Ex. 5.1 |(10pts) What is the capacity of a hard disk with 64 sectors per track with 128 tracks, if one sector is 128 bytes.

128 Bytes* 128 tracks* 64 sectors= 1MB

Ex. 5.2 | (20pts) Consider the hard disk defined in Ex. 5.1. If we use the directory entry definition in the following figure and have the directory table with 100 of these entries (i.e., MaxNumFiles = 100), what is the largest file this file system can create assuming we use the contiguous allocation method? Hint: the size of a character is 1 byte; the size of boolean is 1 byte, and the size of an integer is 4 bytes. Let MaxFileNameLength be 7. (Note that for the size of one entry, DO NOT make it to the closest power of 2.)

((7+1) * 1) + 3*1+2*4
=(MaxFileNameLength be 7+ 1 for terminating character) * (character size)+ 3 boolean size+ 2 integer size
=19 bytes
Max number of files in our system is 100 files
The total number of the directory table is 1900 bytes
For a sector is 128 bytes, we need 15 sectors to store the directory table.
So the largest file can be created= 128*128*64-15*128=1046656 B

Ex. 5.3 | (20pts) Consider a system with a hard disk with 1024 tracks. Each track has 256 sectors. The disk sector size is 128 bytes. We want to use a FAT table to keep track of files and free-blocks. Each FAT table entry is 4 bytes. Calculate how many entries there are in the FAT table and how big the above FAT table is, in terms of memory requirement.

256 sectors/track * 1024 tracks =262144 entries.= 2^18
2^18 * 4 = 2^20=1MB
1MB/128Bytes=2^13 sectors are needed

Ex. 5.4 | (20pts) For the hard disk in Ex. 5.3, what is the largest size of file that can be created by this scheme? Assume the directory file size is 400bytes.

400/128=4 sectors
256*1024(total size)-2^13(FAT size)-4(file size)= 253948 sectors
The largest size of file=253948*128=32505344 B

Ex. 5.5 | (10pts) What are the three access time delays in hard disk access? Explain them.

Seek time delay: When anything is read or written to a disc drive, the read/write head of the disc needs to move to the right position. The actual physical positioning of the read/write head of the disc is called seeking. The amount of time that it takes the read/write head of the disc to move

from one part of the disk to another is called the seek time delay. HDD must move the arm onto the right track.The seek time delay can differ for a given disc due to the varying distance from the start point to where the read/write head has been instructed to go.
Rotational delay: A rotational delay is the time between information requests and how long it takes the hard drive to move to the correct sector. In other words, it is a time measurement, in ms (milliseconds), of how long before a rotating drive can transfer data.wait until the right sector comes underneath the arm.
Transfer delay: Transfer delay is to access the sector to transfer the data to the disk controller and eventually to the main memory. It is the time interval between the start of the transfer and the completion of the transfer. It not only depends on the rotational Speed of a disk. But it also depends on track and sector density. More the density, lesser is the time. Disk transfer time also depends on the amount of data to be transferred.

Ex. 5.6 | (10pts) Explain the transparency in a distributed system.

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Access Transparency: Clients should be unaware of the distribution of the files. The files could be present on a totally different set of servers which are physically distant apart and a single set of operations should be provided to access these remote as well as the local files. Applications written for the local file should be able to be executed even for the remote files. The examples illustrating this property are the File system in Network File System (NFS), SQL queries, and Navigation of the web.

Location Transparency: Clients should see a uniform file name space. Files or groups of files may be relocated without changing their pathnames. A location transparent name contains no information about the named object's physical location. This property is important to support the

movement of the resources and the availability of services. The location and access transparencies together are sometimes referred as Network transparency. The examples are File system in NFS and the pages of the web
Migration Transparency: This transparency allows the user to be unaware of the movement of information or processes within a system without affecting the operations of the users and the applications that are running. This mechanism allows for the load balancing of any particular client, which might be overloaded. The systems that implement this transparency are NFS and Web pages.

Replication Transparency: This kind of transparency should be mainly incorporated for the distributed file systems, which replicate the data at two or more sites for more reliability. The client generally should not be aware that a replicated copy of the data exists. The clients should also expect operations to return only one set of values. The examples are Distributed DBMS and Mirroring of Web pages

Concurrency Transparency: Users and Applications should be able to access shared data or objects without interference between each other. This requires very complex mechanisms in a distributed system, since there exists true concurrency rather than the simulated concurrency of a central system. The shared objects are accessed simultaneously. The concurrency control and its implementation is a hard task. The examples are NFS, Automatic Teller machine (ATM) network.

Failure Transparency: Enables the concealment of faults, allowing user and application programs to complete their tasks despite the failure of hardware or software components. Fault tolerance is provided by the mechanisms that relate to access transparency. The distributed system are more prone to failures as any of the component may fail which may lead to degraded service or the total absence of that service. As the intricacies are hidden the distinction between a failed and a slow running process is difficult. Examples are Database Management Systems

Ex. 5.7 | (10pts) Discuss the pros and cons of mechanical hard disks and Solid State Drives. (Full credits if give more than 3 pairs of comparison.)

HDD is a traditional hard drive that are basically metal platters with a magnetic coating that stores your data. The platters spin, allowing a read/write head on an arm to access the data. This contraption is stored in a hard drive enclosure.

SSDs have the same functionality as HDDs (they boot up your system and store applications and data on your computer), but instead of a magnetic coating on platters, SSDs store data on flash memory chips. These chips are installed either on the system’s motherboard, on a PCI/PCIe card, or in a hard drive box. SSDs aren’t the same thing as a USB thumb drive, even though they also use flash memory. (SSDs are much faster and more reliable.)

Pros and cons:

Speed — Speed is one of the SSD’s primary pros. SSD-driven machines will boot up in a matter of seconds. HDDs take time to speed up, and even when operating will run slower than an equivalent SSD. This means an SSD-driven computer will boot up faster, launch applications faster, and will run faster.

Fragmentation — When HDDs start to fill up, large files often become scattered around the disk platter (a.k.a. fragmentation). SSDs, on the other hand, aren’t affected by where the data is stored on their chips. So you don’t have to deal with fragmentation on an SSD.

Durability — HDDs are mechanical and subject to mechanical failure. SSDs are therefore much less likely to get broken, and are more reliable in preventing your data from disappearing due to drive failure.

Physical Size — Because HDDs rely on mechanical, moving parts, they take up more physical space than SSDs. SSDs continue to get smaller, and are more convenient and portable.

Noise — HDDs are just plain noisy when the plates are spinning and the reading arm is moving back and forth. SSDs are completely quiet, since there are no moving parts.

Price —In terms of dollar per GB, SSDs are more expensive. They’re a newer technology, so they’re going to stay more expensive than HDDs for a while. Currently, you’ll pay about 7 cents per gigabyte for an HDD and 14 cents per gigabyte for an SSD.

Overall, although HDDs are cheaper, SSDs win on performance. Also, SSDs don’t wear out as quickly as HDDs. If you buy an SSD, it will probably become obsolete before it wears out. HDDs, on the other hand, will usually wear out before you want them to. So although SSDs are pricier up front, they can be a better deal in the long run.