[SEQUENTIAL FILES] The notion of sequential file is important. File means,dosya in Turkish, piece of paper.It is piece of paper and we call this as file. That could be long file where you roll beginning and end like in historical documents or old religious books where you can roll the whole book into one piece of paper . It is a file. So, as you read the file the typical behaviour is going up and down in the file by modifying these rools so you can roll uprward to read continuation of the text if you start from the beginning as you read you need to roll the file slowly until the the end once you read to end , end of the file, has a marker as a special character (EOF) at the end of the file as a ASCII character when you see this, it means oo this is the end of the rood, so no more text to read, sometimes file has an index at the beginning and by looking at index you look for a certain chapter, you look for the chapter and immediately move to a certain chapter and read from that chapter so the first one is called as SEQUENTIAL access, second one is actually, RANDOM ACCESS. Random means it doesn’t mean by chance , it means anywhere you want. One after another from the beginning, is named as sequential access, in sequential access, like old kaset teyps, remember these kaset teyps in 1980s, you need to listen to song 1 , 2, 3… you can not jump, in order to jump, you need to move forward. So that is pure sequential access, for the random access, CD room or old record players, mechanical record players that you can select the track hold the arm put it in a different song, s o you can randomly select each individual song on the CDs or records, so that is random access. If we are storing certain things of the same size each of them has certain size, so when you need to access to the fifth element you probably know where you look for so you can immediately go to a specific position and read it, that is random access or in file in C language, file systems it is also called \*direct access\* so you can reposition the pointer , we call it as file pointer, you can reposition where you are interested immediately by calculating calculating multiplying the size of the record with 5 and then if you fastforward your kaset types to certain (sektıns)? , fastforward like, fastforward 3 seconds then it will jump into 3.song. so that is direct access, random access , usually interchencably, and sequential access means one after another but file itself sequential, file is sequential, it is all files are like this.sometimes it might be fragmented. For exmp, In the book of Islam, you have different pieces of segments, those segments are separately written, and at the end, they collected. Order of writing is different than order of arrangement. Arrangement is from the biggest one to smallest one. Therefore , you have a file. Unlike book of “..” you have a file with different size of segments and each segment is written in different time so the location and the chronological is different than positional order. So, in harddrives on in storage system pieces of assume that this is a long file, pieces of information , maybe in different position in the file, in the file system. This part, for exmp. Part a ,b,c,d,e,f.. might be chronological order. 1=2.,b=2,c=3,d=4.. but on the file, it could be f,c,a,d,e,b. on the file system, it could be a different order that is possible. Therefore, in these cases we need to track of each beginning , (her surenin başı vardır ya hangi sayfa) we need to keep track of which segment starts where. In the page, therefore whenever we need to go particular page, we don’t scroll from the beginning directly go to a specific part. That is how ‘’’…..’’’ . you have multiple records. Each record is containing certain piece of information that information could be fixed length or variable length if it is fixed length your task is easy for exmp. Each chapter in the religious book is 10 pages. If you need go to 20. Chapter go to page 250 and here you go. No problem. But sizes are variable, then you need to have an index containing the beginning of the parts,chapters. Which is slightly difficult but it allows you to have variable sized chapter. Otherwise, if you don’t have anything to say in those chapters, you need to have blank pages. If you are okay with blank pages, then you have equal sized chapters without any problem.in some cases, these information is actually same size, such asstudent numbers of yours, grades.all Student numbers are of the same size. If I have fixed length of arrangement no more spaces wasted. [Soruya cevap = for example it is a textbook and it has 1-2-3-4-.. chapters. When you record it in the hard drive, on the arrangement of harddrive, chapter1 maybe saved a different place, so , in chronological sense, it could be located in somewhere else. It means that the first chapter is still the first chapter but in memory, it might not be first one. Since files are quite long, we may not have single place to store those information, separate peace of information maybe saved on harddrive, or database, databases as well which is important,so certain part of information can come before or after , regardless of the logical order of the arrangement, in file, they may stay in different places. For exmp, first student maybe in somewhere else second student is somewhere else bc first student is changed later so as you change things on the file, delete insort, places change. So, order of looking might be different than order of arrangement that is true for all similar system for operating system for databases for algotihms ü there is a reailty and there is a logical view logical view is usually different,ü usually, not always,, sometimes it is the same, usually different from the reality. Hani sırada adama abi ben senin arkandayım diyip kenarda oturmak gibi. Onun gibi. Gerçek arrangementle gerçek sıra aynı olmayabilir. Chapter index. You have an index and you have pointers so in the index certain piexces of information is labeled so whenever you want , ali boyacı for exmp. In the index ali boyacı, it says 2 . veli gümüş 3 . etc.. so, this is index(2,3), so whenever I need ali boyacı? Where is he? He is here(2). So, these student record can be automatically accessed, if you read this, it is automatically retrieved. You don’y have to read everything from the beginning. Otherwise, if you have access sequentially to the file, you have to check here . are you ali boyacı? No. are you? No. are you? No. until you find ali boyacı, there is no way, but if you have an index, then you can immediately move to a particular record and read it. Maybe(?), something else such as student number. Student number. How am I going to do these? (reversing places of ali boyacı and veli ) there are multiple ways, you can constract them as you enter information second if these are already written you need to find where they are, it is called as indexing like google indexing the web, which information is where ,in which address, in the google, you have “GTU” and in the index file, you have”” <http://unicersisdf.com.edu.tr/gtu> “” so some file, that comes as the first element, when you search, gtu on the web page on the google searching webs. It has multiple items , google keeps a list of character, so whenever you enter a certain keyword, it automatically looks for the record but where is this? Now it is stored. Tnis one (GTU) is put in a function for exmp. GTU,ODTU,ITU,Hasan there are other words, so, one strategy would be to have a file for each search term but if the user enter ITU in search engine, call is your algorithm in google search engine, files, this one bc this(tablonun solunda itu yazıyor sagındaki yeri gösteriyor) contains the first items related to ITU. It should contain address but in order to find, how am I going to behave I will take a temporary variable, is it ITU? No. Is it? Yes. So I make a search. It is not easy bc it is a linear search. It will be very small. Instead, ITU I can make a function like this: what is the ascii value for I for T for U. Lets make a function lets intvent a function that calls these functions to get another number, So, lets make a file. There is a place http//…… and this is related to ITU. If a search for itu as user, this function is executed ratherthan is it itu no no no.. I make a computation, find that index, at these index I write resulting addres therefore if the search term is ITU it is actually bound automatically, if I enter GTU , gtu will map to a different solution, probably 321. I don’t have a nice function but it is still a function. So, here 321 recordu contain https/…. Then, I make a function that helps me to search terms . We call it as HASHING . [ Soruya cevap = Hashingde aynı şeye düşme şeyi vardır. Probablitisesi vardır.In that case , you make a linked list, so put another pointer to point to second element so you go here(325.adresin bir altı) and is it ITU? No. then look at the second place is it itu? Yes. Then go on.[soruya cevap = veriden lokasyona fonksiyonla geçiş yapmak diyelim. Mesela Ahmet gündüz diye birini arıyorum tek tek insanlara soracağıma Ahmet gündüz buraya gel derim gelir. 2 kişi gelirse kimin benle işi var? diye sorarım o gelir.] [OPENING AN INDEXED FILE SLAYTI] . Indexler ITU , GTU, ODTU… etc. gösteriyor. And locations are automatically found. If this is a hash index, then that method is used, if it is linear index, you can search one by one. It could be binary tree,too. For example, ITU starts with I, then is it smaller than the middle? No. then go to the right., again go to the right, is it big or smaller, go to the left go to the right . that(using balanced binary search tree) will speed you up significantly. You can have log n(?) complexity rather than …. Complexity. [ HASHING SLAYTINA GECIYOR] . In the hashing, each record has a key field.(which is GTU, ITU). The storage space is divided into BUCKETS. A HASH FUNCTION computes a bucket number for each key value. Name(GTU,ITU,ODTU) enters to the function. The result chooses immediately input. GTU’s address is here, ITU’s address is here etc. so, depending on what you provided, the function is executed, and then, location is determined. Rather than searching. That’s it. That is no more to understand. [ HASHING THE KEY FIELD VALUE 25X3Z to one of 41 buckets slaytına gecıyor ve orada complex bir örneği açıklıyor, HASH COLLISION diye bir terimden bahsetti. Hash fonksiyonunda asallar önemlidir, çakışma olmaması için kullanılır.Kriptolojide asal sayılar önemli. ((31.05 – 37.53 dakikaları arası bu örnekten bahsetti.)) ]. [ THE RUDIMENTS OF A HASHING SYSTEM slaytına geçiyor]. That place contains three record when divided by 41. What am I going to do? One strategy is to allow data base to contain maximum three fields but what happens if 4 is coming? (hani her harfin olduğu koltuğa 3 sandalye koyduk belki biraz kalabalık olur aynı yere düşer diye, 3 amca geldi oturdu. 4.gelirse? her zaman bir B planına düşüyorsun. We call it as COLLISION. Generally, when table is becoming full, collision is important, even if it is not full, different values can map into same hash location. One solution is increase size general data base pocket size,which is easy solution. Second, change the function, when you change the function . When you change the function releases are shafled. (hani dedik ya ali , veli sit here bc of my function in my brain, if there are two many people change my function get up everybody I have changed my mind now im letting you sit based on nation identity number.those who start with 1 start here , with 2 start there. If there are a lot of muhammet in the room, they can sit in different places. ( 40.31 -45.05 arası çizerek neden o sayı 41’di ve çakışma durumlarını pointerla nasıl çözülebildiğini açıklıyor. Dolu olan adres bloğuna ikinci elemanı da yazabilmek için onun hemen altına bir pointer koyuluyor. O pointer da ikinci elemanı gösteriyor.). [DATA MINING SLAYTINA GECIYOR.] . There is keyword named as data mining, since this is an introductory course, we have data mining class in 4.sınıf selected course, you can take that. In data mining, why do we say that it is mining? There is data base and data mining. Something special with mining. Special thing with mining is, get rid of dirt and soil. Gold is very limited. In order to get some gold, you need to destroy mountain,forest. It is mining. Attracting valueable things among unnececessary unvaluable items. When we talk about data base, all of the data is probably the data I need. But When I m doing data mining, that data is not visible yet, it is in a lot of different data and needs to be cleared like gold. And rest of the information is not needed and irrelevant.That’s why executing valuable information and knowledge from big data chunks for information is called as data mining. Those data to be mined is stored in certain database system.It is called as DATA WAREHOUSE. Data warehouse and database are different. Database is dynamic. You can update , insert, delete. New record can be added modified. But in data warehouse, it is static. No change. Like a gold mine. How did gold formed?.... . But in order to mine, no change in amount of gold and no change in locations of gold, what you do is extracting the gold, so that is data mining, gold mining. That is what mining is. From the static information, when you are mining for exmp, for bitcoin, what do you mine? Mesela 6421 no’lu adreste Ahmet adı varsa, Ahmet yazarsan 6421’e ulaşırsın.Ama 6421’de ne var? ne olursa 6421’e gelebilir? Sorusunun cevabı yok.Bunu iterasyonla bulmaya çalıştığın zaman mining yapmış oluyorsun. Adreslere tek tek giderek içi dolu mu boş mu diye bakıyor bitcoin minigcileri. Önceden çoğu boştu ve boş alan bulmak kolaydı ama git gide zorlaşıyor çünkü artık dolu dolu dolu oluyor. Hashing bitcoine örnek oldu iyi oldu(?). [ 54.50’den 58.22’ye kadar tekrar çizerek anlatıyor. Hashing sistemini(?).]