Unit 7 Storage Solutions in a Silicone Nutshell

**Corey Crooks**

**Purdue University Global**

**IT332 – Principles of Information Systems Architecture**

**Jonathan Abramson**

**August 23rd, 2023**

Disks allocation isn’t as simple as one may think. There are many different factors that affect what data goes in which spots of the storage solution a user may be leveraging to retain information. The simplest way to understand storage solutions in the modern computing era would be to break the concepts into the smallest forms, and work up from there.

One popular way programs interact with a computerized system is through memory. This memory can story the program either contiguously or non-contiguously. Essentially, this determines if the program is stored sequentially or not (Watson at Louisiana Tech Universtiy, 2023).Contiguous storage is a discipline of program storage that dictates the program is stored in memory as one solid block of memory addresses. This can be particularly essential to decrease complexity and timings needed to access and write to specific programs. The other way of storing is simply the opposite of that. Non-contiguous storage dictates the program is stored into segments that may be fragmented through the storage medium in question. These fragments exist in different memory addresses, and are not one-after-the-other like in contiguous storage (Watson at Louisiana Tech Universtiy, 2023). This can fix an issue with contiguous storage in that given the variety and complexity of programs and applications running on today’s hardware, finding a contiguous block of memory addresses large enough to store the data required may be impossible. Non-contiguous storage instead will take the address blocks that are given, and simply look for more if need be.

These types of storage solutions are important to understand, but it doesn’t fully explain how these files are linked to one another. Indeed, there are many segments of blocks for a file in non-contiguous storage, but the Directory Table and File Allocation Table can point the computer in the direction of what blocks hold information relating to which files for these programs. The File Directory Table is responsible for holding a File Name, Starting Block, and metadata associated with each file (Jared Brown, n.d.). This is used by the computer while looking for a specific file, which can be found in the file name cell. The next cell is responsible for communicating the starting block. This is crucial information to starting to string together non-contiguous data. The starting block can be used with the File Allocation Table in order to point to the starting point, and next blocks in memory. This is done by the two cells the File Allocation Table’s storage solution offers. The first cell keeps track of whether or not the cell is being currently utilized, while the second cell points to the next block of data in the file’s sequence (Jared Brown, n.d.). This can be used to start and complete a chain of data. For instance, a file can be read together starting with the directory table’s starting point, following that starting point, and then using the File Allocation Table to read and follow the next segment until the end of the program. Now that there is a great way to store and read files, there needs to be a way to find files. This technique is called File Indexing. File indexing works by giving files a unique identifier, called and index, and using that as the aforementioned “metadata” to essentially label a specific file (Montana State University, n.d.). This allows files to be very easily read and searched through without having to read through the entire file to determine if it is the content you are looking for.

These concepts have been widely used throughout the digital age. Now that storage solutions have been explained using file directories and allocation, disk allocation comes easily. These addresses are not only used in Random Access Memory, but also used in other common storage solutions like a Tape drive of the old days, a Hard Disk Drive, or even one of the many forms of Solid-State Drive (Virginia Tech, 2023). All of these mediums use similar allocation protocols detailed by these directories and storage methods to ensure files and structures can be read and wrote to efficiently and effectively without the lost of other important data along the way. The main difference between these drives are usually the physical aspects to them and the unique advantages they offer. For example, a Hard Disk drive may be much more reliable than a Solid-State Drive due to the physical aspect of the data being stored. On the other hand, a Solid-State drive is many times faster than Hard Disk technology simply because of the lack of moving parts, and need to wait for disk revolutions (ITI College, 2023). Another similarity between these drives are the way the data is logically organized on a macro scale. On a typical Windows installation, it is simple to see the disk layout of a user’s system through the Disk Management tool suite. In this, a user is able to set a number of key information about the drives that they are using in their system. For example, a user will be able to separate their physical drive into smaller logical drives called volumes or partitions. These partitions cannot exceed the total length of the host drive, as they require the space on the host drive in order to function (The Open University, 2023).This could be commonly seen in a system if in the windows explorer, a user has a number of drive letters assigned to their computer’s storage such as “C:”, “P:”, and so on despite only having one storage solution. This could be important for a number of reasons such as designating a specific space for backups, boot drives, and more that a computer may need to function, and maintain secure operation.

One last important concept to fully understanding the basics of storage mediums would be fragmentation of this data. As storage has its lifecycle, many programs will need access to it. As these programs all request access, it would be impossible to give each program a contiguous space on the storage all at once. Instead, non-contiguous storage is utilized to speed writes, as it is possible to operate as needed in today’s computing climates with such technology. This does create somewhat of an issue, however. All of these programs writing to different parts of the storage medium creates what is referred to as Fragmentation (The Open University, 2023). This is labeled so because of the fragmented nature of the program’s storage on the disk itself. On certain storage solutions such as a Hard Disk, this can lead to slower read and write times due to a need to wait for the revolutions of the disk to access specific parts of a program, as that is not stored sequentially. This is where a logical process called Disk Defragmentation comes into play. Disk Defragmentation re-arranges a user’s disk to optimize the layout in order to speed read and write times by ordering the data in a more logical and streamlined way—essentially un-doing all of the fragmentation caused by the random access from the disk (The Open University, 2023). Although, it is important to note that technologies like Solid-State drives do not need to wait for physical components to make revolutions, and do not suffer significant performance penalties from the defragmentation process.

Throughout the many storage solutions in today’s digital era, it can be easy to be lost in how exactly they all work. From partitions, logical drives, and fragmentations to file allocation, contiguous storage, and indexing, there are many different logics that go into making the storage in these computers function as efficiently as they do. Although they do have their own issues with either approach, it is important to understand exactly what is needed in each storage, to better utilize it in a personal or professional capacity.

**Medium Comparison Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Storage Medium** | Hard Disk | Tape Disk | Flash Drive | Solid State Drive |
| **Cost Ranking**  **(higher is more expensive)** | 2 | 3 | 1 | 4 |
| **Speed Ranking**  **(higher is slower)** | 2 | 4 | 3 | 1 |
| **Pros** | -High capacity for cheap  -Will function as expected over a long period of time. | -Very durable  -Lower failure rate than HDD  -Can keep data perfectly even in long periods of non-use. | -Very Portable  -Very Cheap  -Small form factor helps organization for different drives for different things.  -Easy to share data with other systems  -Will work on most consumer systems  -“plug and play” | -Extremely fast  -Varying form factors help increase availability depending on system motherboard. |
| **Cons** | -Power usage even when idling  -Prone to mechanical failure | -Slower than HDD  -Forced sequential storage | -Easy to lose  -Low overall capacity  -Very little data integrity  -Vulnerable for social engineering virus attacks | -Costly  -Not as high capacity as HDD or Tape Disk  -Must have power regularly or data will degrade |
| **Source** | (The Open University, n.d.) | (FujiFilm, n.d.) | (Ismail-Beigi Research Group, 2023) | (Uriarte, n.d.) |

# **References**

FujiFilm. (n.d.). *Disk Storage vs. Tape Storage*. Retrieved from tape-storage.net: https://www.tape-storage.net/en/storage\_comparison/article\_01/#:~:text=High%20Reliability&text=However%2C%20tape%20storage%20today%20has,a%20highly%20reliable%20storage%20system.

Ismail-Beigi Research Group. (2023). *Flash Drives: Application*. Retrieved from yale.edu: https://volga.eng.yale.edu/teaching-resources/flash-drives/application

ITI College. (2023). *Hard Drive vs SSD – Which is Better for Business*. Retrieved from iticollege.edu: https://iticollege.edu/blog/hard-drive-vs-ssd/

Jared Brown, S. G. (n.d.). *CS 111 Lecture 11: File System Design 11/6/2013*. Retrieved from ucla.edu: http://web.cs.ucla.edu/classes/fall13/cs111/scribe/11g/index.html

Montana State University. (n.d.). *Indexing Files*. Retrieved from cs.montana.edu: https://www.cs.montana.edu/courses/spring2004/435/lectures/IndexingFiles.html

The Open University. (2023). *Formatting a hard disk*. Retrieved from www.open.edu: https://www.open.edu/openlearn/digital-computing/introducing-computing-and-it/content-section-5.3

The Open University. (2023). *Fragmentation*. Retrieved from www.open.edu: https://www.open.edu/openlearn/digital-computing/introducing-computing-and-it/content-section-5.5

The Open University. (n.d.). *3.4 Backup media*. Retrieved from open.edu: https://www.open.edu/openlearn/mod/oucontent/view.php?id=48328&section=4.4

Uriarte, T. T. (n.d.). *Solid State Drive Architecture*. Retrieved from Rochester Institute of Technology: http://meseec.ce.rit.edu/551-projects/fall2010/1-4.pdf

Virginia Tech. (2023). *Dynamic Storage Allocation*. Retrieved from CS3 Data Structures and Algorithms: https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/Dynamic.html

Watson at Louisiana Tech Universtiy. (2023). *3.3.1 Contiguous storage* . Retrieved from watson.latech.edu: http://watson.latech.edu/book/information/informationOrganization1.html