High Performance Computing

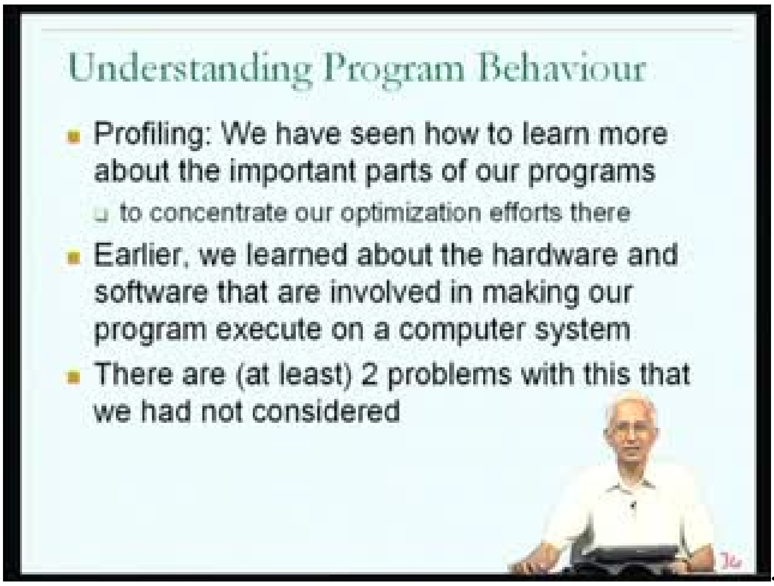
Prof. Matthew Jacob
Department of Computer Science and Automation
Indian Institute of Science, Bangalore

Module No. # 08

Lecture No. # 34

Welcome to lecture thirty four of the course on High Performance Computing. You will recall that, in the previous lecture, we looked in some detail at different techniques for identifying the important parts of our program, using tools called profilers. And we move into a new topic in today's lecture. But let me just recap quickly, to give you a some perspective on where we are right now.

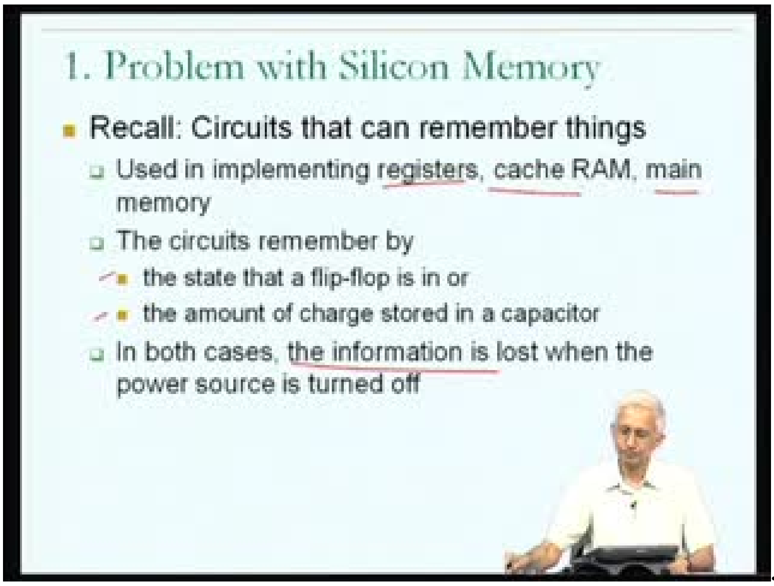
(Refer Slide Time: 00:35)



The objective of the various things that we are hearing about in this course is for us to understand, what happens in the hardware and in the software, when we run a program on a modern computer system. And in the previous class, we learned about profiling- a technique through which, we can learn more about, which are the important parts of our program from the perspective of concentrating or optimization efforts, possibly only on those parts; may be a small number of functions or small regions of the program. For example, important loops of the program, which we might identify using basic block profiling of some kind.

Now, in the previous topics of this course, we have looked at the hardware and the software, through which the execution of programs is enabled on a computer system. And in fact, at the time that we were looking at the different aspects of architecture like the hardware which is involved in the execution of instructions or the hardware which is involved in remembering, in other words, the memory and the cache, as well as this operating system and the software side of things, there were in fact, couple of problems which we did not address, but which we will now have to take into account, in order to progress further in our understanding of what happens when a program executes. I am going to desperately talk about, at least two of the problems that we have not considered in the scheme of things that we were discussing until now.

(Refer Slide Time: 01:59)



Now, one problem that, in fact, we did not talk about at all, when we talked about hardware, was a problem with the kinds of memory that we are assuming, I used in the construction of a computer systems. In particular, the silicon memory. Now, you will recall that the silicon ,what I mean by the silicon memory is the memory circuits through which main memory is created or through which the cache memory is created and so on. And so, we are talking here about circuits; electronic circuits that are capable of remembering things. And as we saw they are, these are the circuits that are used to implement the registers within the CPU, whether they be general purpose registers or special purpose registers, they are also the circuits that are used to implement the RAM of the cache, the capacity to remember blocks of data within the cache, and further they are the kinds of circuits that are use in the construction of main memories.

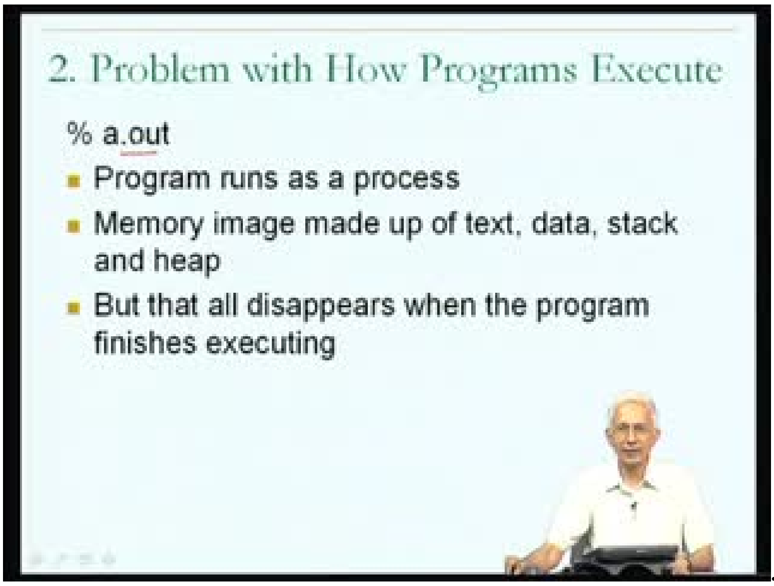
Now, you will recall that, we describe the way these circuits operate, indicated that there are two main kinds of circuits, and they defer in how they remember things; the one kind of circuit remembers by the state that the circuit is in, and at one point, I had used the word flip-flop to refer to one possible implementation of a circuit to remember one bit of information.

So, the state that the circuit is in, would determine whether a 0 or a 1 is being remembered by the circuit. The other possibility was that, a circuit could remember by the amount of charge, electric charge that is stored on a capacitor, which is part of the circuit. The more the charge, possibly that could be interpreted as remembering a 1, less charge could be interpreted by the hardware as remembering a 0.

So, these are the two main ways in which the circuits remember, either by the state the circuit is in or by the amount of charge that is stored on a capacitor, which is part of the circuit. Now, in both of these cases, the question that we did not ask was, what happens when the circuit is turned off or the computer system is turned off. It does not happens in both cases, the information is lost when the power source is turned off. So, when you power down a computer, the contents of the registers are lost.

The contents, all the data are instructions which have been remembered in the cache RAM are lost, and whether contents in the main memory are lost. When next, you turn on the same computer; the contents that were there at the time of power down will no longer be present. That is what I mean by saying that the information is lost. So, this is our problem which we had not talked about, because until now, we did not concern us. But as we are going to see, we are now going to start being concerned about this. This is the problem with the kind of silicon memory that we are assuming; the computer system uses to build its registers, cache and main memory. So, that is one aspect of the problem. In other aspect of the problem is the whole idea of how we expect programs to execute on a computer system.

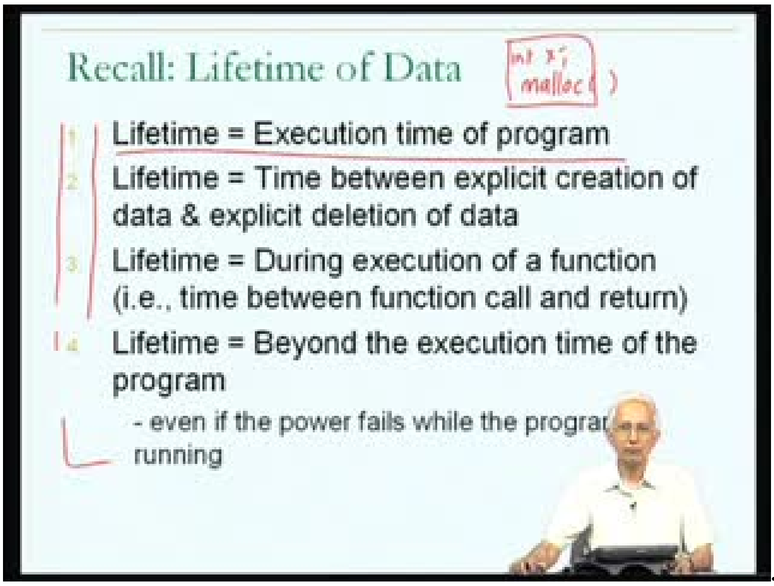
(Refer Slide Time: 04:36)



So, we understand that, when we type a dot out, in response to the shell prompt, the program will be loaded into memory and then will execute. And when the program is in memory, it runs as a process to the instructions of the processes an operating system abstraction, but when there is an active process which is running, instructions of that process are executed on the CPU and the data of that process may enter the registers of the CPU and get manipulated and so on. But, the main state information about the process is actually stored in memory, and we talked about the memory image of the process is being made up of its text, which is the instructions comprising the program of which the process is in execution, the data, the stack and the heap. The different kinds of data with different life times that are associated with the execution of the program that is represented by that process. So, the memory image of the process was in integral attribute of the program in execution. When the values of variables change, they are reflected in the change in the values of the variables within that memory image.

Now, once again, the question that we did not ask is, what happens when the program finishes execution. In other words, the a dot out program has finished execution, and it had computed various pieces of information, various kinds of data and the data could be in the text, I am I am sorry, the data could be in the data segment or the stack or the heap. But when the program finishes execution, certainly all the data seizes to exist. So, all the instructions, the data, the stack and the heap, which were in memory when the program was executing, disappear when the program finishes executing.

(Refer Slide Time: 06:12)



And this requires us to relook at our discussion of the lifetime of data. Because very clearly, the execution models that we have been talking about up till now, are leaving something out. So, let me just remind you that, when we talked about the life time of data, we looked at three possibilities- one was that, there could be pieces of data which came into existence when the program starts execution, and it seizes to exist when the program finishes execution. And that is why we would talk about the lifetime of such a piece of data as the execution time of the program. And you could imagine that, if you had program in which there was a global variable x which was declared, then this variable x would come into existence when the program starts executing and it seize to exist when the program finishes executing. And this variable x might be implemented in the data segment, the data part of the memory image of the program of the representing the process.

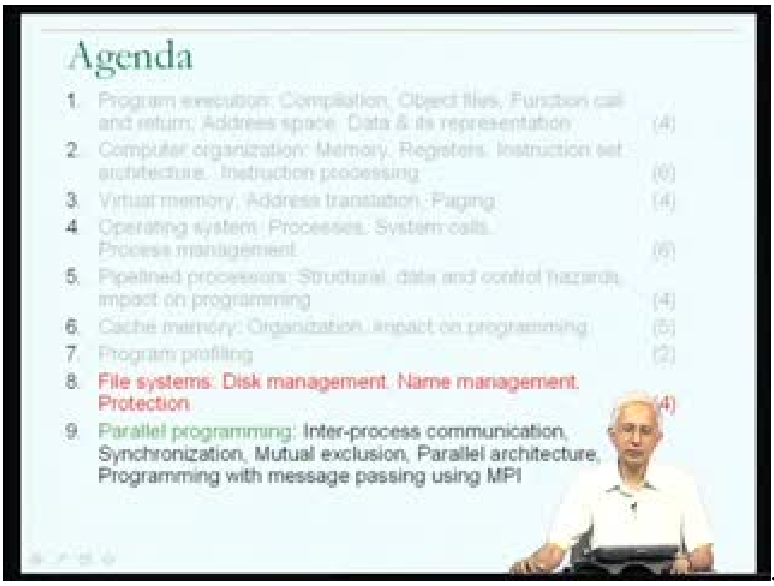
So, this is one possibility about life time. The second possibility about life time that we saw was there could be pieces of data, whose life time was, the time between the explicit creation and their explicit deletion. And we know here that, by talking about dynamically created pieces of data, and they may be, they may come into existence because, the program executes a function such as a malloc, memory alloc through which, a variable is created in memory, in the heap region of memory and is associated with the pointer that points at it through which the program can refer to that piece of data.

So, this was the second possibility and relates to dynamically allocated variables which would be handled in the heap part of the memory image. The third possibility of life time that we saw was that, there could be a piece of data which comes into existence when a function is called and seizes to exist when the function recall returns. And we saw that, such variables would be, could be local variables of the function, they could be the parameters of the function, could even be the return value of the function, depending on how tightly you look at this definition of life time. But in all these cases, it would be the model that we understood was one way, these data with this kind of a life time would be allocated on the stack region of memory and therefore, the stack would grow and shrink as functions call and return.

So, all three of these, and these are the only three life times where data that we talked about. In all three cases, the best that one could expect is that, the lifetime of piece of data could be entire execution time of the program. This relates back to the previous slide that we had, where I had explicitly said that, this is a problem, this seems to be restrictive. You cannot have data that survives after the program finishes executing. And the lifetimes that we had seen up to now, would clearly of that kind, the largest possible life time was an execution time of a program. We clearly need to be talking about a longer life time than this because, there must be situations where we want to have data that exists after the program has seized, has finished execution.

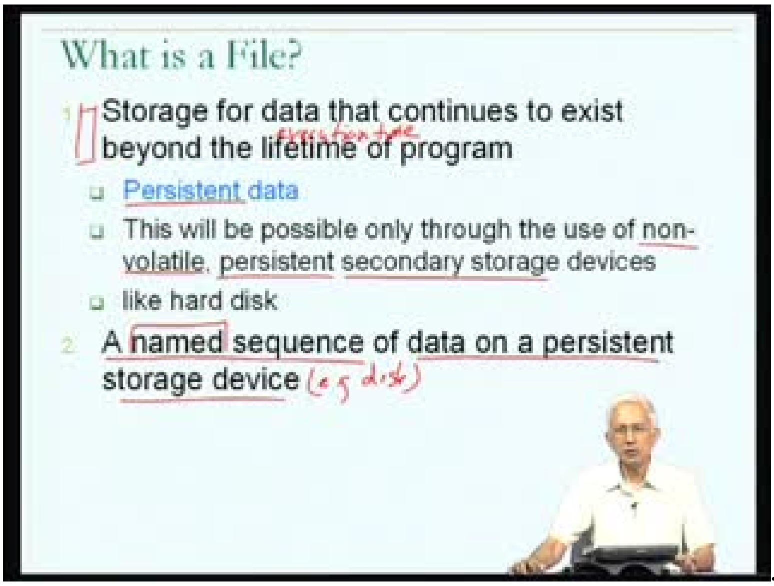
So, I am going to add a four possible life time and the fourth possible life time is, of data which continues to exist beyond the execution time of the program. So, the data may have existed before the program started executing, the data may have been used by the program meant was executing, but the data may continue to exist even after the program has finished executing .And in fact, I will go even further and say, the data might even continue to exist if power fails while the program is running. This is an even stronger constraint. You will note that, talking about the program or the data continuing to exist when the program has finished executing is one thing, but you talk about data which might continue to exist even if, by accident, the program does not finish executing completely, but rather the computer goes down when the program is in execution is even stronger requirement; which is very clearly not going to be possible, if we are talking about using the kinds of silicon memory that we were talking about up to now.

(Refer Slide Time: 10:22)



So, for the first time, we are talking about a class of data whose life time requires that something other than silicon memory be used, for this storage of such data; which is whole new class of data, which is why we have clearly moving into a new topic and the name of the topic is item number eight in our agenda is what is refer to as file systems.

(Refer Slide Time: 10:30)



So, we have all used we have all used files. I have even referred to the file system calls when we talked about system calls. And we can, we all have a rough idea what a file is, but let me just go ahead and say, the file is the storage associated with data that continues to exist beyond the lifetime of the program that may have created it or the life time of a program that may have used that particular piece of data.

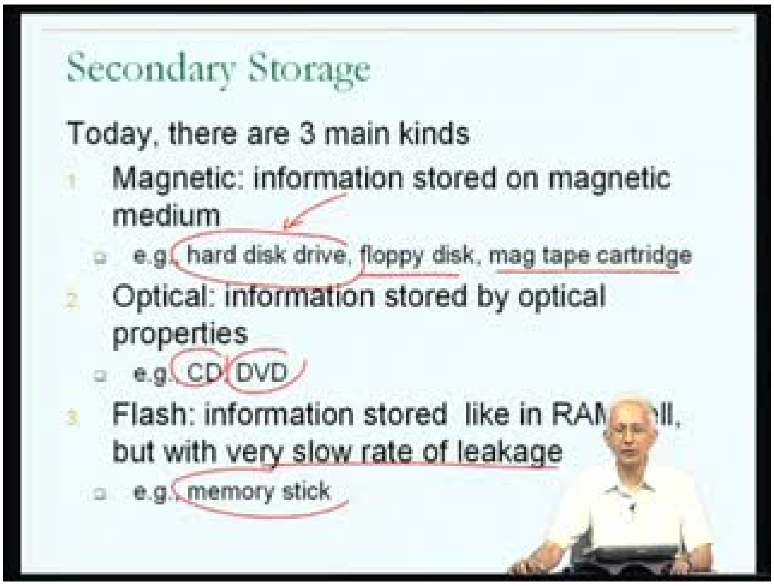
So, the file directly addresses the problems that we had identified in the previous few slides. So, files essentially exist to allow data to live beyond the life time of a program. And what we meant by the life time of a program is the execution time of the program. So, the data will continue exist after the program has finished executing. Now another word or a word which is relevant in connection with such data is to refer to the data as persistent, in the data continues to exist or persists beyond the execution time of the program that may have created that piece of data. In order to have such persistent data, we have very clearly, going to have something other than silicon memory; some form of storage which continues to be able to store or to remember things beyond the execution time of a single program. And if you think about the different kinds of storage that we saw, the only form of storage which has this property is what we called secondary storage. We talked about main memory which was primary storage. We talked about various other silicon memory, but here, very clearly, we have to be talking about non silicon memory which was, in general, what we were including in our class of secondary storage devices; which we, in some sense where including under the I O devices in our list of hardware.

Now I will introduce another term in addition to persistent in i i identify in talking about secondary storage devices. And that is to talk of them as non-volatile. And that too has this implication of volatility- is the implication of seizing to exist or being temporary in nature. Non-volatile has implication of being something which has a more long lasting or persistent nature. So, these terms non-volatile in persistent would both characterize the kind of storage devices, they clearly must be used for the storage of data that has to exist beyond the life time of a program, the execution time of a program.

So, in other words, I could actually define or explain what a file is, as an alternative to describing a file as the storage for data that continues to exist beyond the execution time of a program; knowing that such data could only be stored on a secondary storage device or a non-volatile persistent storage device, I could say that; and one example of such storage device, the hard disk is an example. But, I could then go ahead and say that, a file is collection of data on such a persistent storage device.

So, I could, in answer to the question what is file, an alternative answer, the answer number two might be, a collection of data on a persistent storage device. I have added one more requirement of a file and that is, that a file must have a name. Because it is only if the file, if this data which is stored on the persistent storage device such as the disk, it is only if that collection of data has a name, then it can be referred to by a program. Remember, we are talking here about data which can be used by a program. Hence I have added the requirement that, a file must have a name associated with that. In the absence of a name, a program would not be able to refer to the data on the disk.

(Refer Slide Time: 14:04)



So, we have this understanding what a file is. And let us just take another look at secondary storage. We are going to look at secondary storage in a little bit more detail. Now today, there are three main kinds of secondary storage. The first kind of secondary storage is persistent a non-volatile, but the way that it stores information is through properties of magnetic media. So, these are typically referred to as magnetic storage devices. The second kind of secondary storage device remembers or stores information by the optical properties of certain devices and hence they are referred to as optical storage or optical secondary storage devices. And the third, which I do not know if I have talked about in much detail before, is what is known as flash secondary storage. And here, what we actually have is something which is a form of silicon memory, but special in that, even though it is a silicon memory, it continues to be able to retain the information that is stored in it, well beyond the power source has been disconnected.

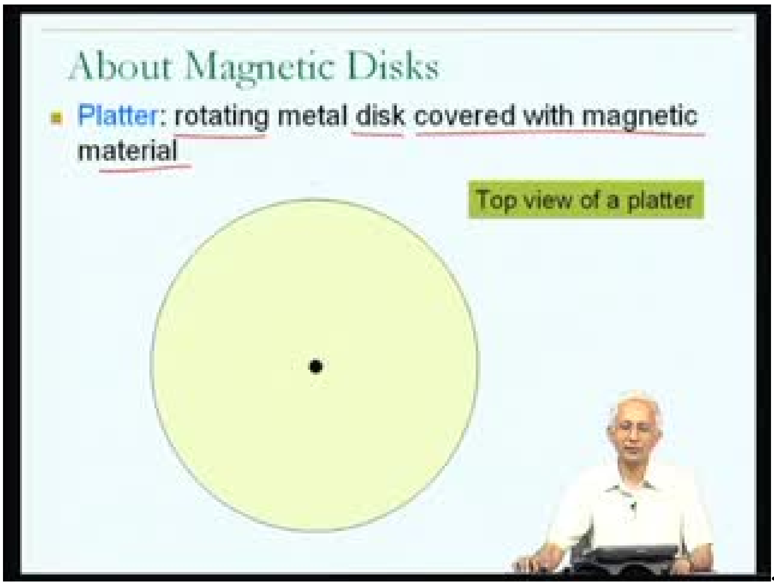
And that is because; it has a very slow rate of leakage of the charge or of the information that is stored in the silicon memory.

So, flash is something like the silicon memory that we talked about, but special and that it is very carefully designed so that, the information dissipates from it or leaks the way from it at extremely slow rate, when the power source is switched off. And therefore, one can still think of the flash memory as being persistent or non-volatile. The information continues to exist after the power source has been removed from the flash device. And we saw examples of magnetic storage. I talked about the hard disk drive ,every computer system, most computer systems that you have dealt with, have the hard disk drive in the not too addition pass very popular form of storage was the floppy disk.

One would carry floppy disk containing ones program or data inserted into the floppy disk drive in the computer system from which information could be read, files could be read. And even today, people sometimes use magnetic tape cartridges, which are long reels of magnetic material on a tape. So, once again, the storage is done by the magnetic state of the magnetic information, the state of magnetic information stored on the device in all three cases .We have also seen some examples of optical secondary storage device. The compact disk, the VCD, the DVD ; these are all forms of the, these are all secondary storage devices, non-volatile persistent, in which the information is stored through some optical property of the medium. Finally, flash also, something you would have encountered before; many of you use memory sticks, many of you use MP three players which contain no hard disk, the MP three player that you have, clearly does not have a hard disk or an optical storage device, but rather it remembers the music or other pieces of information that you have stored on it on a flash secondary storage device.

So, these are all examples of things that we have seen before. And in fact, all or anyone of these could be used for the storage of files. Now, as it happens, for the most of the computer systems that you will be using today, the file system would typically be, at least one of the file systems that you have access to, would typically be on a hard disk at a certainly possible that you could have files stored on floppy disk, magnetic tapes cartridges, and you could certainly have CDs and DVDs and VCDs containing files that you could insert into the system and read the files from and the same is two of memory sticks, but, an important part is played by the file systems which are provided on hard disks in the computer, many of the computer systems that you will deal with today. Both desktops and many of the laptops still use hard disks. So, I am going to concentrate a little bit more and talk about the characteristics of a disk drive, a hard disk.

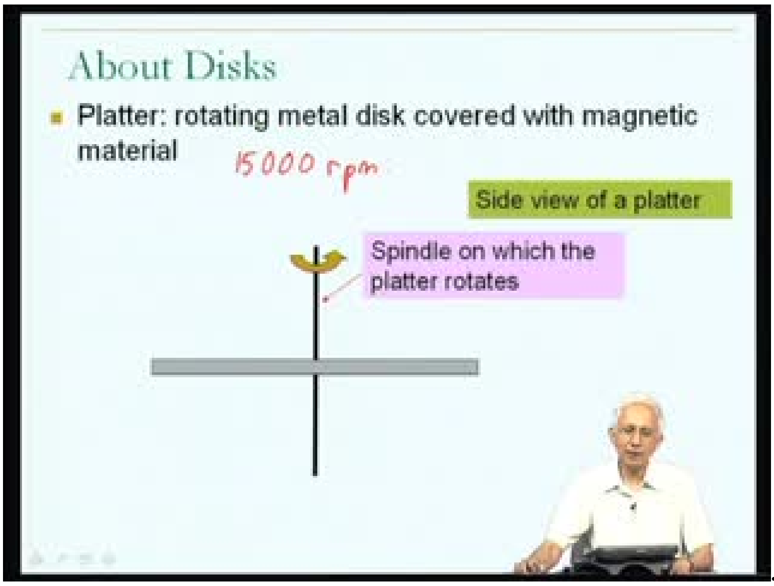
(Refer Slide Time: 18:03)



So, going to look at the properties of magnetic hard disks. Now, if one was to open a hard disk and look inside, one would see that the magnetic material is actually coated on to a surface of something called a platter and this is the surface from which the word disk comes. The word disk is actually just referring to a circular, flat object and basically the magnetic disk contains such disks, and then technical term which is given for each of those disks is to refer to it as a platter.

So, there is a platter which is covered with magnetic material and the information, the data or the instructions that are to be stored in the file are actually stored in the magnetic, by the magnetic state of the coating on the platter. Now, one key piece of information about the platter is that, it is rotating and therefore, when we think about a platter, one could have, and I am, if I was viewing, I broke open a hard disk and I view the platter from the top, what I would see would be a circular disk. And I am showing you the circular disk over here, we are looking at it from the top. Now, it is rotating therefore, would also be beneficial for us to look at from the side, but for the moment, we just think of it as a hard disk. I am looking at the platter from the top.

(Refer Slide Time: 19:31)



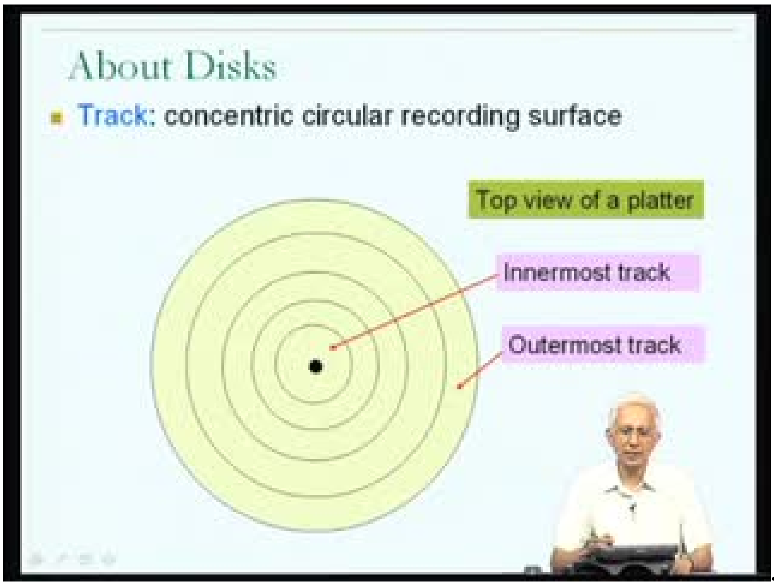
So, this is the platter and it has a magnetic coating on it, which is what I am showing by this yellow color. Now, if I look at it from the side, this once again is the hard disk, the platter looked at from the side. Recall that it is rotating, which means that, it must be rotating around its…, I had showed you the center of the of the disk.

So, very clearly, when one looks at it from the side and if this disk is rotating, if this platter is rotating, it must be rotating on a spindle on an axes of some kind. And that is what I show in this diagram. We are looking at the platter from the side now. And I am also showing you the spindle on which the disk is rotating.

So, this spindle on which is rotating and I am also showing the direction in which it is rotating. Now the typical disk today, it might be interesting to note how fast the disk is rotating. And at this point, it is not clear why the disk is rotating at all, from the discussion that you had so far. But let me just mention that a typical disk today, the speed of rotation of the disk could be something like 15000 revolutions per minute.

Some disks may operate at 10000 revolutions per minute. But there are disk today which rotate at 15000 revolutions per minute. So, in every minute, they are rotating the 15000 times, which means these are rotating at a fairly high speed right. And in order to do this rotation, as we had learnt earlier, they must have motors, there are many mechanical devices in the hard disk, unlike the silicon memories in the CPUs that we saw up to now, where there were no mechanical devices. The hard disk certainly has many mechanical devices such as the motor, motor is electrical in nature, but it does this movement which is why I refer to it as a mechanical device.

(Refer Slide Time: 21:08)

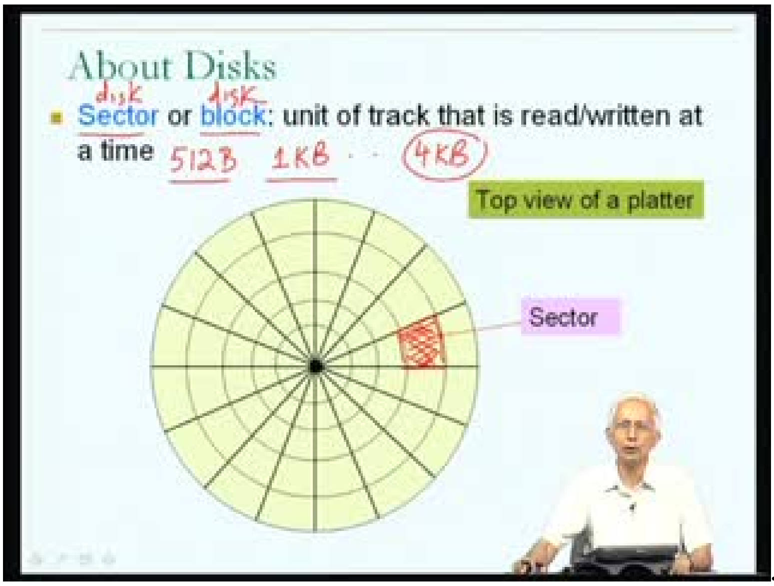


Now, so much for the platter. Now, we know that the, I am once again showing you the top view of a platter. We are looking at the inside, what is inside a hard disk. So, the platter is coated with magnetic material, but magnetic material is not just painted on top, but rather has some structure to it. And one can talk about the different concentric coatings of magnetic material across the surface of the platter by the word, track.

So, a track is one concentric circular recording surface on a platter. For example, here I am showing you the outer most track of this particular platter. So, there is one track of information and there could be tracks of information internal to that. In this particular example, I am showing you about five tracks along, on this on this particular surface, for a real describe today, the number of tracks could be tens of thousands, twenty- thousand, thirty -thousand or forty- thousand tracks.

So, the number of concentric circles of magnetic material on a single platter is on the order of a few tens of thousands. I am unable to draw a ten-thousand or thirty-thousand concentric circles on this. I am just showing you five, for purposes of example. So, we understand that the word track is referring to anyone of these concentric coatings of magnetic material. So, I could talk about the innermost track of this platter. And I could refer to the outermost track of this platter. And you could observe that, the number, the amount of information that could be stored on the outermost track is likely to be more than the amount of information that could be stored on the innermost track. Because, the innermost track has less area than the outermost track, if one make assumptions. But, one could also imagine that there could be some variations on how the layout of the disk is done, in addition to this basic framework.

(Refer Slide Time: 23:05)



Now, the next thing that we need to understand is that, each of the tracks on a particular platter of a hard disk contains units of information that can be read from the hard disk. And the basic unit of information that can be read or written from the hard disk is not one entire track, but a unit portion of a track which is referred to as a sector or a block. So, a sector or block is a unit of track that is read or written at a time from the disk. So, for ease of illustration, I am going to show you sectors as being radially oriented on this, on the surface of the disk. So, for example, this region of that particular, the second track from the outside on this disk, I have shaded a particular region of the track which might be what I will refer to as a sector.

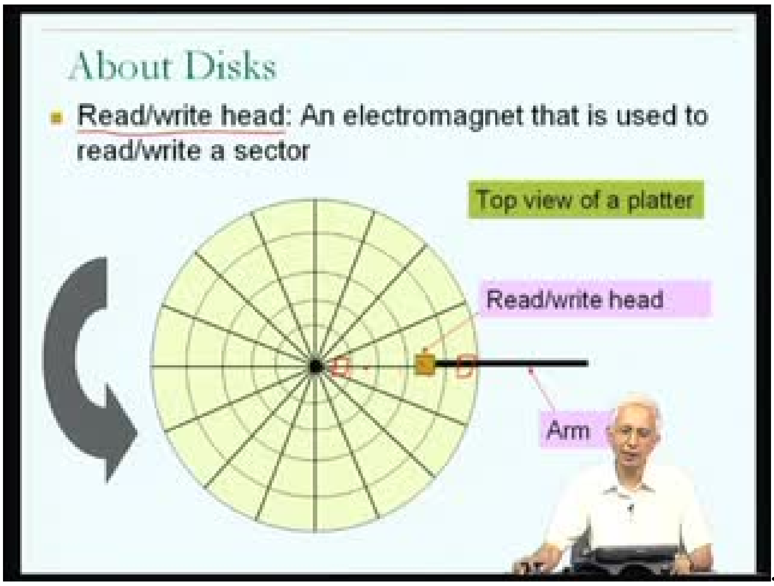
So, you will notice that, on any one track, there are many sectors. And the sector is the basic unit in which one can read or write from hard disk. Now, I have used the words sector, I have also used the word block; and currently the word sector is more often used to refer to this unit of read or write from a hard disk. People talk about a disk sector. But it is still the cases that, people will, some time refer to this by the word block. I have used the word block in connection with cache memories. Therefore, it would be a little bit confusing, but in the current discussion, when we are talking about file systems, whenever we hear the word block, just understand that, it is meant to be synonymous with the word sector. And if necessary, we can qualify by talking about this as a disk block or a disk sector. Now, the question that will come to mind at this point is, I have told you that the number of tracks on a typical disk today, is a few tens of thousands. But the basic unit of reading or writing information from the disk is not a track; it is on the other hand, a disk sector.

So, you will ask, how big is a sector on a typical disk is today. And the number which we could use in connection with this could be something like 512 bytes. Until not too long ago, 512 bytes was a typical disk sector size. In some cases, it may have been about 1 kilobyte, but increasingly, it has been the case that the size- a basic unit of information that can be read of a hard disk. The benefit of having slightly larger sectors is that, from a single read from the hard disk, one can get more information if the sector size is more.

So, increasingly the trend has been towards having sector sizes that are larger than 512 bytes. In fact, there is a move to have some kind of a standardization through which 4 kilobytes will become a typical sector size in a year or two. But for the purposes of our discussion of file systems today, I will use examples of something like 512 bytes or may be 1 kilobyte as a typical disk sector size.

So, essentially we understand that, in a single read from the disk, one can get one sector of information. So much for our understanding of the disk, we understand that the disk, if you break it open, has information which is stored on metallic, flat metallic surfaces, circular surfaces called platters. The platter is coated with magnetic material in concentric circles called tracks and each track is broken up into individual units called sectors, where a sector or a block is the basic unit of read or write from the disk.

(Refer Slide Time: 26:41)

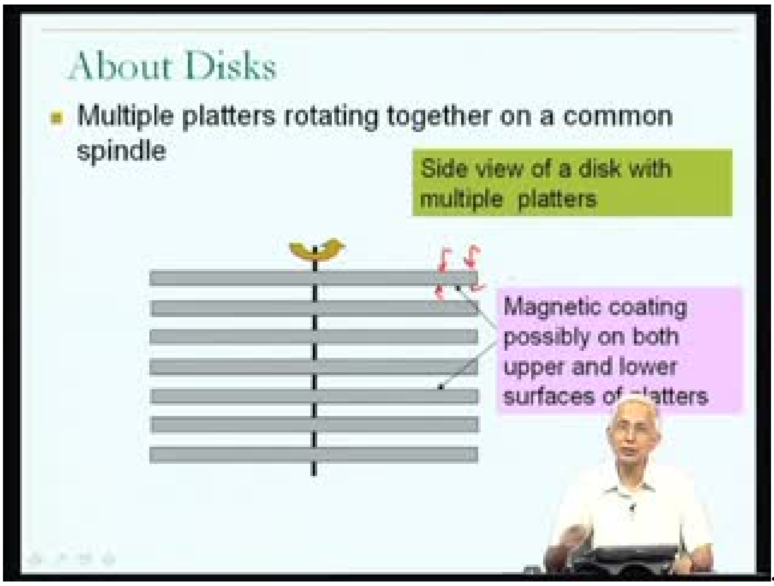


Now, in order to do the read or the write, there will be a need for a device which can read or write the current contents of a particular sector. And the name given to the piece of mechanism that can do the reading or writing from a sector is to refer to it as a read/write head. So, what is the read/write head? Very clearly, it must be a magnetic or electromagnetic device and in fact, it is an electromagnet, they can be, that can either read or modify a sector from a hard disk.

So, associated with the hard disk, there must be this read/write heads, through which one can read/write information to or from the hard disk. So, let us suppose that I have, for this particular sector; I have one particular read/write head which is currently located that… This brown object, which is currently over the platter. Then, the thing to remember here is, this platter is rotating. Remember, that is rotating on its axis, which is perpendicular to, we are looking at, we are currently looking at the platter from the top. And if I had looked at from the side, you would have noticed the platter striking up, but now, this platter is rotating in some particular direction. For example, it could be rotating in the clockwise or the counter clockwise direction. But the read/write head, if there is only one read/write head for this entire platter, then there will be a need for this read/write head to be able to move in and out and therefore, one would expect that, it will be attached to some kind of a mechanical device, which I will refer to as the arm. And that the arm can be cause to move in or out, so that the read/write head could be placed over anyone of the different tracks associated with the hard disk. So, if I move the read/write head into its innermost position, then it could be used to read from the innermost track. If I move the read/write head to the outermost position, then it could be used to read from the outermost track and couple with the fact that, this platter is rotating. Remember that, the platter is rotating. In this particular example, I am assuming the platter is rotating in the counter clockwise direction. Then it is enough if the arm can just move in and out. There is no need for the arm to have to move radially. It just has, I am sorry, it just has to move in and out along this one radius. It does not have to move up or down.

So, the arm has to move in or out to be over the correct track and then with rotation of the disk, sooner or later, the correct track will come under the read/write, that correct sector will come under the read/write arm, read/write head. So, in units of our sector, in other words, 512 bytes let us say, information can be read off the disk or could be written into the disk, through this device called the read/write arm, read/write head which is known to on a arm.

(Refer Slide Time: 29:24)

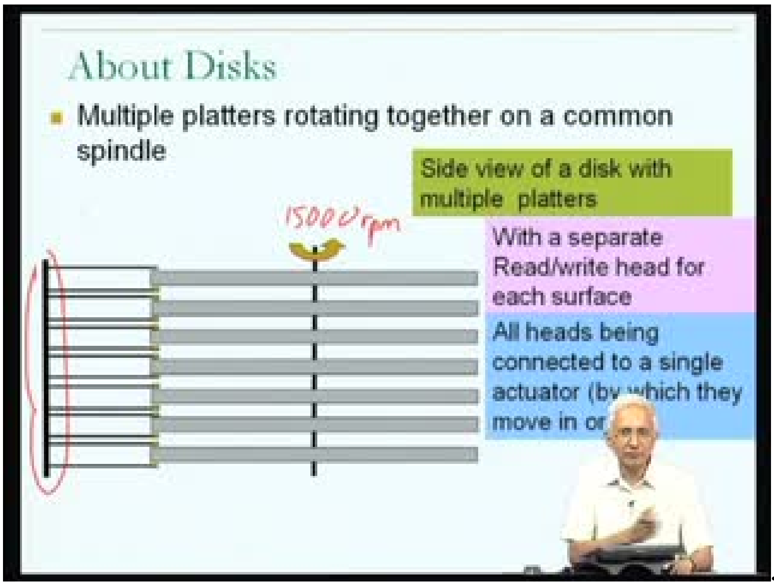


Now, the next variable that we have to understand is that, a typical disk today does not contain only one platter, but actually contains multiple platters, all rotating together on a common spindle. In other words, the picture that we had in mind up to now was that, there was a spindle which is rotating and attached to that is the platter, which we are now looking at from the side view. But in reality, the disks of today, even the disk in your personal computer or your laptop, has multiple platters.

So, there could be, as in this example, where we have seven platters, all rotating on a single spindle. What is the advantage of doing this? The amount of information that can be stored on this disk is now, many times what would have been possible with only one platter. Further, it is possible that, we could have magnetic coatings not only on the top surface of each platter, but also on the bottom surface of each platter.

So, rather than thinking of the capacity of a disk has been determined by only the number of platters, in terms of one surface for platter, we understand clearly that, there could be two magnetic coatings on each platter. What does it mean, if you have two magnetic coatings on each platter? So, there is a magnetic coating on the bottom of this platter, there is also magnetic coating on the on the top of this platter, there is also magnetic coating on the bottom of the platter. But very clearly, I will need a separate read/write head for each of the surfaces. So, there will be a need, not for one read/write head for each platter, but for one read/write head for each surface.

(Refer Slide Time: 31:04)



So, I need one read/write head for the upper surface top most platter. Another one for the lower surface of the top most platter and so on. So, associated with such a disk, in which there are multiple platters, there would have to be a separate read/write head for each surface, not for each platter. And on the left over here, I have shown you, for this example, how many read/write heads might be needed. You notice that, there is one for the uppermost surface, one for the lowermost surface of the upper platter. Similarly, for each of the platters, there are two read/write heads for a total of fourteen read/write heads. I am indicating the read/write heads by the small brown boxes towards the outermost side of the platters.

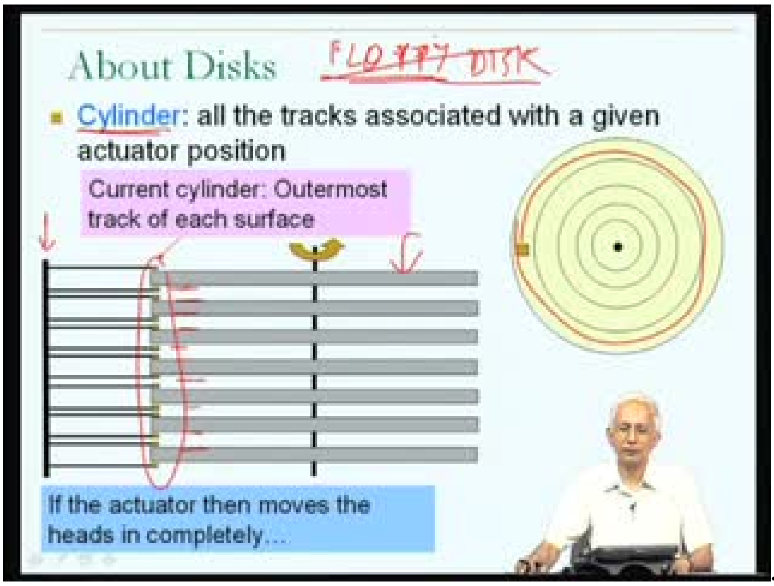
Now, you will also notice, if you look very carefully at this diagram that, I am showing the read/write heads as not being in contact with the platter. But slightly above the platter. And this may be an important consideration, you will remember that the platters are rotating at fairly high speeds, possibly 1500 revolutions per minute and therefore, if the read/write head was actually in contact with the platter, very soon the magnetic coating would gets scraped off.

So, obviously, there would be a need to have a sufficient gap between the read/write head and the platter, so that, the reading or writing can happen through the electromagnetic properties, but it should not wear away this magnetic coating of the disk. Hence, I show a small gap.

Now, we had talked in the, when we first learnt about the read/write head, I talked about the need to mount the read/write head on in arm which could move in or out, to allow the reading or writing to be done for any of the tracks of a particular platter. Now, we now have a need to have one arm for each of these fourteen in this example, read/write heads. So, I show fourteen of those arms and this raises a few possibilities, it could be possible to have a disk in which each of the arms could independently move in or out.

But that would make the design of the disks somewhat more complicated than if I had a single movement of all the read/write arms together, which is what I am going to assume in this case. So, I am going to assume that, all the heads are connected to a single actuator. In other words, all the read/write arms or actually connected together, as I show in the leftmost side. So, this connection, among all the read/write arms, is what I am going to refer to as something that is moved in or out by a motor, which of what the actuator term comes in. So, the general idea here is going to be that, we will assume that all of the read/write arms move in or out together. Therefore, there is only a single motor involved in that motion. And the alternative would have been at the complication of a separate motor and separate control for each of the read/write arms for each of the surfaces, which would have added to the complexity; therefore, the power consumption and the noise made by the hard disk and so on. So, this seems like a more palatable assumption. So, this is the picture that we have in mind. If you look at the current situation of this particular disk that I am showing, the situation is that, the actuator is in such a condition that, all the read/write heads for all the surfaces are at the leftmost extent of the disk.

(Refer Slide Time: 34:09)



Now, if I had looked at this from the top rather than from the side, this is what the diagram would look like. You notice that, in the in the side view, that read/write arms are on the outer side of the disks and therefore, if one looked at the top view, and in the top view, I see only the read/write head of the uppermost surface. That is what the top view will show me. And I see that, the read/write head is over the outermost track. And if the actuator had moved in, in fact, I will now introduce the term cylinder.

So, in the current situation, you will note that the read/write head for the uppermost surface is over; its outermost track and the same will be true for each of the surfaces of this particular read/write of this particular disk.

Its read/write head for anyone of the surfaces, its read/write head will be over its outermost track. Therefore, we could actually have a term for the outermost tracks of all the surfaces within this particular hard disk. And that is what the term cylinder is used for.

So, all the tracks associated with a given actuator position, in other words, the corresponding track from each of the surfaces of all the platters within the hard disk for a given position of the actuator is what we will refer to as a cylinder. And the term cylinder makes, seems to make some sense because, if you remember that, from the side view, we are referring to a track as being a circle, then when I have this collection of circles one on top of the other, a collection of circles one on top of the other like this, when viewed from side, it is actually going to look like a cylinder. Hence the name cylinder, to refer to the collection of all the tracks from all the surfaces associated with one actuator position.

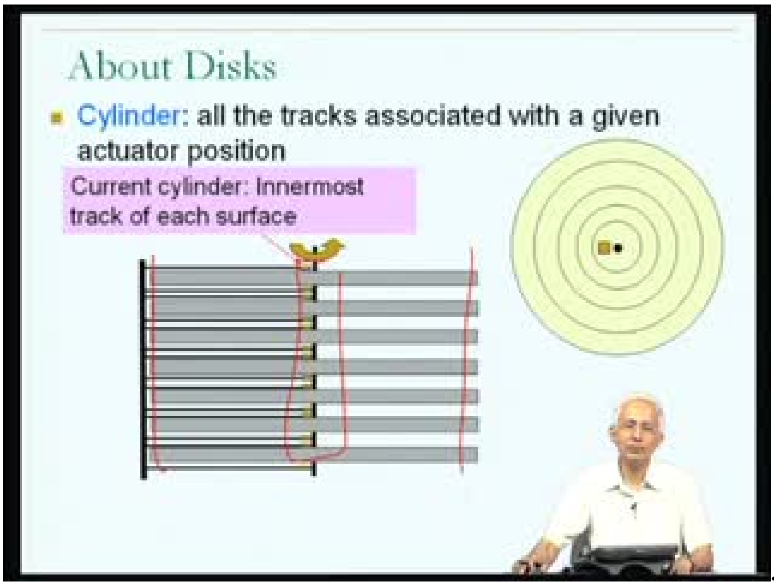
Now, just as a brief aside, here we are talking about hard disks, but one could remember, as I mentioned that there were and there still are, certain other kinds of secondary storage devices called floppy disks. Some of you may even have used floppy disks. As a name suggests, the floppy disk is floppy; in that it is not firm. We talked about the hard disk as having a platter made of metal. A floppy disk is not too dissimilar. The only difference is that its platter is not made of metal, but of some thin plastic material which is coated with magnetic magnetic coating.

So, the principle of operation of the floppy disk is similar to that of the hard disk, other than the fact that it is not a firm surface, but has a thin surface which is flyable and hence the word floppy. And in the in floppy disks too, one had this possibility of having a floppy disk which has a magnetic coating both on its upper surface and on its lower surface. But there is not really the possibility of having many floppy, many floppy platters in a single floppy drive, because of the fact that it is not a very firm surface. Therefore, one cannot have read/write heads that is, it a little away from the each of the different surfaces.

So, the floppy disk are somewhat restricted in its capacity and its capabilities, but was present in that because of its floppy nature and the low cost of the materials used to conduct to construct it. It is possible to have floppy disk available at a very low cost. So that was just aside in a floppy disk. We are currently still interested in hard disks. So, we have understood that, there are platters with multiple surface, two surfaces per platter, there are tracks and sectors and blocks where the information is stored and this concept of read/write heads and cylinders. Now moving ahead, for any particular position of the actuator, for example, in this diagram, the actuator is in such a, actuator is such that, the read/write heads are over the outermost track of each surface.

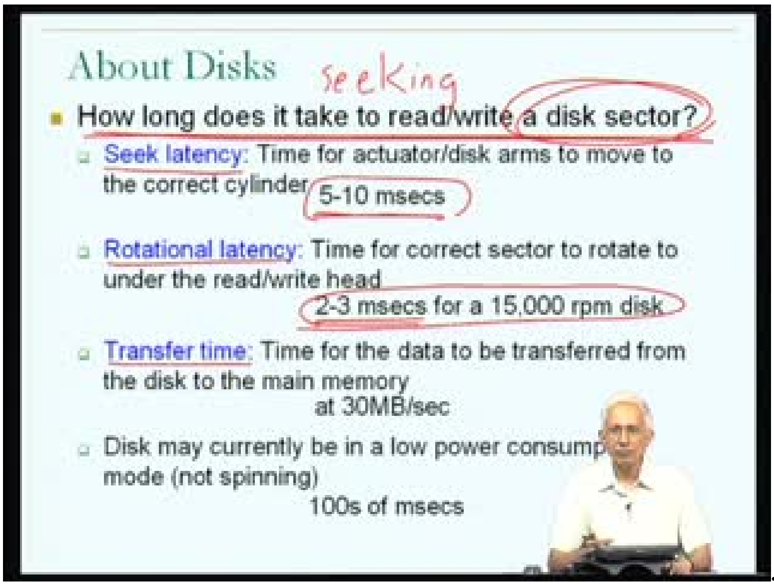
Now, what if the actuator was moved. The disk was activated so that the movements of the read/write heads went. So, all the read/write heads over the innermost track. Then, in other words, if the actuator then moves the heads in completely, then the diagram would look like this.

(Refer Slide Time: 38:11)



Remember that, in the previous situation, the read/write heads were over the edge- the outer edge of the side view. But if the actuator moves everything in, then the read/write heads will be over the innermost part, just off the spindle. And in the upper view, we would actually notice that, the each read/write head was over the innermost track and that would be referring to now as the innermost cylinder. So, I can talk about the innermost cylinder and I could talk about the outermost cylinder. So, in the current diagram, the current cylinder is the innermost track of each surface.

(Refer Slide Time: 38:51)



Now, with this understanding about what is happening inside the disk, some additional information about the floppy; we can ask some other questions about disks. One important question is and this is the question that we have approximate answers to earlier without a proper understanding of what was happening inside the hard disk. But the question of how long does it take to actually read/write a disk sector. We know that the size of a disk sector could be 512 bytes or 1 kilobyte. But that is the unit of read/write from our hard disk. And the question is, how long does it take to read or write from a hard disk of the kind that we are that we are currently talking about.

Now, given our understanding of how the hard disk is constructed, we can work out the various steps that may have to take place, in order to read information out of the hard disk. Now, how does this requirement to read information from a hard disk arise? It arises because a program made a request through which it became necessary to read from a hard disk. For example, we are talking now about file systems. So, it is possible that a program did a read operation on a file, as a result of which it became necessary to read from a hard disk. So, all of this is going to be initiated by program behavior. Now, the piece of information that is required from the hard disk is going to be available in some particular sector and therefore, there must be some notion of a sector address. So, we are talking about reading a particular sector, may be sector number one thousand of a hard disk. Now, one thing which will certainly have to happen is that the read/write arms will have to be moved into such a position So, that they are over the current, the correct cylinder as far as the sector of interest is concerned. For example, if the particular block that, this particular sector that is required by the program is on the outer most track of one particular surface of the hard disk, then one thing which will have to happen is that the read/write arms will have to be moved so that, they all are over the outermost cylinder of the hard disk. On the other hand, if the sector that is required is on the innermost track, then the read/write arms would have to be move by the actuator until it is over the innermost cylinder of the hard disk.

Now, the amount of time that it takes to move the read/write arms so that, they are positioned over the correct cylinder or track is what is referred to as the seek latency- the operation of moving the read/write arms is what is referred to as seeking and therefore, one component of the amount of time to read a sector out of the hard disk is the seek time and this is basically the time for the actuator to move the disk arms to be over the correct cylinder. And one knows the correct cylinder based on which disk sector is required by the current read write accesses.

Now, once the read/write arms are positioned over the correct cylinder, one has to of course, wait for the correct sector to come under the read/write arms. Remember, from our diagram, that is not enough for the read/write head to be over the correct cylinder. It could well be that the sector that we are interested in, is a sector which is not currently under the read/write arm. But as the disk rotates sooner or later, that particular sector will come under the read/write head. So, the second component of the amount of time that it takes to read from a disk to read a sector from a disk block is what I will refer to as Rotational latency. And this is the amount of time and that it takes for the correct sector to rotate to and become under the read/write head.

And so, this is the second component of the amount of time that it takes to access to read or write a disk sector. Now, once both of these, once the seeking has been done and the rotation has been done, then we are in a condition where that read/write head can actually read/write information to or from the disk. The next question is, how much time does it take to transfer the information that has been read by the read/write arm into the memory; which is where the data was supposed to be read to or to transfer the information from the disk into the computer that the processor. And this is what is I will refer to as the transfer time- the time for the data to be transferred from the hard disk to the computer system may be to the main memory.

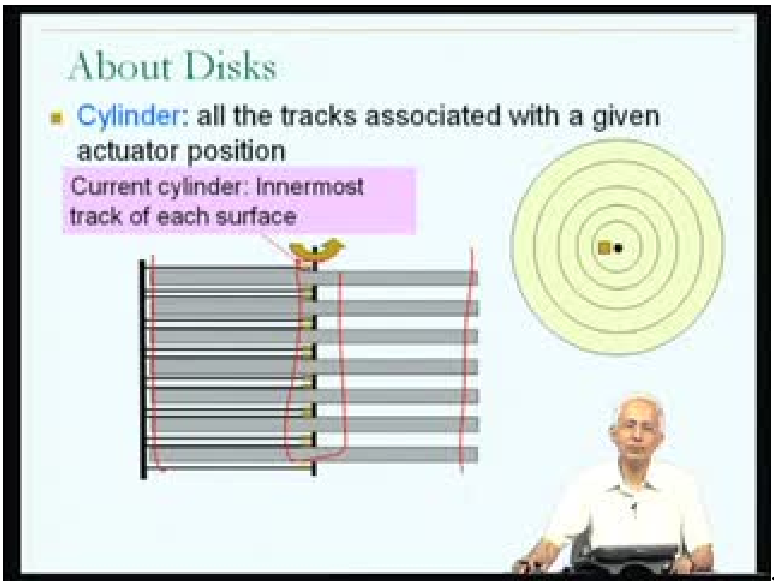
And this also has to be viewed as part of the read/write time for the disk. So, I will add it as the third component. Now, there is a fourth possible component and let me first motivate the component before putting its name up. Now, you will notice that these disks, the way we have described them, are fairly complicated electromagnetic and mechanical entities. And in a computer systems of today, where it is important to try to keep the amount of power consumed by the computer as low as possible or the amount of heat generated by the computer as low as possible in the interest of environmental reasons and keeping the cost of running the computer down, it may occasionally be necessary for the disk to be powered down. For example, if you consider that, there is a situation where there has been no disk activity for a significant amount of time, then instead of the disk continue rotate, the disk rotation could actually be seized. And subsequently if there is a need to access the disk, the disk could be rotated again.

So, in order to conserve the amount of power consumed by a disk, there may be a need to have a low power mode of operation of the disk in which possibly it is not rotating. So, if the disk is currently in a low power mode of operation and that is when the request to read/write from the disk happens, then there may be the need for some startup time, in order to get the disk rotating. So, I will add that as a possible fourth component to the amount of time that it takes to read or write from a disk sector and that I will just write it as the disk may currently be in a low power consumption mode, possibly not spinning, in which case, it may take some time for it to come back into a mode from which the read or write of the data from the disk can be done. So, these are four of the components, four main components of how much time it takes to read or write from hard disk, to read a sector from a hard disk. Next, we need to have some idea about what each of these individual components is actually means in terms of seconds or milliseconds or micro seconds.

Now, let me just put some numbers down and then will try to understand where these are coming from. Now, first of all, we have put the typical seek latency as being something in the nature of 5 to 10 milliseconds. And how do we understand this? We understand that, basically, the activity involved in seeking is moving the read/write arms from where ever the read/write arms are currently located, until they are over the correct cylinder associated with the request that has been made. The read/write, the sector that has to be read. And this will involve a motor moving the actuator mechanism through which a motor will actually have to move the read/write arms in or out. And depending on how far away the cylinder is from the current location, the required cylinder is from the current location of the read/write arms, the amount of time that it will take will differ. Therefore, when one is given a number like this, one is obviously being given an average number; not a typical number, but an average number. And the kind of average number which is usually used is a number which is given based on the amount of time that it takes to move the read/write heads by about one- third of the width, the radius of this of the cylinder.

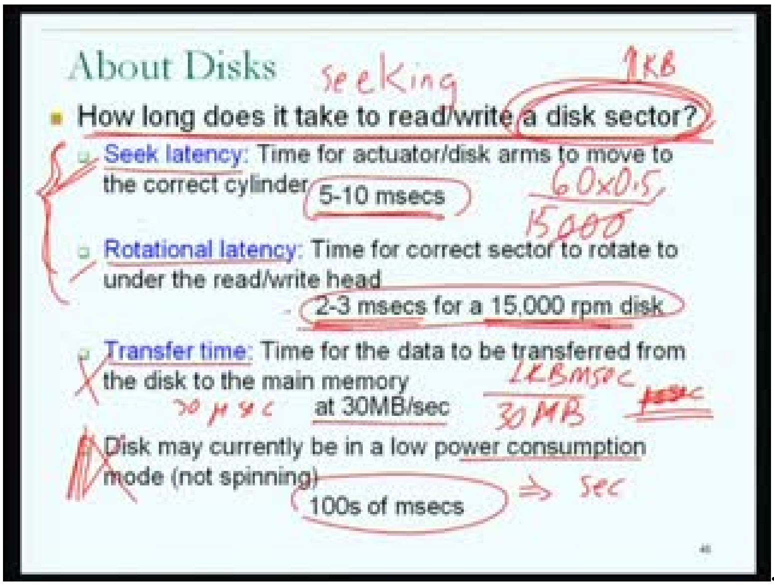
So, about one-third of the distance from the outer most to the inner most track is considered as the norm for reporting a typical seek time. So, the typical seek seek time is, we are told, are in the order of 5 to 10 milliseconds for hard disk today. And this could clearly have something to do with a number of cylinders as well as the speed with which the read/write arms can be moved in or out from either from a resting position or from a moving position.Now, the next number that we have is that, the rotational latency can be expected to be 2 to 3 milliseconds, and this number was calculated based on a 15000 rpm disk. In other words, the disk that is rotating at 15000 revolutions per minute.

(Refer Slide Time: 38:19)



.

(Refer Slide Time: 47:27)



And where could this number come from. Once again, we are clearly being given some kind of an average based on, because you will understand that, once again, going back to our diagram, when the rotation happens, it is quite possible that the disk is, the read/write head is actually, currently over the current cylinder, in which case, there will be no rotational latency. But it is possible that the read/write head is completely away from where should be, and there it has to rotate almost the full rotation, until the read/write head is over the correct sector. So, there are many possibilities. How does one come up with an average? And the answer is that, the average that is typically computed is major under the assumption that, you have to rotate about half the circumference of the platter. In other words, about half a rotation.

So, how does one calculate, how much time it takes to do half a rotation. For example, for a 15000 rpm disk, just use the definition of with rotational speed. This is a disk which is capable of revolving 15000 times per minute. Therefore, to revolve 15000 times per minute; therefore, in 60 seconds, it is able to do 15000 revolutions. And what I want to know is, how much time does it does it take to do one revolution. I am I am sorry, how much does it take to do half a revolution. Therefore, I will have to calculate half of the amount of time that it takes to do 1 revolution.

So, this was 60 seconds for 15000 revolutions. Therefore, 60 divided by 15000 seconds for 1 revolution multiplied by half to get the time for half a revolution, and in terms of milliseconds, you will notice that, this is along the ball pack of what this number that we have over here. Therefore, for a disk with a slower rotational speed, this number would be higher. For a disk with higher rotational speed, this number would be lower. But that is very clearly, the rotational latency, average number is related to the speed of rotation of the disk. What about the transfer time now. This obviously would refer to the speed with which or the rate at which information can be moved from the disk out of the disk. And we are given this number that it could happen, it is something like 50 into 30 mega bytes per second.

Now, that is the rate at which the data can be transferred. In our particular question, we are asking how much time would it take to transfer one disk sector. And for ease of calculation, if I assume that we are talking about a disk sector as being of size 1 kilobyte, then I could calculate the transfer time. I want to know, how much time it takes to transfer 1 kilobyte, if it takes 30, if I can transfer information at the rate of 30 mega bytes per second. And one can very easily see that, the the nature of the time here is going to be on the order of micro seconds because we have kilo on the top, and we have I am sorry, while it is going to be a low number in comparison with this 5 to 10 milliseconds that we have over here. But one can calculate the amount of time that it could take to transfer one this sector assuming some size for the this sector such as 1 kilobyte based on this transfer a speed.

So, the kilo and the mega are different by about 10 power 3 and therefore, if I remove the 10 power 3, I will get milliseconds over here, over one-thirtieth of millisecond, which is on the order of 30 microseconds, which is what I was talking about over here. So, ball pack we may be talking about few 10 of microseconds for the transfer. So, just going back, the seek latency on average could be 5 to 10 milliseconds, the rotational latency could be 2 to 3 milliseconds and the transfer time could be a few tens of microseconds. Therefore, in some sense, one might say that the transfer time is negligible in comparison to the seek latency. The rotational latency itself is small compare to the seek latency. And one might argue that the seek latency is what is going to dominate the read/write time for a sector.

And that therefore, one could argue that the amount of time that it takes to read or write from a hard disk, a single sector, could be something like 5 to 10 and then you add 2 to 3 to that and come up with something like 7 to 13 or 7 to 13 milliseconds, which is the kind of number that we have talked about earlier. I had loosely said that, it takes a few milliseconds to read a sector from to read, from a hard disk. Now, we have left out one component for which there is a much larger time associated with it. I have suggested that, if the disk is not spinning ,then to actually get it into a more from which the data, the seeking, the rotation and latency transfer could actually occur, may be tens or may be even hundreds of milliseconds and what this means is that, we are getting closer to a single possibly a second, as a total amount of time that you would take to do the entire disk access.

But hopefully this is not the typical case, but is going to happen really, because if you have a computer system in which the disk is fairly active, then it would really going to this low power consumption mode, and therefore, the need to get it spinning and get it initialize would not happen very often. And therefore, in talking about the typical amount of time to access a block or a sector out of disk, one could, one would be forgiven if one just ignore the last component and concentrated on the seek and the rotational latency. The transfer time as we have seen is significantly smaller.

With this, we have a good understanding about what is happening inside the hard disk and just to distil out one or two important lessons from this. We understood the structure of what is happening inside the hard disk and in effect, we realize that seek and rotation are the two important components of time in accessing our component of a hard disk. And that, if the operating system is going to play a part in trying to improve the performance of the hard disk by some kind of scheduling mechanism, then either the seek or the rotation are the two mechanisms of the disk which it may have to try to control.

Now, with this introduction to file systems, the background of what is happening inside a hard disk, we will now be in a position in the next lecture, to move ahead and try to talk about the functioning of a file system. How the files are actually located on such a hard disk, how the operating system helps to provide the files that the individual blocks or sectors of the file to a program, when the program request those the particular file, sector or block. And fundamentally, the background that we record from the hard disk has now been covered and in the lecture to come, we will look at the file system from the perspective of the operating system. The key functionality is system calls associated with that and the impact on the execution time of our programs. Thank you.