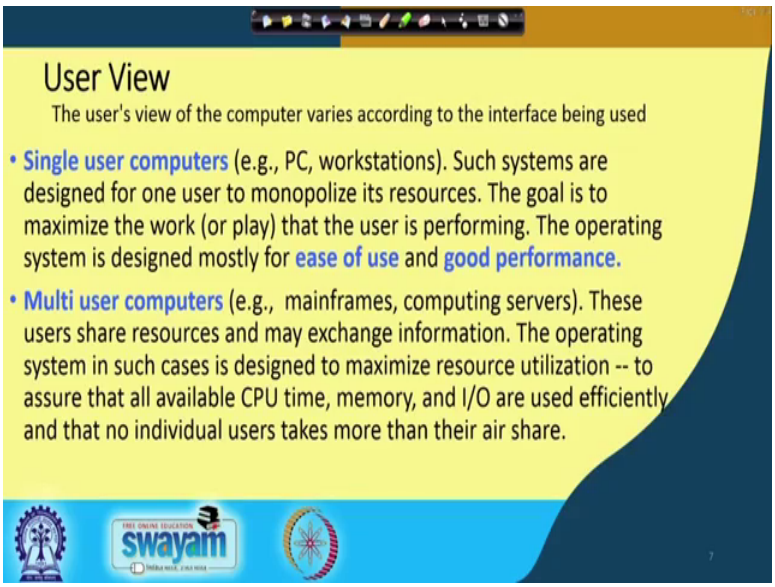
Operating System Fundamentals
Prof. Santanu Chattopadhyay
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 02
Introduction (Contd.)

(Refer Side Time: 00:28)



So Operating System consists of two views; one is the user view or and another is the system view. So in the user view also you can think about different types of computers. A major classification of a computer system in terms of this user view interface is that some computers they are for single user usage.

For example if you have got a tablet, if you have got a desktop PC or a workstation, where, a single user is working at a time. So that way it is that these systems are designed for one user to monopolize its resources. I do not have to bother about if there are multiple users and since the user is alone in the system. I do not need to be concerned about protection of data across the processes created by this particular user. Because we have the, it is up to the users responsibility to ensure that it is everything is fine.

So they do not try to one program does not try to steal the data of others or one program does not try to modify the data which is held by some other program. So here the goal is to maximize the work that the user is performing. I have purposefully put it as play because many computer systems particularly for personal use.

So they are used for playing games or for entertainment and all .So that performance has to be maximized. So if you are say listening to your music, then it should not it should not be the case that processing of that file takes a time and as a result the music is interrupted in between .So that should not happen. So my computer system should be such that my operating system should be such that these facilities are provided very easily. The operating system is designed mostly for ease of use and good performance like on a desktop, you will hardly find say touch screen type of interface.

Whereas if you are having a tablet or mobile phones and things that devices like that, they are also we have got an operating system. But there it is it is likely that the user will be using that type of interfaces. So, we have to think about the ease of use and the performance issues. So normally people will not like to attach a keyboard with a cell phone. So that is why on the cell phone screen itself that keyboard has ever has been made. But on a desktop PC people will not like to have such a keyboard available on the screen itself.

So most of the time because we have got a separate keyboard full fledged keyboard available which is going to be used by the user. So there that makes the difference between a single user system and a multi user system from user's perspective. The second part second type of system that we have is the multi user computers. So we have got mainframes and computing servers.

So, mainframes so they are big systems. So, they are used for different purposes maybe for high level computations, large amount of scientific computations and all or it may be for providing some services some database server. So like that we can have it like that or it can be some computing server all together. So it can be for example, may be I have I am running a laboratory class, where there are hundreds of students who are writing their programs and that way the system has to provide computational service to all these users.

So these multi user systems so there we will have multiple users and that they will be they need to be their jobs need to be completed in an efficient fashion. So to go through it so these users share resources and may exchange information. So in a multi user system it may be the case that the one piece of program is developed by multiple people and these multiple people, they all these programs need to run simultaneously and since they are part of a big system so they have got some data common between them.

So, something computed some data item value computed by say program 1 is used by the program 2. So that way they need to exchange the information and they have to share resources also. Maybe some process 1 is writing on to a file and this file has to be read by process 2 for doing some other operation. So this way this file becomes a shared resource. So when process 1 is writing on to the file, process 2 should not read it or simultaneously, when process 2 is reading it, process 1 should not try to modify it. So, otherwise the file content will become inconsistent and either of the two processes may start misbehaving.

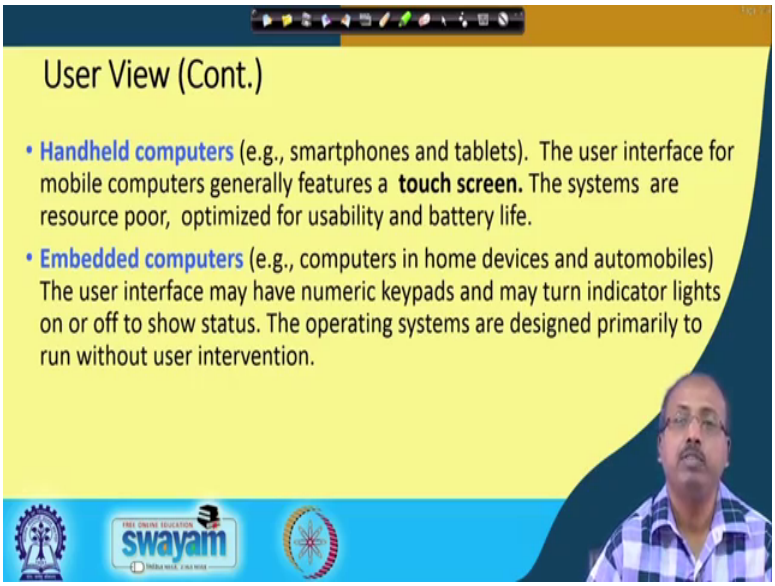
So that is the issue so the operating system must provide this type of synchronization, where they will be ensuring that this components this exchange of information can take place easily. So the operating system in such cases are is designed to maximize resource utilization. So here ease of usage may not be a big issue but this resource utilization. So percentage of time these resources are used, like in a computer system if I have got say two printers connected then they are say two disk drives connected and all.

So what I would like to have is 100 percent utilization of the printers 100 percent utilization of the disk drives. So, whether it is possible or not that is a different issue, but I would like to as a as a system designer, I would like to have 100 percent utilization of all these system resources. So this CPU time, then the memory and I O, they should be used efficiently and we have to ensure that individual user's takes they cannot take more than their air share. So, this is very important because normally when we are writing programs and executing the programs in an institute or in our college.

So, we do not we do not pay for that, so but in an in a in an organization where they are you providing this computing service, so there everything is payable. So the user should be user should be charged for the amount of CPU time that the user has used amount of disk space that the user has used, amount of main memory that the program has used, so like that. So there it is the case that if I have if the user has paid for certain amount of resources. So the user should not be allowed to spend more than that.

So that is that is what is said here that they should not take more than their air share. So, this multi user computers, they becomes more difficult. So the operating system design there becomes more difficult. So we will try to see how this can be taken care of in the operating system design.

(Refer Side Time: 07:44)



Handheld computers: so this is another class of user view of computers. As I said, the smart phones and tablets and as I said that they have generally have a touch screen which is not available in other computer systems that we have looked into like desktop or servers like that.

These systems are resource poor, so they do not have very large amount of memory. So in if you are using a cell phone, then we have got some memory in the range of GB only , gigabyte 32 GB or 128 GB, so like that. But we cannot have very large amount of storage ok. So, that makes it difficult for storing large amount of data. So of course so what you should do is that periodically you should upload the data to some place where we have got huge amount of ah huge space available. But the operations are optimized for usability and battery life this is very important.

So if for some doing some computation your battery that is the energy that is taken is very high then the battery will drain out very soon. So the web browser that we have in a desktop system and a web browser that you have in a mobile phone so they should not be of same capability because the amount of computation that is needed for doing this thing on a cell phone, they should be substantially low compared to that in a general computer system. Or if you are trying to do some computational jobs so they are normally not put into these handheld computers because of the reason that the amount of power needed will be more so as a result the battery will drain out.

So the handheld devices they are optimized towards these web browsing and things like that. So we can we should be to making phone calls and all. So very simple type of operations and there the battery life becomes a very important issue. Then we have got embedded computers. So embedded computers, so now everywhere that you see any device or big mechanical or electrical device that you see. So many of them have got some computing platform within computing resource or computing machine within that.

For example, when you are driving a car there are a large number of processors which are inside the car. So which is actually controlling the operation like this braking system fluid, then the fuel injection system, this power window, power steering, then this environment control AC.

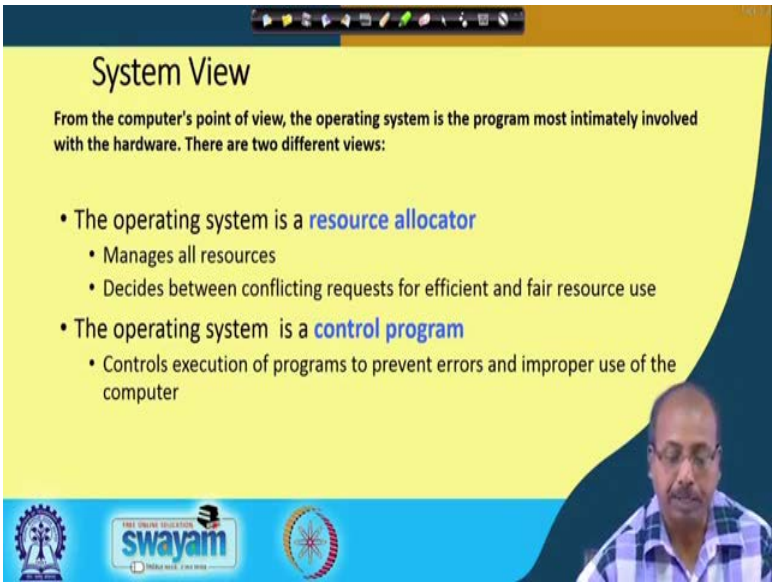
So, like that all these things are taken care of by the processors that we have within the car. Now, apparently as a user of the car, so we do not see all those processors ok. So they are just hidden, so we just see a few steering wheels and buttons here and there to press and that way the car moves. But we do not understand that.

Similarly, if you look into your washing machine, then it has got a number of sensors and actuators and processors which are actually doing the computation to control different phases of the operation like how much water should be taken, what is the temperature level accordingly, how many how many turn the clothing cloth bead made and all that. So how much time the spin dry should work? So this way we can have there also we have got computer inside, but it is dedicated to some particular work.

So, this, so they do not have interfaces that we have in general computer systems, but they have got interfaces like say numeric keypad and some indicator lights here and there that can show the status. So they operating systems that we have there. So, they are designed primarily to run without user intervention. So the users view of the system changes though in all the cases starting from your desktop computer to mainframes to this handheld cell phones to these embedded computers like refrigerator, fridge and refrigerator, washing machine, AC and all that.

So, everywhere you see or car, so everywhere you see some processor is there and some operating system is there, but their goals are different and the view of the system is different. So nobody will like to do some computation on the embedded computer that we have in a car so that way it changes.

(Refer Side Time: 12:28)



Now, from the system viewpoint the operating system is the program most intimately involved with the hardware because they are directly talking to the underlying hardware. So from computer point of view so this is the point. So, as a system designer, so we have to see like how can I talk to the operating system, I talk to the underlying hardware and how can I provide the services to the user and those services are ultimately done by the underlying hardware.

So I have to tell the hardware to do certain things in certain fashion and I should be able to give those comments in a proper way. So in this way as from a systems perspective an operating system is a resource allocator. So it manages all resources it has to manage all the resources and it has to decide between conflicting requests for efficient and fair resources, so this is very very important. Because I have got multiple processes, particularly in multi user systems. So, we have got multiple user processes and they are requesting for system resources and I have to make sure that they are not given simultaneously.

If the resource depending on the type of resource, for some of the resources like printer I cannot give it simultaneously to more than one process. Whereas, if you look into some other type of resource for example, the screen. So normally if the screen is divided into a few windows and multiple user programs, so they can be made to run on different windows or even if it is a same window.

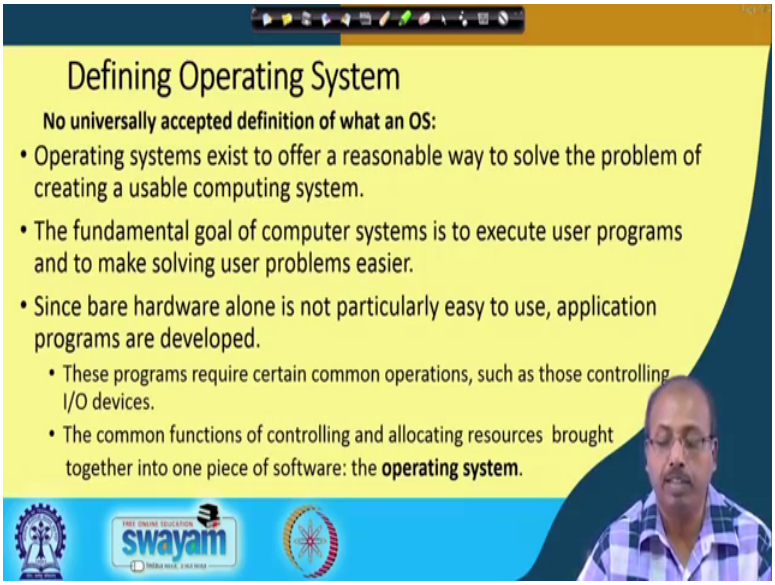
So, it really does not matter if different programs they write on to the same window may be because of this writing on to the same window only, so the effect of the overall application turns out comes out. So that way so in some cases, we are required to allow simultaneous access, some cases we are we should prevent the simultaneous access. In some cases it is possible that the resource was given to the process, but we can take it back.

For example, for every process to execute it needs the access to the CPU, so CPU will be executing the process. Now, suppose I have given a process for execution in to the, CPU in between a very important task comes and I have to do that immediately. Then for a CPU, we will see that it is possible that we suspend the currently running task and give this CPU to the new newly arrived urgent task and when that task finishes, we restore the original task and that continues in the as if nothing has happened. Only some time has passed, but as far as the system is concerned nothing has happened.

So, this way the processor can be taken back from the process or the CPU can be taken back from the process but think in terms of the printer. So printer is given to the process, but even if some other processes arrived that needs the printer immediately, but that cannot be given. Because the printed result will be mix of outputs from two different programs and that may not be useful. So this way the resource allocation is a very important issue. So it has to manage the resources and it has to decide between this resource requests.

At the same time we have to take care of the case that some process is the not facing the problem of starvation , that is it is not it has requested the resource, but it is not getting it at all, so that should not happen ok. So, that has to be taken care off. Then this operating system is a control program, so it controls execution of programs to prevent errors and improper use of the computer. So, this is another view of the operating system so as a control program.

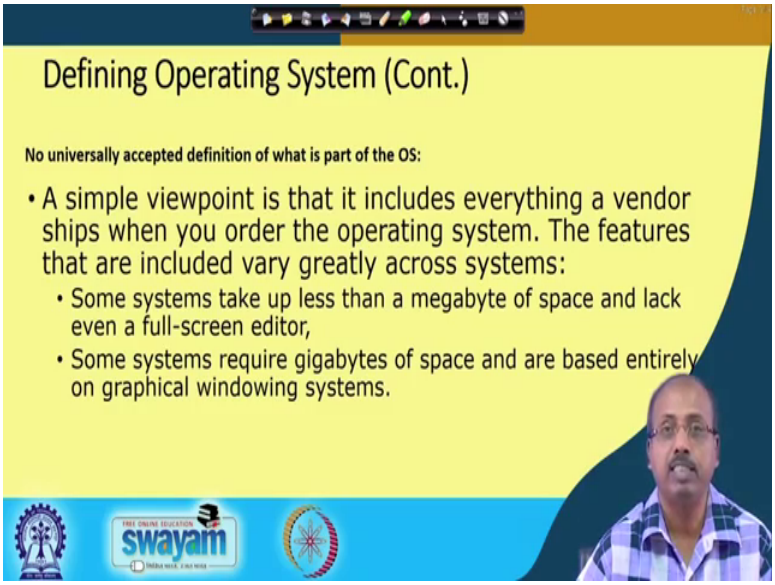
(Refer Side Time: 16:32)



So again coming back to the definition of operating system. so as I already said there is no universally accepted definition of what an operating system is. However, we can think of several definitions like this. Operating systems exist to offer a reasonable way to solve the problem of creating a usable computing system, so this is one type of. So, this is basically in some sense we are talking about the interface between hardware and the user, so again the same thing. Fundamental goal of computer systems is to execute user programs and to make solving user problems easier. So this is the second point. So we will try to make the problem-solving experience of the user better. So bare hardware alone is not particularly easy to use. So application programs or application programs are developed. So if you as I said that if you have got only the hardware and the operating system, then it is not easy to develop applications. So, we have to have some application programs developed.

So these programs require certain common operations such as controlling the I O devices and the common functions of controlling and allocating resources brought together into one piece of software the operating system. So ultimately so we can say that the controlling and allocating resources. So, they are we and do it separately; we can make different different programs for controlling different if I hardware modules. But they are taken together into the paradigm into a single program which is the operating system.

(Refer Side Time: 18:16)



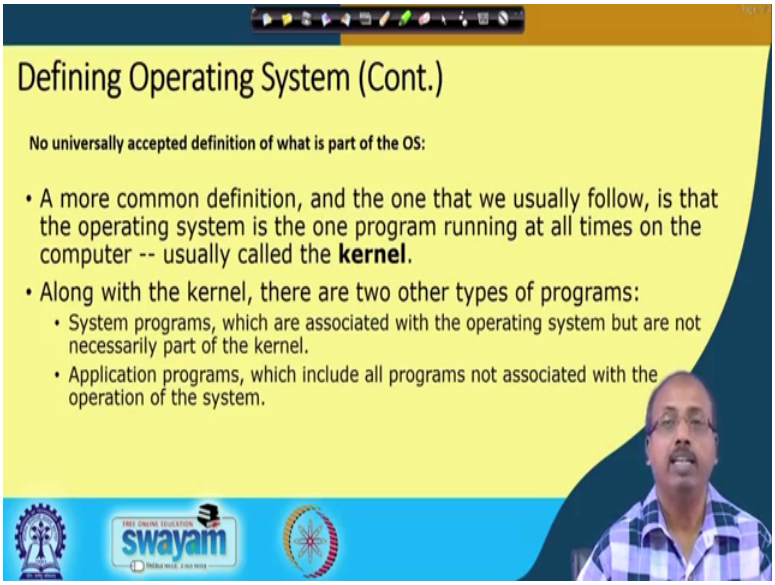
So, that defines what is an operating system. Again another view another definition may be like this a simple view point is that it includes everything a vendor ships when you order the operating. So this is a very very I should say general way of telling what is an operating system. So everything that you get from the vendor when you ask for an operating system is the operating system.

And so the features that are included vary greatly across systems, some systems take up less than a megabyte of space and lack even a full screen editor. So may be that when you are buying a microcontroller or microprocessor board. So, it also has got an operating system, they call it a monitor routine or something like that. So, they do not have these interfaces for this screen and all. So they can, so, there also we have got an operating system. There may be that there is a LED or LCD display and there is a keypad that is given.

So you can as soon as you switch on that board. So it comes up with a message like welcome or something like that and then you just go on entering your program in some hexadecimal code through the interface. So that is it that is one type of interface that you can have. So these devices so, they have that is also a computer system because that is doing the computation and the total space needed may be less than a megabyte and even a full screen editor is also not there. Whereas, some systems which are more very very general. So that will require gigabytes of space and based entirely on graphical windowing system.

So, you get a graphical user interface and all. So very advanced operating systems like say Windows or Linux or say Sun Solaris and all. So they come up with very big and very well architect this software that can be that and that is used for controlling the behaviour of the overall system. So this way we can get different definitions of operating systems.

(Refer Side Time: 20:25)



A more common definition and the one that we usually follow is that the operating system is the one program running at all times on the computer usually called the kernel. So we will say that operating system is something that is always running in the system. So, in a whenever you are asking for some service from the system so we have to send a request to the operating system.

For example, I want to execute a program. So I have to tell the operating system see I want to execute this particular program. So what the operating system will do in turn, it will create an environment in terms of this memory space in terms of IO devices and in terms of the CPU time and then once it is created, it will transfer the control to the to my program. And when my program finishes, it becomes a responsibility that the control is given back to the operating system.

So, while writing program, so we do not take into consideration all these things, but what this compiler designers do is that they depending on the system onto which you are putting your program, for which you are developing your program. So it uses calls

available in the operating system so that you can get control from the operating system or you transfer or you can transfer control to the operating system.

So while starting the program, the operating system we request the operating system to get control from it and while ending the program, we should have appropriate code in my program which will be giving the control back to the operating system.

So something is always running at the background. So, that we call the kernel. So along with the kernel there are two other types of programs that will be running one is called system programs. These are associated with operating system but are not necessarily part of the kernel and there are application programs which include all programs not associated with the operation of the system.

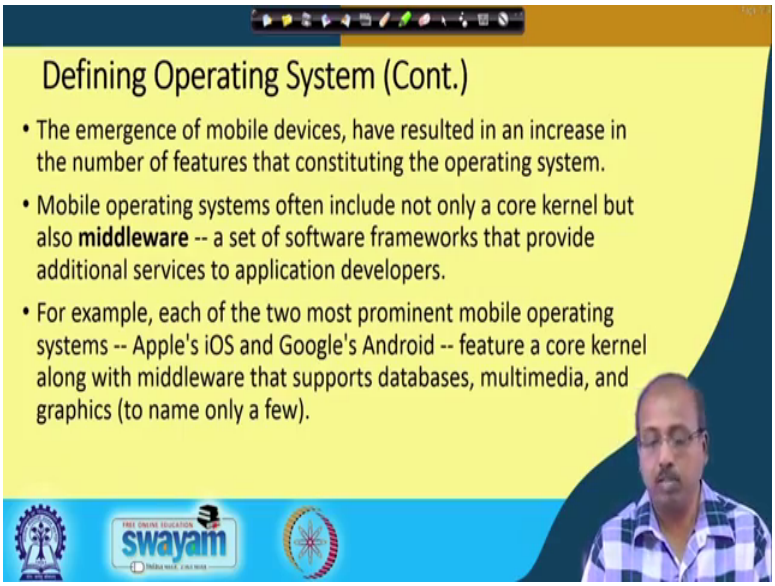
So, apart from the basic operating system that talks to the hardware get the jobs done by the underlying hardware, so there are some other programs which are used which actually helps the user to develop application programs. So these are like the database that these are like the compilers; like the linkers and so, they are actually called system programs or systems software.

And you can in some sense you can say that operating system is also part of this system software, but what we are talking about is these programs. So, we are excluding operating system because we are that is a big piece and these other software. So, they actually take help of the operating system to get their job done. As I already said the compiler to get control of the CPU, so it has to use some call from the operating system.

Similarly, to give the control back to the operating system, it has to use some services of the operating system only. So that way or the database manipulation routine. So it has to do some file manipulation. So, that the file manipulation whenever it needs to do. So, it has to talk to the operating system to get the file access etcetera.

So that way we can think about the system program. So they are actually using services from the operating system. And application programs are those that are designed on top of these system programs. So, which will be doing operations for the system. So that is for user things about doing a particular job. So that is done by the system program and that is that will take help of the underlying operating system.

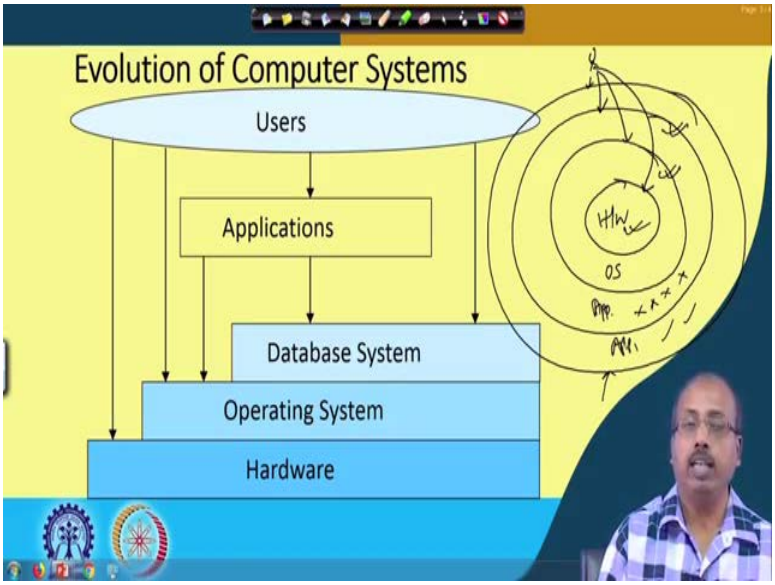
(Refer Side Time: 24:19)



So, the emergence of mobile devices so, they have resulted in an increase in the number of features that constituting that constitute an operating system. So, the many many features have been added like say sending a short messages or SMS. So normal operating systems we did not have that, but any operating system on a cell phone must provide me a facility by which this SMS can be sent. The mobile operating systems often include not only a core kernel, but also a middleware: a set of software frameworks that provide additional services to application developers.

So as I said that some extra thing that are provided by which the application developers they can develop their piece of software. For example, each of the two most prominent mobile operating systems apples IO's and Google's android, they feature a core kernel along with a middleware that support database multimedia graphics. Many more are there these are these are just a few. So, over on top of the basic how the operating system or the kernel operating system, there are some middleware that can talk to the operating system services and make the development of application on those systems easy.

(Refer Side Time: 25:36)



So how did this of computer systems evolved. So, started with the basic hardware and the users. So initial computer systems as I said that the basic hardware, so it will understand the meaning of electrical signals. So these electrical signals the users can give to the hardware and the hardware will respond to that. So that way it can continue. So, this users and hardware, so there they were at the first two components in a computer system.

Then came the operating system and the users are talking to the operating system. So to get the services from the hardware users are talking to the operating system and accordingly the users are getting that job done by the hardware through the operating system. Then comes the application developments. So users start developing applications and those applications are taking help of the operating system and the to get the job done. And finally came the intermediaries like the database system or many other the compilers and all. So those things were being developed and they made the life of the user even more simple.

Because they can they have got now another layer on top of operating system by which they can talk to the underlying operating system and underlying hardware. So that makes it ah very easy. And so, how many such levels are there at this point so, between these applications and this operating system; how many layers are there. So that is now a big question and many a time many a days so we develop one piece of software and using that software, we develop another software. So, as we are increasing this level, so or this level of indirection so that way that as an end user of the system the life is becoming

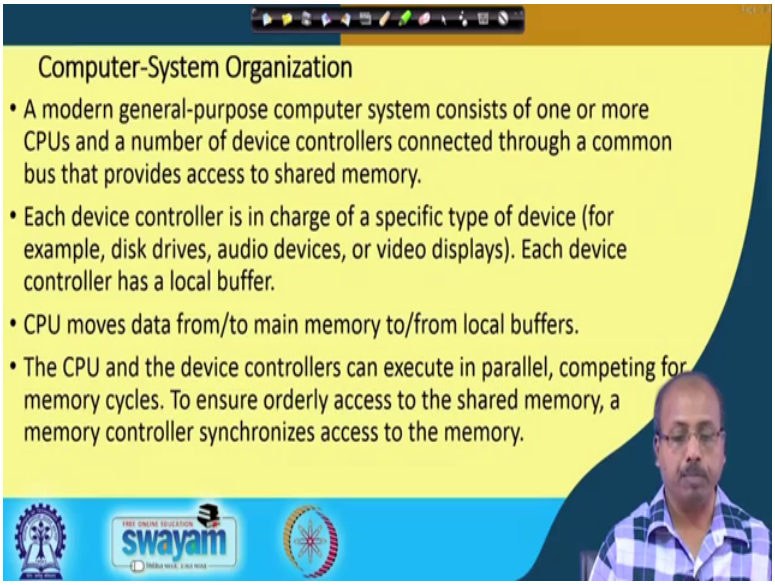
simple. But so and that way those systems are evolving. So the life of the system life of the user is becoming easy.

So user can very easily develop some application. For example, say this Java. So Java, if you write a program in Java then that can be ported to any system and so, we do not need the compilers and also we have. So we have got interpreters available and they will be to doing the job, but the problem is that. So, there is some machine that comes in between or so, some software that comes in between that can interpret this Java code and get it executed by the underlying operating system or the underlying hardware.

So that way it is not a machine code unlike that so like a C code, compiled C code. So that is directly a machine code and that can be given directly to the hardware. So, this availability of the intermediary software so they are helping me like if you are porting a program written in Java to another system. So it is very simple. So, you just take the byte code and take it there; whereas, in case of compiler.

So, a program a compiled version of the program for system one cannot be easily ported to system two if the underlying processor is different. So what you need to do is that you need to compile the program again at the second at the second platform and develop the machine code there for execution. So this way the evolution is taking place and it is a continuous process. So layers after layers are getting added and that way the system is operating.

(Refer Side Time: 29:07)



So in fact you can think of it as if we have got a situation where we have got this basic hardware we have got this basic hardware and on top of that, we have got we have started developing the layers. So OS is a layer, then some another layer, then some another layer, so, as a user of the system. So user can talk to this layer directly or can talk to this layer or the user can talk to the upper this layer or the user can directly talk to the hardware. So everything is possible. So if you are trying to get maximum from the system, then you should talk directly to the hardware.

If you are so but talking to the hardware is difficult because you have to send electrical signals. So,if you want to be slightly higher level, then you can talk to the operating system. Then even in that case you need to know the details and services provided by the operating system. So that makes it difficult. So, you can go slightly higher level; at this level that we have got some application software's and this application software, they are specific for a particular job and maybe that makes the software development job easy.

Then you have got another layer of application software. So that uses the features available at this layer to develop some facilities at this point and again. So, as a user, so I can use this application layer and develop my software. So this way we can think about adding layers after layers to the basic operator hardware over and above the basic hardware to make the life of the user easier and easier.