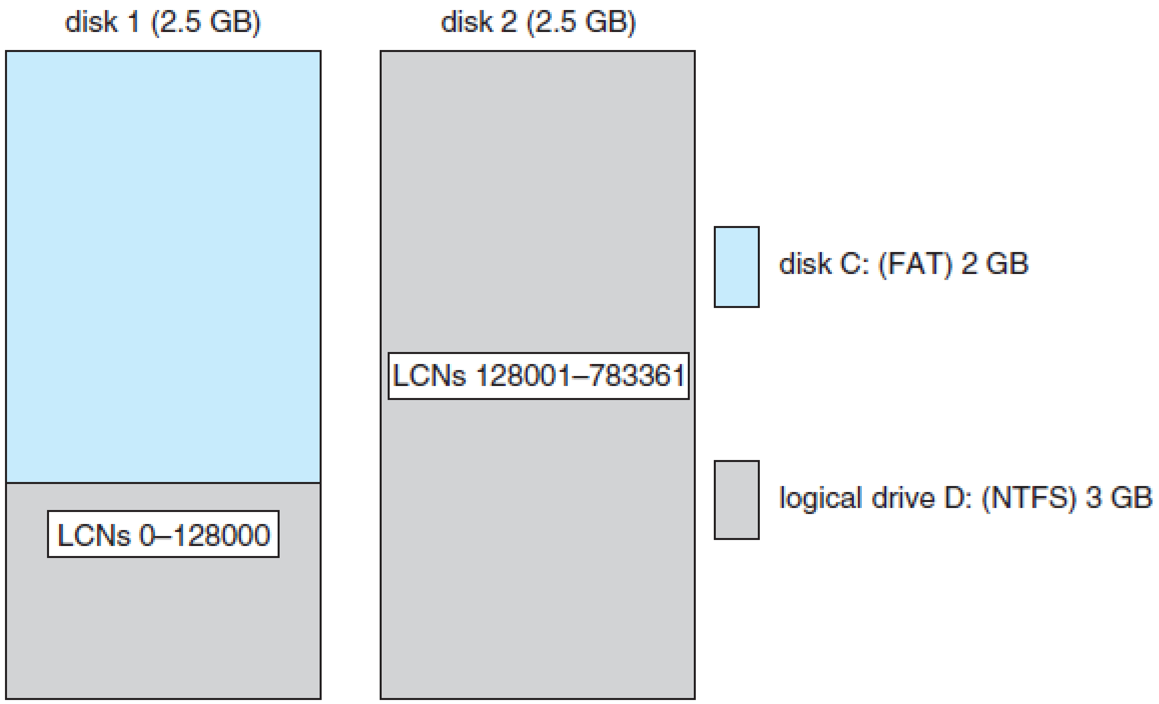
File System - Security

- Security of an NTFS volume is derived from the Windows object model.
- Each file object has a security descriptor attribute stored in this MFT record.
- This attribute contains the access token of the owner of the file, and an access control list that states the access privileges that are granted to each user that has access to the file.

Volume Management and Fault Tolerance

- **FtDisk**, the fault tolerant disk driver provides several ways to combine multiple SCSI disk drives into one logical volume
- Logically concatenate multiple disks to form a large logical volume, a *volume set*
- Interleave multiple physical partitions in round-robin fashion to form a ***stripe set*** (also called RAID level 0, or "disk striping")
  - FtDisk uses a stripe size of 64 KB – 1st 64 KB of the logical volume are stored in the 1st physical partition, 2nd 64 KB in the 2nd physical partition, and so on
  - Windows also supports RAID level 1 (mirroring) and RAID level 5 (*stripe set with parity)*
- Disk mirroring, or RAID level 1, is a robust scheme that uses a ***mirror set*** — two equally sized partitions on tow disks with identical data contents
- To deal with disk sectors that go bad, FtDisk, uses a hardware technique called ***sector sparing*** *(i.e. formatting a disk drive leaves extra sectors unmapped as spares)* and NTFS uses a software technique called ***cluster remapping*** *(i.e changes the pointers in the MFT from bad block to unallocated block)*

Volume Set On Two Drives

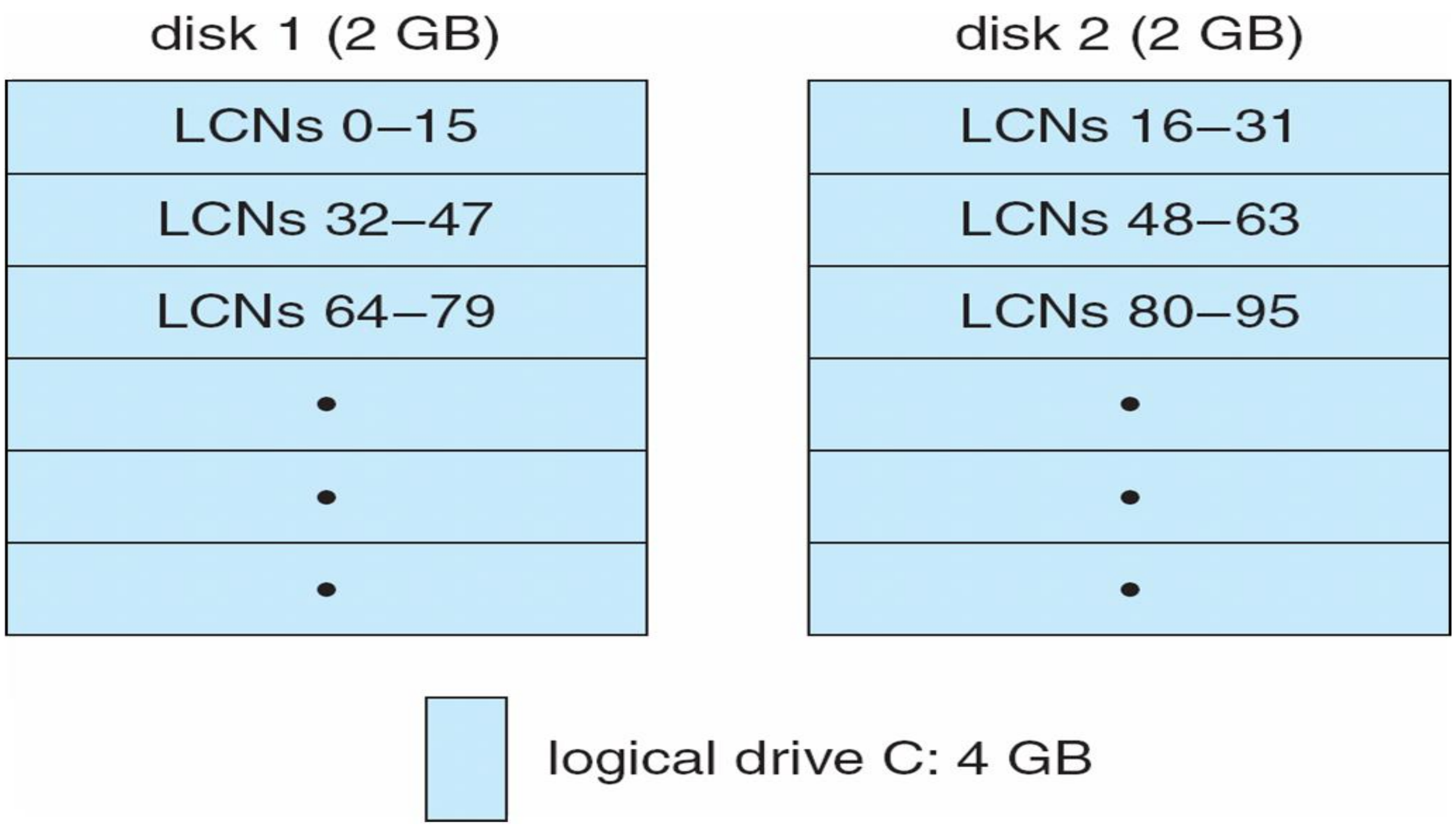


Logical volume or a *volume set can consist of up to 32 physical partitions A volume set can be extended by extending the bitmap metadata on the NTFS volume to cover the newly added space (bitmap contains information about free or used data blocks/clusters on a volume)*

Fig 1==> 128000 x 4 KB = 0.5 GB

Fig 2 => 655360 x 4 KB = 2.5 GB

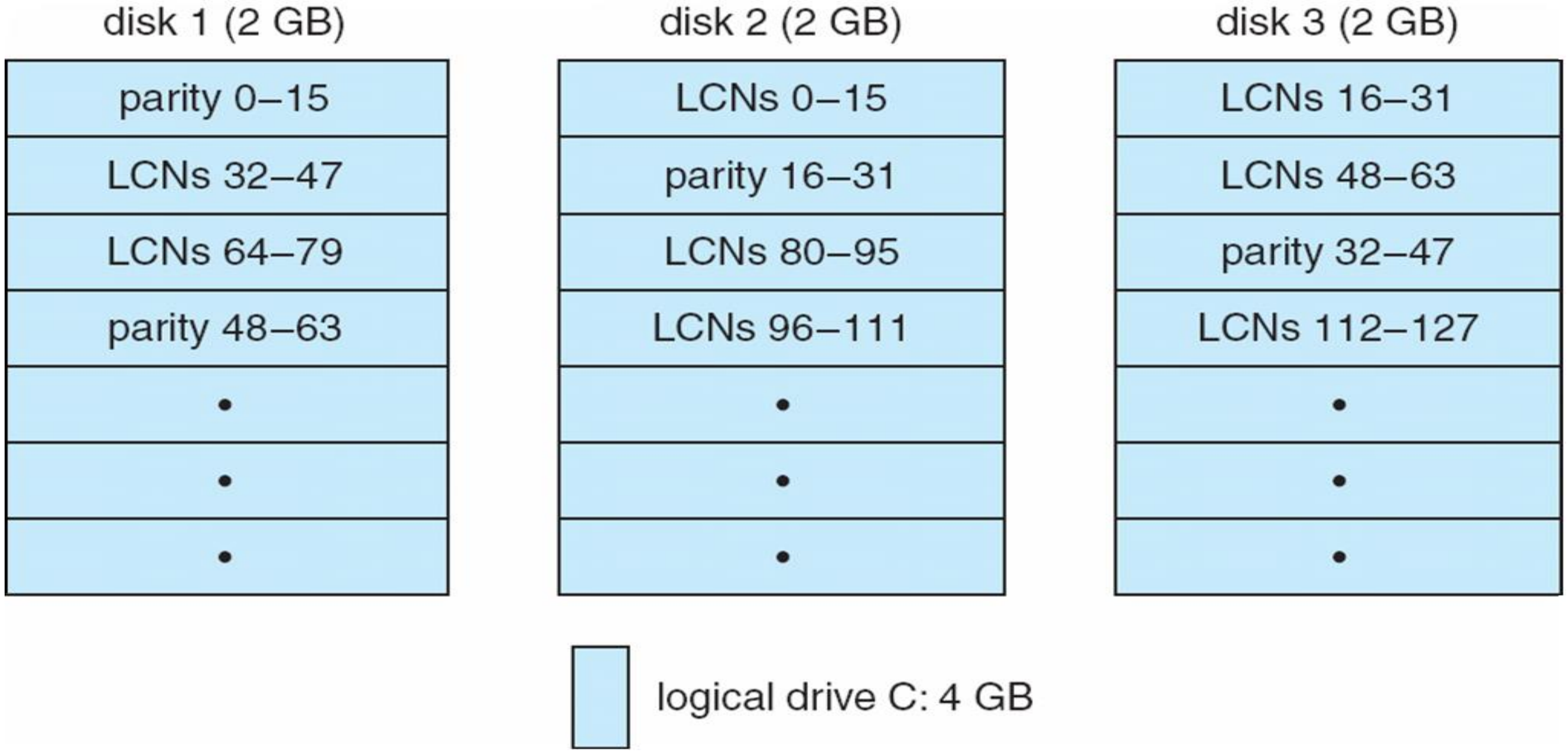
Stripe Set on Two Drives



disk 1 (2 GB)	disk 2 (2 GB)
LCNs 0–15	LCNs 16–31
LCNs 32–47	LCNs 48–63
LCNs 64–79	LCNs 80–95
•	•
•	•
•	•

🟦 logical drive C: 4 GB

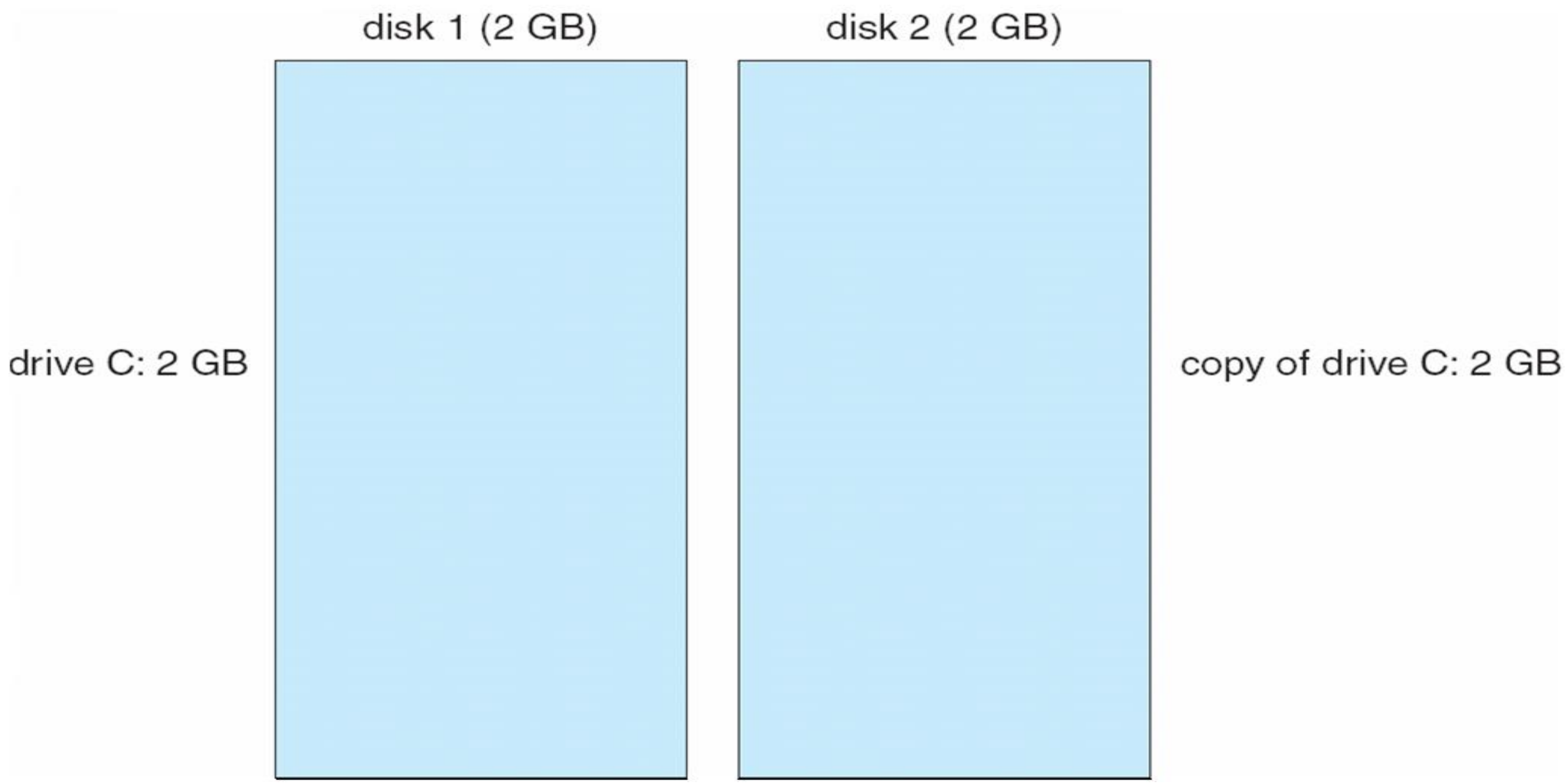
Stripe Set With Parity on Three Drives



disk 1 (2 GB)	disk 2 (2 GB)	disk 3 (2 GB)
parity 0–15	LCNs 0–15	LCNs 16–31
LCNs 32–47	parity 16–31	LCNs 48–63
LCNs 64–79	LCNs 80–95	parity 32–47
parity 48–63	LCNs 96–111	LCNs 112–127
•	•	•
•	•	•
•	•	•

□ logical drive C: 4 GB

Mirror Set on Two Drives



File System — Compression

- To compress a file, NTFS divides the file's data into *compression units*, which are blocks of 16 contiguous clusters.
  - To improve performance when reading contiguous compression units, NTFS pre-fetches and decompresses ahead of the application requests.
- For sparse files, NTFS uses another technique to save space.
  - Clusters that contain all zeros (ie. never been written) are not actually allocated or stored on disk.
  - Instead, gaps are left in the sequence of virtual cluster numbers stored in the MFT entry for the file.
  - When reading a file, if a gap in the virtual cluster numbers is found, NTFS just zero-fills that portion of the caller's buffer.

File System (Contd.)

- Mount Points, Symbolic links, Hard Links
  - Hard links: A single file has an entry in more than one directory of the same volume
- NTFS Journal – describes all changes that have been made to the file system.
  - Used by services such as User-mode (to identify what files have changed), search indexer (to re-index files) and file-replication (to replicate files across the network)
- Snapshots to create a shadow copy of volume for backup (and recovery of files if accidentally deleted)

PES UNIVERSITY ONLINE

THANK YOU

Kakoli Bora

Department of Computer Science Engineering

k_bora@pes.edu