**Week 04 Journal - Class Handouts + Chapter 7 from A+ textbook** Seraphim Gerber

For a computer system to execute programs, the data they access must be in main memory. Ideally, we would want the programs and data to permanently live in main memory, however that’s not possible as main memory is too small and it loses its contents when power is disabled. Because of this, secondary storage is expected to be able to hold large amounts of data permanently.

Secondary storage is provided by disks. They come in multi-platter disk packages, with each platter having two surfaces and being divided into single tracks and each track having several sectors. A cylinder is the set of tracks which are on the same position for all disk surfaces. To access a sector, you must know the drive number, track/cylinder, surface, and sector. A sector is the smallest data type that can be read/written on a block device.

In order to use hardware efficiently, the operating system needs a fast access time and disk bandwidth. Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer. Access time is calculated by seek time, latency time, and transfer time.

As files are deleted or created, different areas of the disk become freed or occupied. This is called a free space list, kept track of by the disk manager in three ways: bitmap or bit vector, linked lists, or free sequence lists.

A simple method for storage is called a fixed or contiguous allocation. With this method, a file’s maximum size is determined upon its creation, making it unable to surpass this limit. For linked-list allocation, each file is associated with a pointer next to the block that indicates entries. With segment-based allocation, a segment table is used to allocate multiple regions of contiguous blocks. Indexed allocation uses an index to directly track the file block locations. Multilevel indexed allocation, used by Linux, has certain index entries point to index blocks as opposed to data blocks. Inverted allocation uses a disk block by hashing the file block content to a disk block location.

Hard drives are popular devices for storing data. They can be mounted inside the computer or attached with a USB or other external devices. Hard drives store more than flash drives and come in different physical sizes and form factors.

SSDs, also known as solid state drives, are storage devices that use DRAM or nonvolatile flash memory, unlike hard drives. SSDs solve the main issue of hard drives, moving parts, by using flash memory that makes it low heat producing, reliable, quiet, secure, long-lasting, and fast. They can be installed as internal or external units and are common in tablets, laptops, and mobile devices.

For a hard drive to operate, it needs a set of rules. The rules state the number of heads on the drive, what commands the drive responds to, the cables used with the drive, the number of devices that can be supported, the number of data bits transferred at one time, and more. These rules create the interface that communicates with the harddrive. The two major interfaces are called integrated drive electronics, IDE, and small computer system interface, SCSI. IDE is the most common in home and office computers and SCSI is most common in network servers.

Both IDE and SCSI started out as parallel architectures, meaning that multiple bits can be sent over multiple paths. As transfer rates increase, precise timing is required. With both IDE and SCSI, multiple devices can attach to the same bus. Parallel IDE supports only two devices, whereas parallel SCSI supports more, however, the concept is still the same. When there are multiple devices sharing the bus, they must wait their turn and deal with configuration issues. Today we focus on serial architectures, which are available on both IDE and SCSI. Serial architecture is a point-to-point bus where each device has a single connection back to the controller, and bits are sent out one at a time over a single link. More devices can attach and the configuration is much easier.

Once a hard drive is installed and configured, the drive requires preparation to function. The two steps of hard drive preparation are partitioning the drive and high-level formatting it. Partitioning allows a drive letter to be assigned to one or more parts of the hard drive, dividing it so the computer system sees the hard drive as more than one drive. Fault tolerance is the ability to continue functioning after a hardware or software failure. It can be built into the system to remove the risk of failures.

**Have you used cloud storage in the past? Do you prefer local or cloud storage of your data? Explain your answer.**

Yes. I prefer local storage as it is easier for me to access since I am more accustomed to it and although it isn’t necessarily more private, it certainly feels that way.

**Do you think adding more platters to a hard drive increases or decreases its mean time between failures (MBTF)?**

I think it would decrease MBTF as there are less parts to worry about.

**Based on the storage management handout, do you think contiguous, chained, or indexed allocation is the best method for allocating hard drive space?**

I think linked list allocation is the best as it isn’t limited to a single size and doesn’t suffer from external fragmentation.

**Have you encountered RAID in the past? Based on the resilience/performance tradeoffs, which level of RAID do you prefer?**

No, but I prefer level 1.