**TOPIC 1 - FILE SYSTEMS**

Preconceptions:

A file system is essentially a tree. A root at the top, then branching files working their way down. There is no distinction between a file and a directory. There is an operating system defined file structure, but user data is completely at the whim of the user, and how much they wish to organize their data.

Conversation:

This is the link to the conversation. Copying and pasting the whole thing in here seems ridiculous.

https://chatgpt.com/share/67378cd3-5340-8013-a672-a318d8ae48d4

What I learned:

* EXT4
  + Default fs on Linux
  + EXT4 is backwards compatible with EXT2 and EXT3
  + EXT4 has a maximum file system size of 1 exabyte
  + EXT4 supports files of up to 16 TiB
  + EXT4 uses extents, not block mapping to reduce fragmentation
  + Not just files and directories are files, but so are hardware devices and sockets
* NTFS
  + Default FS on Windows
  + Similar file system size to EXT4, but individual files are limited to just 1 TiB (not that it really matters, though)
  + More fragmentation than with EXT4 because it does not use extents
* APFS
  + Default FS on MacOS
  + 8 exabyte file system size and file size
  + Highly optimized for SSDs
  + Uses extends and copy-on-write
  + Native snapshot support
* Copy-On-Write
  + When data is accessed, no copies are made.
  + When a user tries to modify data, then a copy is made
  + The copy is modified, then written, preserving the original data for other processes
* EXTENTS:
  + Data is stored contiguously, so there is less metadata needed. Just a pointer and size
  + Better at storing large files
  + Reduced fragmentation
  + Better performance since data is not spread out
  + More complex than block mapping
* Journaling
  + A system wherein disk operations are written to a temporary file, then the operations are completed
  + This ensures that in the event of a loss of power, the operations are preserved, allowing recovery to start.

**TOPIC 2 - HDD VS SSD**

Preconceptions:

I know that an HDD is a set of spinning disks, where the bits are magnetically inscribed into the metal, whereas an SSD uses NAND flash to store data. They can both use the SATA interface, but I don't know if they work the same way under the hood.

Conversation:

What I learned:

* Operating systems do handle HDDs and SSDs differently. Since they each have their own advantages and disadvantages, the way the OS interacts with them will change
* DEFRAGMENTATION
  + HDDs require defragmentation to keep read speeds high. This is because the data is physically on different places that take time to find.
  + SSDs have static lookup time, so defragmentation does not matter.
* TRIM
  + A TRIM command pre-erases data blocks so that the later write is faster.
  + HDDs do not worry about TRIM commands, since they aren't focused much on speed.
  + SSDs use TRIM to maintain the high read and write speeds that they are known for.
* Read / Write scheduling
  + HDDs will schedule I/O operations to be optimized for the arm
  + SSDs have constant lookup time, so scheduling isn't much of a concern. They just optimize for total throughput
* Garbage Collection
  + Similar to the TRIM command, HDDs don't worry about this, but SSDs do.
* Hardware Differentiation
  + Given a SATA SSD and a SATA HDD, they can be treated the same, seeing as both are SATA compliant. These means that any system with a SATA interface can support SATA SSDs. That being said, software support will treat them like hard drives, so there may not be much benefit.
  + This is due to the hardware on the disks themselves, which handle the architecture differences and translate it to SATA calls.
* NVME SSDs
  + Given the proper adapters, any NVME SSD can be used on any system with PCIe. That being said, it may not be seen as a bootable drive if the BIOS does not support it.
  + That being said, it can still be used as a storage drive, just not the boot drive.­