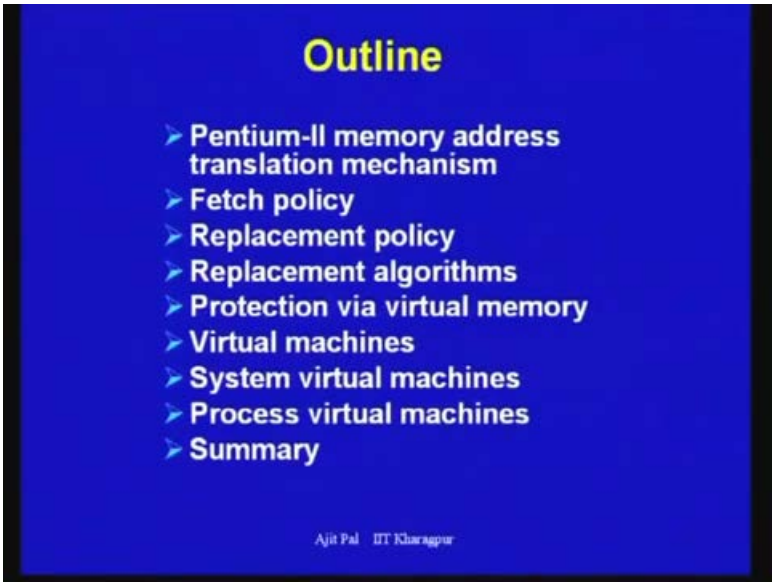
High Performance Computer Architecture
Prof. Ajit Pal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Virtual Machines

Hello and welcome to today's lecture on Virtual Machines, we have been discussing virtual memory which is a part of the hierarchical memory organization; and as usual see virtual machines is an extension of the idea of virtual memory. So, today we shall discuss about some of the aspects of virtual memory, we have not been able to cover in the last lectures and then, we shall switch to virtual machines.

(Refer Slide Time: 01:28)

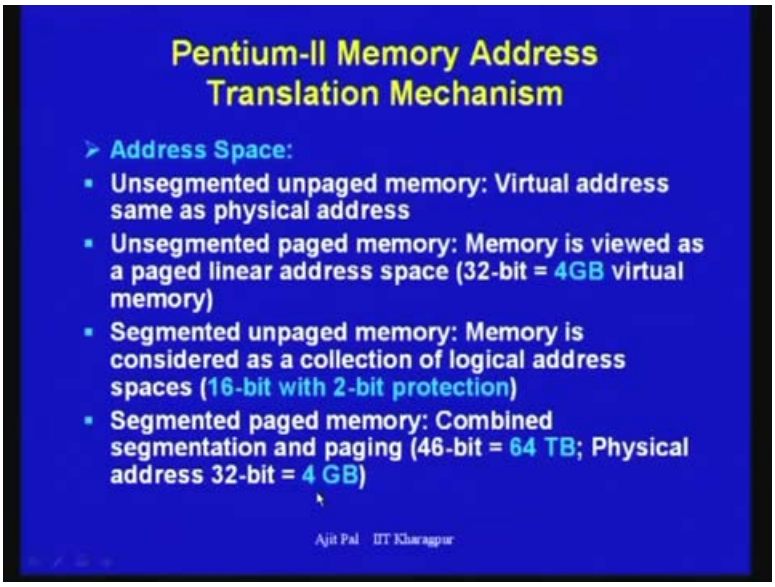


So, here is the outline of today's lecture, first I shall discuss about Pentium-II memory at this translation mechanism, and discuss about several policies like, fetch policy, replacement policy. And then, replacement algorithms which are done as part of the virtual memory management system and then, protection by how the protection is achieved with the help of virtual memory and then, we shall extend this idea of virtual memory to virtual machines.

And particularly as we shall see, there are two classes of virtual machines one is known as system virtual machines, another is process virtual machines, which we shall discuss

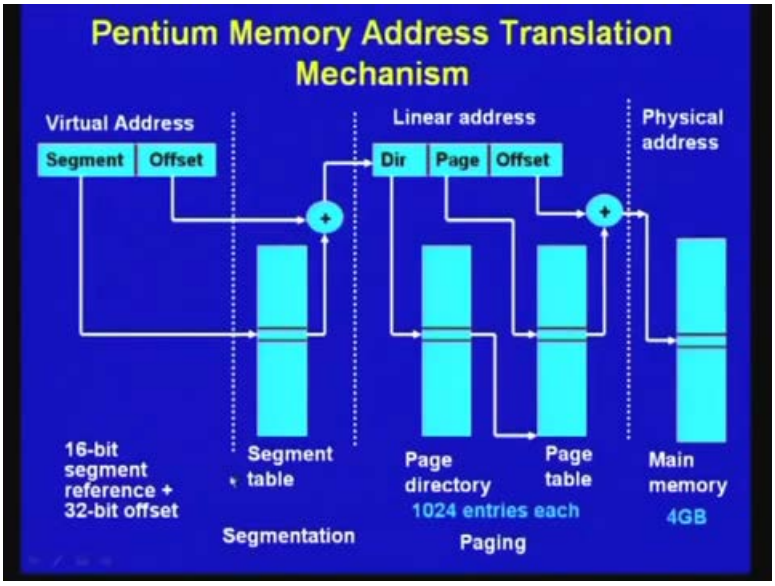
in detail. And then, we shall summarize the lecture, I mean covering virtual memory and virtual machines.

(Refer Slide Time: 02:19)



So, far as the Pentium-II is concerned as we know, there are several alternatives available it can use unsegmented unpaged memory, it can use un-segmented paged memory. It can use un-segmented unpaged memory and segmented paged memory, which combines segmentation and paging, which I have been mentioned in my last lecture. And as it uses 46-bit providing 64 terabytes of memory, that is virtual memory of course, there are 2-bits used for the purpose of protection, so it has got 48-bit virtual address that is generated by the processor. And physical address is 32- bit and with that you can have 4 gigabyte of physical memory.

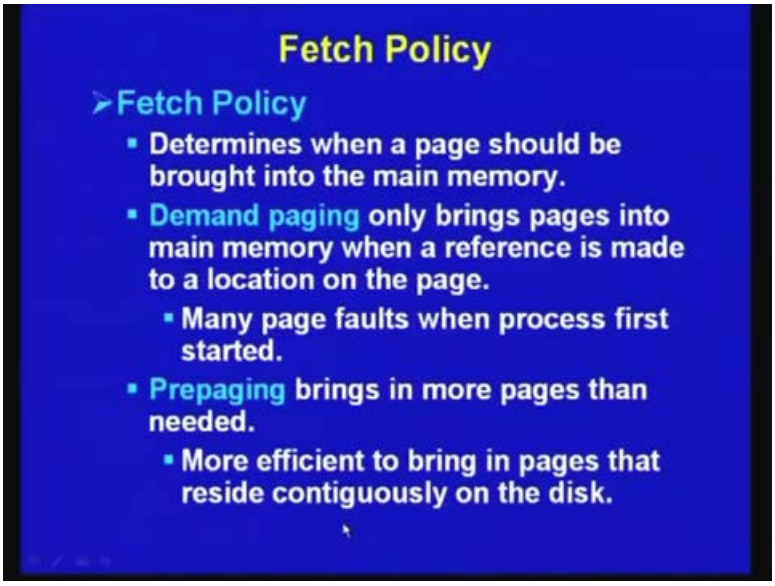
(Refer Slide Time: 03:14)



And this is how the translation take place, it generates the that 48- bit address that is 16-bit segment address, that is the segment reference and 32- bit offset, that is actually is used for the purpose of linear address for translation. So, we have got a segment table and with the help of that segment table pointer that is the base address, and the segment offset that piece obtained. And that provides you the segment address with that offset is added to point to the linear address; and which is as we can see, it comprises three parts, the first is page directory, then page table and offset.

So, page directory provides you the base of page table, from where you get the particular page reference number or page frame number. And which is used to generate the physical address by concatenating the offset to get the 32-bit physical address and that that gives you 4 gigabyte of main physical address. So, this is how the Pentium memory at this end, at this translation takes place.

(Refer Slide Time: 04:47)



Now, we shall focus on some of the policies that is used in virtual memory system, number 1 is known as fetch policy. So, this determines when a page should be brought in to the main memory, as we know all the pages are present in the secondary storage, then from the secondary storage you have to transfer to main memory. And when to do that, that means when a page should be brought in to the main memory, so in virtual memory management system, in virtual memory system the technique that is used is known as demand paging.

So, demand paging only brings pages into the main memory, when a reference is made to a location on the page that means, when a address is generated by the processor referring a particular page, only then that page is transferred from the secondary memory to the main memory. So, that is why it is called demand paging and only recently accessed pages are available in the main memory. And that means, only brings the pages into the main memory, when a reference is made to a location on the page and many page faults when process first started.

So, this is inevitable, because in the beginning not a single page is present in a main memory, so as the processor keeps on accessing different pages, they have to be transferred from the secondary storage to the main memory, and it each will lead to a page fault. So, that will lead to many page faults, there is another policy that can be used which is known as pre paging, so pre paging brings in more pages than needed. So, in

this particular case, it is different from the demand paging, although this is advantageous in many situations, but rarely practiced.

So, this is more efficient to bring in pages that resides contiguously on disk, because of the locality of references, the processor keeps on accessing contiguous memory locations. And if they are stored in consecutive pages, it is preferable to transfer several pages one after to the main memory, so that is your pre paging that means, bringing pages that reside contiguously on the disk. So, more than one page can be transferred whenever a page fault occurs, but as I said this is rarely practiced, because that takes long time, but it can be done to improve the efficiency, so this is the fetch policy.

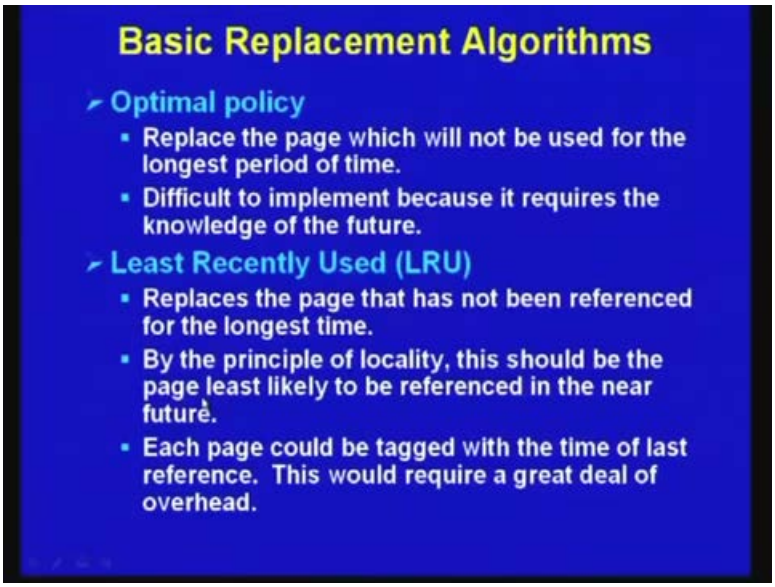
(Refer Slide Time: 07:49)

Then comes the replacement policy, so determines where a real memory a process piece is to reside, this is important in a segmentation system; and paging or combined paging with segmentation hardware performs at this translation. So, placement that means, it determines where it is present in the real memory, that placement is important and particularly as we have seen, because of the limited size of main memory, a restricted number of pages should be present in the main memory. So, continuously there will be swapping of pages between main memory and secondary memory.

So, you have to use some replacement policy, which decides which page is replaced that means, whenever a new page is brought in you have to replace a new page. So, the page removed should be the page least likely to be difference in the near future, so that is the

ideal situation. So, what is being done in practice most of the policies predict the future behavior based on the past behavior, so you cannot really predict the future, but based on the past behavior some prediction about the future can be made, and based on that the replacement is being done.

(Refer Slide Time: 09:30)



And there are several approaches, the first one is known as optimal policy, so they replace the page which will not be used for the longest period of time, so who can tell that, replace the page which will not be used for the longest period of time. It is difficult to predict, because in a dynamic situation when a program is getting executed, it is not possible to predict, possibly the prediction can be done by an astrologer, but we cannot keep an astrologer inside the processor.

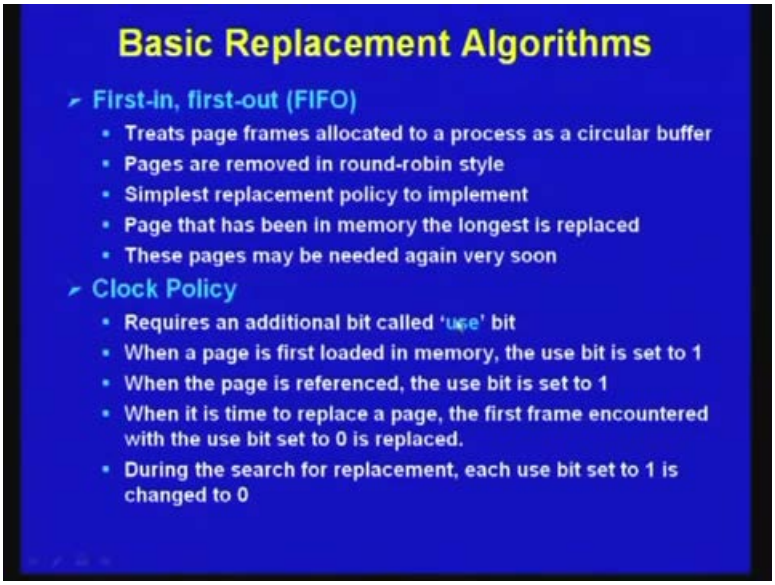
So, it difficult to implement or not possible to implement, so difficult to implement because, it requires the knowledge of the future, so this policy although it is ideal, this cannot be adopted in practice. So, the most commonly used technique that is used is known as least recently used, LRU technique, so it replaces the page that has not been referenced for longest time. So, what is being done least recently used that means, you have a number of pages and there is some mechanism by which you can keep track of which page has not been used for a longest period of time and that page is replaced.

Obviously, this will be helpful, because by the principle of locality, temporal locality this should be the page least likely to be referenced in the near future as temporal locality

tells that, if a page is referenced now, it will be referenced in near future likely to be reused in near future. So, if a page has not been used for a long time, it is unlikely to be used in near future, so that is the basic idea behind this least recently used technique. What can be done each page could be tagged with the time of last reference, so you have to keep a kind of time stamp, for with each page and obviously, this would require great deal of overhead.

So, it is difficult to really keep track of that time, I mean for each page you have to store the information of time, and it involves lot of overhead. So, it can be implemented, but it lot of overhead is involved in this particular technique.

(Refer Slide Time: 12:07)



Another technique that can be used is first in first out or FIFO, so treat page frames allocated to a process as a circular buffer that means, as the pages are that means, you are bringing in the pages, the first page that was brought in is removed first, so first in first out. So, pages are removed in round-robin style, it is very simple to implement possibly simplest replacement policy to implement, and page that has been in memory the longest period is replaced as I mentioned.

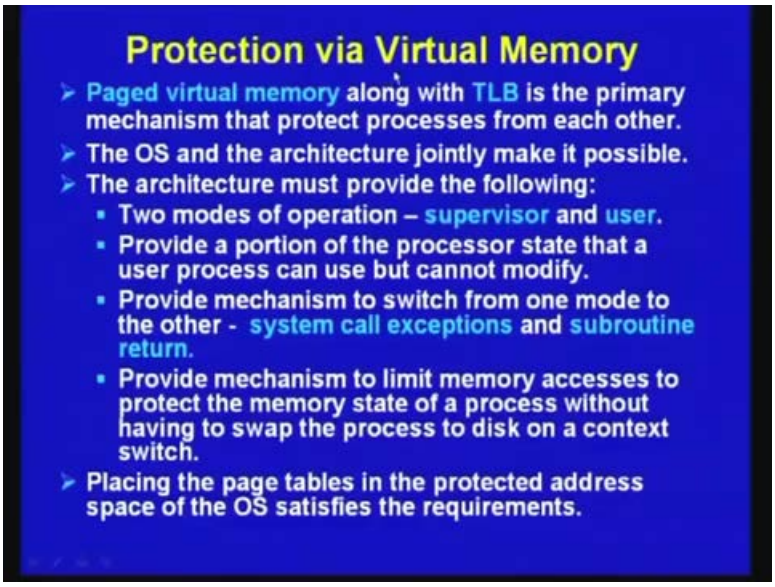
So, these pages may be needed again very soon, so the although it is very easy to implement this first in first out, FIFO type of replacement algorithm, but it does not really give you very good result. The reason is for that is, the page which is being replaced now may be required in near future, so it does not really give you very good

result. Another technique is known as clock policy, so in this case this requires an additional bit called use bit, so when a page is first loaded in memory the use bit is set to 1.

And when the page is referenced the use bit is set, I mean the page is brought and set to 1 and again when a page is referenced to be set to 1 why it is done, because when it is time to replace a page with first frame encountered with the use bit is set to 0 is replaced. So, that means whenever you are replacing a page the first frame encountered with the use bit set to 0 is replaced. But, how you are setting it to 0, during the search for replacement each use bit is set to 1 is changed to 0 that means, whenever a particular page is replaced, all the other use I mean use bits of all the other pages are set to 0. So, subsequently again whenever you reference a page, then you set to 1, so that is how it works.

And this is known as clock policy, because it is dependent on the reference by the user. So, as you can see the overhead in this particular case is very small, because you require only a single bit, used bit. And it has been found that it gives you reasonably good result, and that is the reason why it is one of the most commonly used technique.

(Refer Slide Time: 14:55)



Now, we shall discuss about the idea of protection by virtual memory, we have seen that, that one of the important use of virtual memory is protection. Because, security and privacy has become a very important aspect in present context, because large number of users are using a computer system, it is shared by many users and so on. So, the virtual

memory provides protection in nature and natural way, and we have seen that it is done along with the TLB, TLB actually enhances the performance.

The operating system and architecture jointly makes this possible, and we know that the architecture must provide the following to facilitate this protection, these are the three requirements. Number 1 is two modes of operation, supervisor mode and a user mode and provides a portion of the processor state that user process can use, but cannot modify. So, in there are two modes user mode and supervisor mode, so in the user mode certain protection bits, certain flag bits cannot be changed for example, there is a flag bit known as user supervisor flag bit, that cannot be modified. Similarly, there are many excess bits ready only, read write only, access only like that, those bits cannot also be modified by the user.

So, it should be done by the supervisor, so that means in the supervisor mode the processor state can be modified, but not in the user mode. And the architecture must also provide a mechanism by which to switch from one mode to another mode. And as we know whenever the processor is in the user mode, it can switch to supervisor mode by a system call, so system call is a kind of software interrupt, whenever system call is that is an instruction is executed, control is transferred to the supervisor.

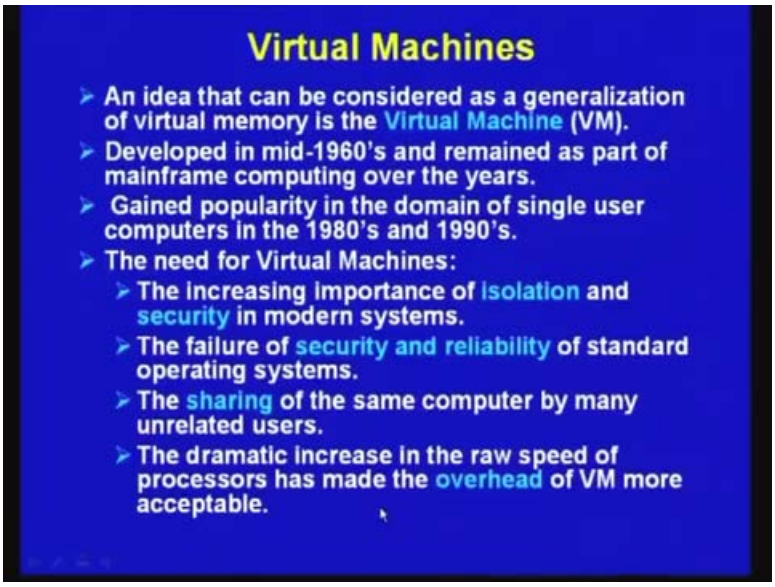
Then the processor state is saved, program counter is saved and it switches to the supervisor mode, and in the supervisor mode necessary operations are performed. And then, the processor returns to the user mode by simple that software return instructions at the end of the supervisor call subroutine, there is a return instruction and that return gives a transfers it back to the user mode. So, this is how the switch from user mode to supervisor mode, and supervisor mode to user mode can be achieved.

And that is also provided by the architecture, and it also provide mechanism to limit memory accesses to protect the memory state of a process without having to swap the process, to disk on a context switch. So, whenever context switch is occurring, then it provides a mechanism to limit memory access, to protect the memory state of a process without having to swap the process to the disk. And these three placing, I mean page tables in the protected address, space of the OS satisfies the requirements.

As we have already seen that page tables, where the information is stored that is kept as part of the I mean supervisors address space, so that means that particular thing cannot

be accessed by the user. So, we know that page table is a very critical information, if the user is allowed to modify the page table, then the protection required by the virtual machines cannot be achieved. So, only the supervisor is allowed to modify the page table and that is the reason why, it is kept in the address space of the supervisor. So, this is how the protection is provided by the virtual memory.

(Refer Slide Time: 19:30)



However, this protection is possibly not enough and that is the reason why the concept of virtual machines was introduced, so this virtual machine concept can be considered as a generalization of the idea of virtual memory. So, we have seen in virtual memory, we are essentially giving the user, the notion of the unlimited memory. But, here we shall see that virtual machines is essentially giving the, that providing the virtualization of the processor itself or the entire systems.

So, I shall discuss about in detail, what are the things provided by the virtual machines, this was developed in mid 1960's and remained as part of the mainframe computing over the years. So, in the beginning, in the 60's this was part of the mainframe systems, so implementation of virtual machines were restricted to system, IBM 370 and other machines. But, only in 80's and 90's these gained popularity in domain of single user computers like PC's in 1980's and 1990's.

Question naturally arises, why do you need the concept of virtual machines to be used in a PC's, workstations and servers, these are the various needs in the increasing importance

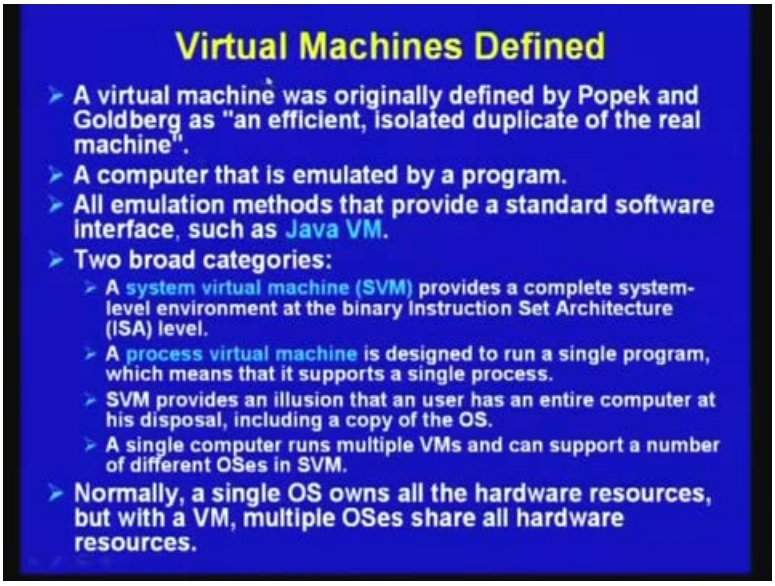
of isolation security in modern systems. So, modern systems are becoming increasingly complex, the application is becoming complex, size of the programs is becoming complex, the operating system size is also becoming very large. So, the protection that is provided by virtual memory is not good enough, so you require additional level of security and protection, and that is provided by virtual machines.

And it has been observed that the failure of security and reliability of standard of operating systems, as I mentioned the operating systems have tens of millions of lines of codes they are pretty large. And there are I mean bugs have been reported, in the standard operating systems, so by exploiting the bugs present in the operating system the security, I mean protection is violated and it becomes more vulnerable to security and protection. And that is the reason why the operating systems cannot really provide you the necessary security, which is implemented in using virtual memory.

So, and moreover the sharing of same computer many unrelated users, that is increasing particularly when in the server domain, it has been found that a server is being used by many users and may be they are unrelated. And as a consequence maintaining securities is become very difficult, and another reason for using virtual machine is a dramatic increase in the raw speed of processors. That has made overhead of virtual machine more acceptable, in the early 60's and the 60's and 70's it was difficult to provide virtual machine.

The reason for that is, that virtual machine was imposing lot of overhead and the speed of the processors was not very high and as we have seen, the speed of the processors have increased dramatically in the last one decade. And as a consequence the overhead imposed by virtual machine is insignificant, and as a consequence it has now become more feasible, more acceptable to provide this overhead of virtual machine. And these are the reasons for providing virtual machines in the present context.

(Refer Slide Time: 24:14)



Now, let us try to define virtual machine, what is virtual machine a virtual machine was originally defined by Popek and Goldberg, as an efficient isolated duplicate of the real machine. That means, we are trying to isolate the real machine duplicating it obviously, by emulation, so we are separating the real machine with the virtual machine, so with an additional level of isolation. So, it may be considered a computer that is emulated by a program that means, we have seen that virtual memory is a emulation of the, is a kind of where the user is given the, I mean illusion that the very large address space is available to it, or the entire memory is available to it.

Here also the user is given the illusion that entire computer system is available to it, but in reality it is not so, so a virtual machine gives an illusion, that the entire computer is available to a to an user, but it may be shared in practice by a large number of users. So, I mean all emulation methods that provides a standards software interface, such as java virtual machine that is an example of virtual machine, java virtual machine is quite popular. And that is an example, then the virtual machines can be broadly categorized into two types as I mentioned.

Number 1 is known as a system virtual machine, so a system virtual machine provides a complete systems environment at the binary instruction set architectural level. So, here a system virtual machine is providing the instructions at architectural level including the operating system. On the other hand, a processed virtual machine which is of the second

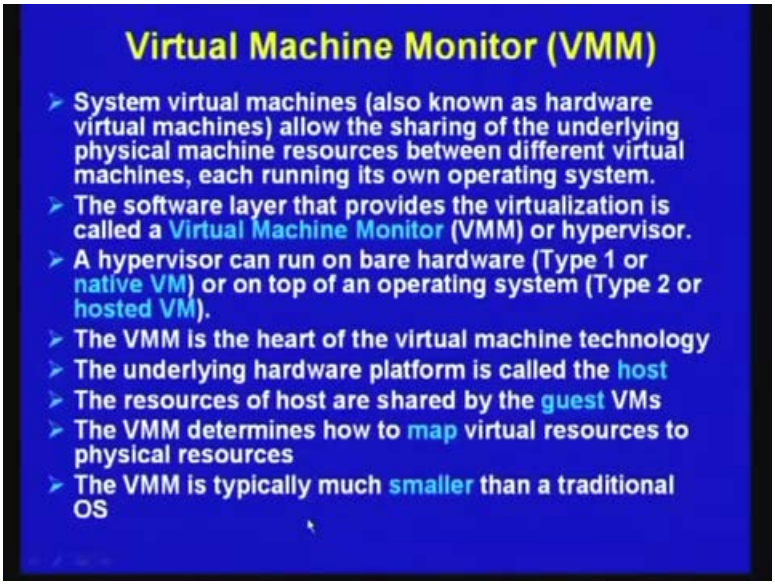
category is designed to run a single program, which means that it supports a single process. So, a process virtual machine can support a single process, on the other hand a system virtual machine can provide a large number of in the entire system is getting emulated including operating systems, you can have multiple operating systems emulated on a single processor.

So, system virtual machines provide an illusion, that an user has an entire computer at his disposal including a copy of the operating system. And a single computer runs multiple virtual machines, and can support a number of different operating systems in system virtual machines. So, it can different operating systems means, it can provide an old operating systems like dos, it can provide a existing operating systems like windows, Microsoft windows and Unix. Or you can provide the emulation of a operating system that is under development, so which is not yet commercialized.

So, that is the advantages of this virtual machine concept, so normally a single operating system owns all the hardware resources, we know that whenever a conventionally, you have a single system is having single operating system. So, all the entire resources processor IO, memory all are under the control of an operating system. On the other hand, in case of virtual machine multiple operating systems share these hardware resources. So, the processor, the memory the IO devices all are shared by multiple virtual machines that can be emulated on a single system.

So, but the users it is transparent to the users, so users feels that entire computer system is at his disposal, but that is not true in practice. So, that is the basic difference of the virtual memory with virtual machines.

(Refer Slide Time: 29:10)



And how is it implemented, so there is a concept called virtual machine monitor, so the system virtual machines allows the sharing of the underlying physical machine resources, between different virtual machines including running of it is own operating system. So, this is done, this software layer that provides these virtualization is called virtual machine monitor VMM or hypervisor, so the software that provides this interface is known as VMM or Virtual Machine Monitor.

So, a virtual machine monitor or hypervisor can run on bare hardware that is the type one or native virtual machines, or on top of an operating system that is type 2 or hosted VM. So, the difference between the two is, in the first case it is running on the native hardware, so the speed of the operation will be pretty fast. On the other hand, in the second case, where it is running on top of the operating system, so an emulated version of hardware is being used. So, in such a case the performance will be lesser compared to first case, so you have got there are two possibilities, one is known as native VM and hosted VM.

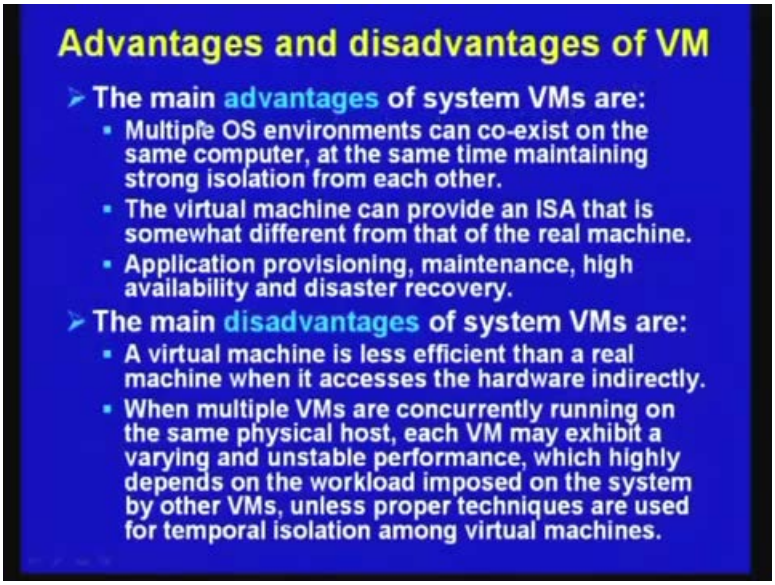
So, the virtual machine monitor is the heart of the virtual machine technology, as I have already mentioned, the underlying hardware platform is called the host. So, you will definitely require a hardware platform, processor, memory, IO devices that is the hardware platform, that hardware platform is called the host on top of which the VMM is working. And the resources of host are shared by guest virtual machines, so we will be

having a number of guest virtual machines running on top of the, which will work on top of the VMM virtual machine monitor.

So, the virtual machine monitor determines how to map virtual resources to physical resources obviously, those guest VM, virtual machines have to will require the processor for execution of their program. They will require memory to save their program, they will require IO, very different types of IO for data transfer, and also to save the data in the hard disk that is a kind of IO. So, how the real processor and the mapping to the hardware is done with the help of virtual machine monitor, so the VMM determines how to map the virtual resources to the physical resources.

Now, question naturally arises how big is the virtual machine monitor, it has been found that the virtual machine monitor is typically much smaller than a traditional operating system. We know that, I have already mentioned that, traditional operating systems are pretty large tens of thousands of lines of chords, on the other hand the virtual machine monitors are relatively smaller compared to the traditional operating systems.

(Refer Slide Time: 32:51)



Now, let us discuss the advantages and disadvantages of virtual machine, the main advantages of the system virtual machines, so here as I mentioned there are two types of virtual machines; one is systems virtual machine, another is processed virtual machine. So, here we have discussing the advantages of system virtual machine, there are several advantages number 1 is, multiple operating system environments can coexist on the same

computer, at the same time maintaining strong isolation from each other, so this a very important aspect.

You see multiple operating systems will be running on a single computer, at the same time there will be strong isolation, they will not interfere with each other, one will not that the security and protection will not be violated, so that is very important. So, there will be strong isolation from each other, the virtual machine can provide an ISA that is somewhat different from that of the real machine. So, as we know the instruction set architecture is essentially the programmers view of the processor, so you may have a native processor, which is having it is own instruction set architecture.

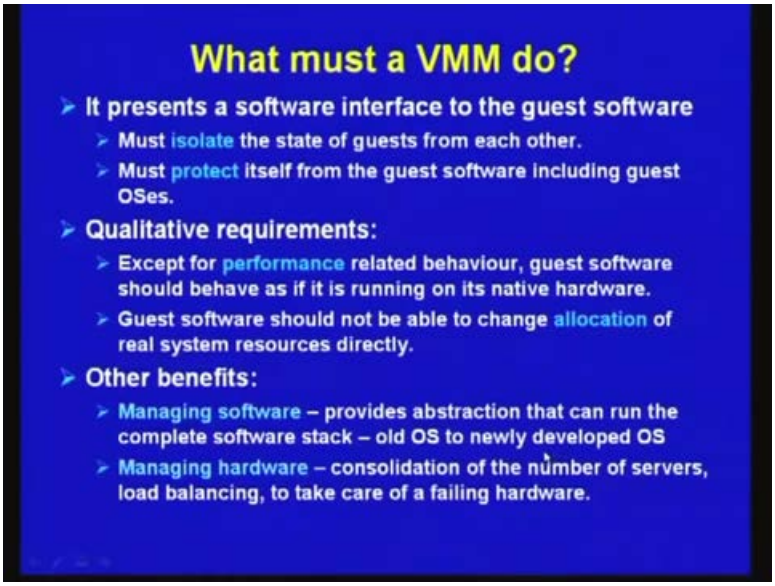
Now, a virtual machine monitor can provide you a different instructions set architecture, that is not provided by the native hardware that means, the instructions set architecture can be different from that of the native hardware, so that is provided by the emulation. So, this is very important and useful in many situations, and applications provisioning, maintenance, high availability and disaster recovery. So, you can see these are the other important aspects of computer systems, that application provisioning, maintenance, high availability and that means, to an user it should be available with the help of virtual machines, that availability is made very high and it provides you disaster recovery.

So, whenever there is some failure say for example, you are having multiple servers, so whenever a virtual machines have implemented on multiple server, if a particular server fails that will be taken care automatically by the virtual machines. That means, it will recover from the disaster or the failure, and that is why it is very useful in many situations of course, there you cannot all the advantages without any disadvantage. So, the main disadvantages of system virtual machines are as follows the virtual machine is less efficient than a real machines, when it accesses hardware indirectly.

So, it is quite obvious that, whenever you are running an application in an emulated environment, it will definitely be definitely be slower than, if you run it on the real processor. So, that is one disadvantage of this virtual machine, because of the efficiency will be lesser the time required will be more. So, when multiple virtual machines are concurrently running on the same physical host, each virtual machine may exhibit a varying and unstable performance, which highly depends on the work load imposed on the system by other virtual machines.

So, this is quite obvious, because each virtual machine is running some application and those applications can be different instances of time, and as a consequence it provides you a it provides a varying work load to the processor. So, as a consequence a particular virtual machine running in application, what kind of response time it will get that is variable, and which is not very stable as well. Of course, by using proper technique there can be some temporal isolation among virtual machines, but is very difficult to implement. So, usually it will be each virtual machine may exhibit varying and unstable performance.

(Refer Slide Time: 37:53)



So, here question arises what must a virtual machine monitor do, we have seen that, that virtual machine monitor is a software interface, what it should do, what it should provide that is being answered here. It presents a software interface to the guest software, so it must isolate the state of guest from each other. So, what do you really mean by a state of guests, state of guests is a, we have seen that each guest application will require it is own status information, which have already mentioned that page table, and various other things, those things are to be isolated from each other. And must protect itself from the guest software including guest operating systems.

So, the virtual machine monitor should also protect itself from the guest software including the guest operating systems, so the guest operating systems will be running in a single system and it should be done. That means, virtual machine monitor has to be

protected from different operating systems, guest operating systems that will be running. Then there are some qualitative requirements, except for performance related behavior guest software should behave, as if it is running on its native hardware. As I have already told there can be some degradation in performance, but apart from the degradation in performance, the quality of the result should be identical to that of running in the real hardware.

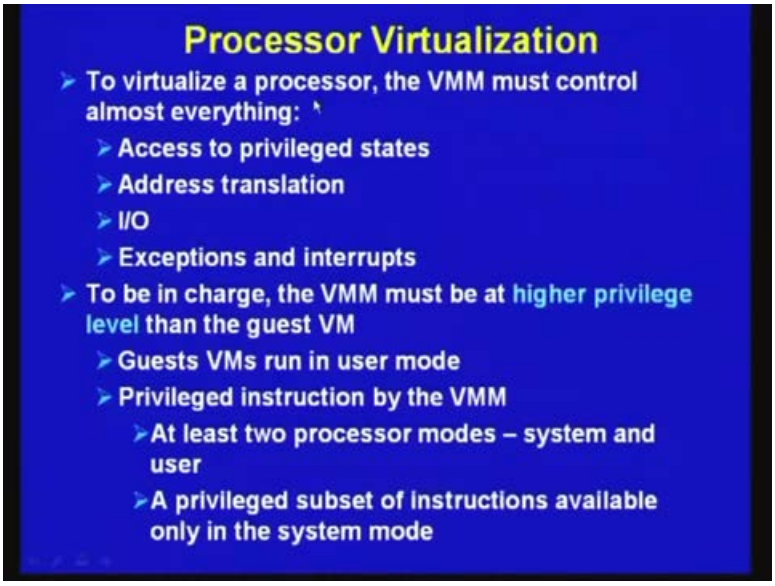
So, this is very important, because without this the virtual machines concept will not be acceptable. And guest software should not be able to change allocation of real system resources directly, so that means, we have seen that each guest virtual machine is provided some part of the hardware resources. So, that allocation should not be modified by the guest virtual machine, it should be done by the virtual machine monitor. And there are other benefit like, managing software it provides abstraction that can run the complete software stack.

As I have already told old operating system like dos and newly developed operating system along with the existing operating system, that can be provided by the virtual machine monitor. So, it can manage provides the abstraction that can run for complete software stack for different types of software's, then managing hardware here you will require the consolidation of the number of servers, load balancing and to take care of failing hardware. As I was mentioning that you may be having a large number of servers server farm, so where a large number of users or applications are running.

And the virtual machine monitor should provide a kind of load balancing that means, all the servers should be equally loaded, so that the performance level is higher. Similarly, you have to do a kind of consolidation of the hardware that means, all the different types of servers say a particular server can efficiently run a particular application. So, the virtual machine monitor will allocate different applications appropriate for a particular server.

And then, as I mentioned it will take care of the failing hardware in a real life scenario, some servers or hardware hard disk and other types of devices can fail. So, the virtual monitor will take care of it will managed the hardware and it will make it transparent to the user.

(Refer Slide Time: 42:33)



Then comes the processor virtualization which is one important aspect of the virtual machine monitor, so to virtualized a processor the virtual machine monitor must control everything. Access to the privilege states, as I have already mentioned that there will be two states, that supervisor state, user state, so that access to the privilege states will be provided only to the on a supervisor mode, but not in the user mode.

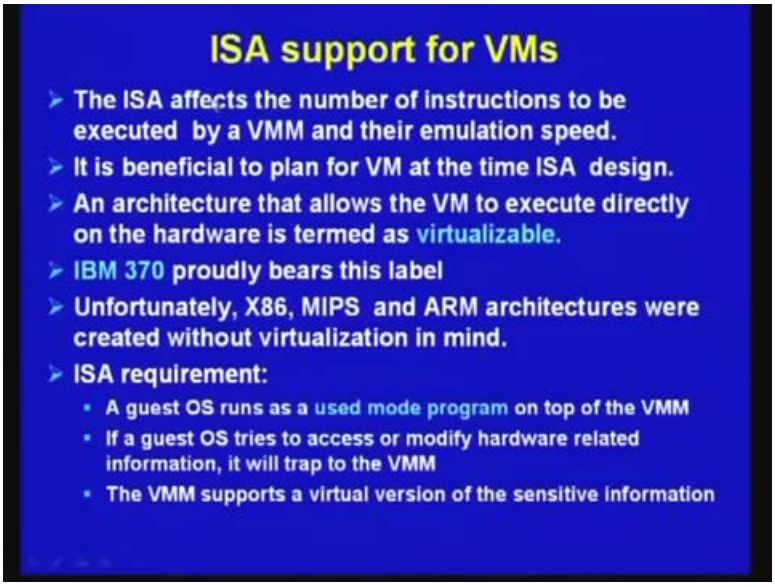
Then at this translation which does the mapping of the physical and the logical address, virtual address to physical address that translation has to be done, mapping has to be done with the help of page table. And that page table should not be modified by the user virtual machines, it should be done by virtual machine monitor, then the access of different IO devices. So, that access of different particularly that virtual machines, different virtual machines will try to act different IO devices, that should be done through the virtual machine monitor, then there will be exceptions and interrupts.

So, whenever exceptions and interruption are generated, control should be transferred to virtual machine monitor, and they will be handled in the supervisor mode under the control of the virtual machine monitor. So, to be in charge virtual machine monitor must be at higher privilege level than the guest virtual machines, so there can be several privilege levels. So, user virtual machines will have some privilege level, virtual machine monitor must have another privilege level, so privilege level of the virtual machine monitor should be higher.

By that we mean some of the instructions which can be executed by the virtual machine monitor, cannot be executed by the guest virtual machines. So, that means, the guest virtual machines will run in the user mode, on the other hand the virtual machine monitor will operate in the supervisor mode or the system mode. And as I have mentioned that will require two processor modes system mode and user mode, and privilege subset of instructions available only in the system mode.

So, those subset of instructions can be executed in the system mode, but cannot be executed in the user mode, so this is how the processor virtualization can be done.

(Refer Slide Time: 45:28)



Then comes the instructions set architectural support for virtual machines, we have seen that the virtual machine monitor is implemented by software, so it is a question arises how efficiently it can be implemented. So, that instruction set architecture determines how efficiently it can be implemented, so whenever the instruction set architecture is designed that time you have to plan for virtual machine. So, if you do not plan the virtual implementation of virtual machine, when the instruction set architecture is designed.

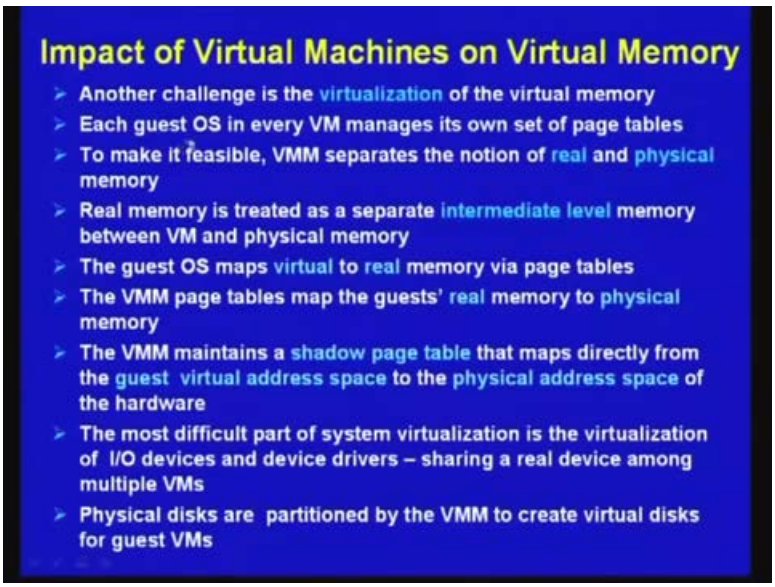
Then implementation of instruction set that virtual machine will be difficult for example, an architecture that allows the virtual machine to execute directly on the hardware is termed as virtualizable. So, some processors can be categorized as virtualizable, like IBM 370, so IBM 370 proudly bears his level, because different virtual machines can be

directly executed in their native hardware. On the other hand, unfortunately the X 86, MIPS and arm architecture were created without virtualization in mind.

So, when their instruction set architecture were designed for MIPS processors or arm processors or the 8086 series of processors, family of processors. Then virtualization was not considered, because virtual machine concept is not a very old concept it is a relatively new concept, which is used in the present context. So, then whenever it is done your implementation will be less efficient, so the instruction set architecture requirements are given here. A guest operating systems runs as a user mode program on top of the virtual machine monitor, if a guest operating system tries to access or modify hardware related information, it will trap to the virtual machine monitor.

That means, in the user mode you should not have instruction, which will allow you to modify the hardware related information those various types of flag bits that is available. So, if an user, in the user mode that is tried, then it will trap to the virtual machine monitor. And the virtual machine monitor supports, a virtual version of the sensitive information, so the sensitive information is supported with the help of the virtual machine monitor.

(Refer Slide Time: 48:38)



Now, what is the impact of virtual machines on virtual memory, so this is another challenge of virtualization of the virtual memory we have seen that, each guest operating system in every virtual machine manages, it is own set up page tables. So, for each

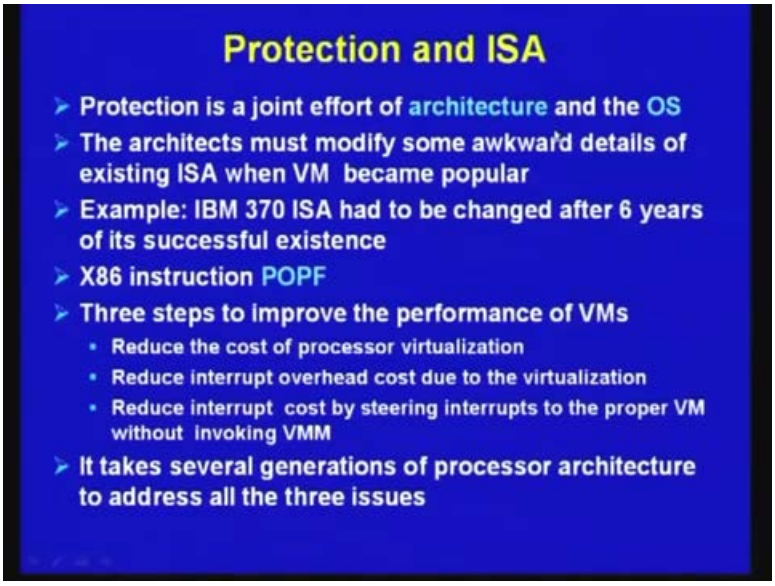
virtual machine you will have a page table, now then page table will map the virtual address to the physical address. So, each virtual address, each virtual machine will do so and that will that will lead to some conflict, so to make it feasible, the virtual machine monitor separates the notion of real and physical memory.

So, what is being done another level of abstraction is provided, so earlier you had virtual machine, virtual memory and physical memory. Now, you are introducing a real memory, so real memory is treated as a separate intermediate level memory between virtual memory and physical memory. So, the guest operating system maps virtual to real memory via page tables, so it is done in two levels and the virtual machine monitor maps the, virtual machine monitor page tables maps the guests, real memory to physical memory.

So, virtual to real is a job of guest operating system and the real to physical memory mapping is done by the virtual machine monitor. So, the virtual machine monitor maintains a shadow page table, that maps directly from the guest virtual address space to the physical address space of the hardware. So, the managing the virtual memory using virtual machine is relatively easy, but managing the IO devices is more difficult, the most difficult part of the system virtualization is the virtualization of IO devices. And particularly, each IO device may have it is own device driver, so IO devices along with that device that has to be virtualized.

So, that means, sharing of real devices among multiple virtual machines, so that is more difficult task, what is being done physical disks are partitioned into by the virtual machine monitor to create virtual disks, for the real virtual, guest virtual machines. So, here the again virtual machine monitor is helping in creating virtual disk, so some kind of partitioning is done, and each guest virtual machine can access that partition part of it. So, the hard disk can be considered as a kind of IO device, similarly the IO devices can be also managed in this manner.

(Refer Slide Time: 51:43)



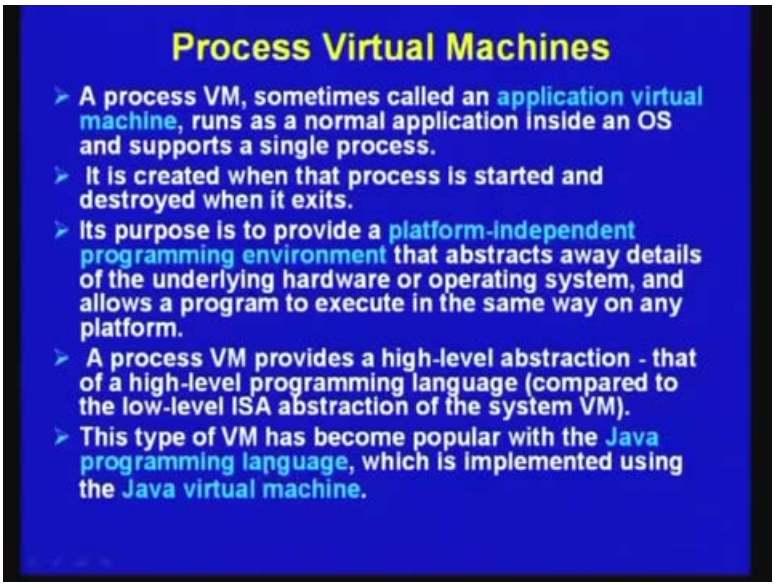
Now, comes the question of protection and the instructions set architecture, as I mentioned protection is joint effort of architecture and the operating system, so it should be done together. So, the architecture must modify some awkward details of existing ISA, when virtual machine become popular. So, earlier virtual machines were not popular, but subsequently when virtual machine became popular, it is necessary to modify the instruction set architecture.

So, that some of the instructions or features is modified, so that virtual machine implementation is feasible for example, that IBM 370 instruction set architecture, had to be changed after 6 years of successful existence. Similarly X 86 instructions I am not going into the details, there is a instruction called POPF, so it is related to access in flag bits, modifying flag bits, so this creates a problem. So, they have to be modified, so that you can implement virtual machine, so there are three steps to improve the performance of virtual machines.

Number 1 is to reduce the cost of processor virtualization, to reduce interrupt overhead cost due to virtualization, reduce interrupt cost by steering interrupts to the proper virtual machine without involving virtual machines, these are two provided together by the architecture and the operating system. And it has been found it takes several generations of processor architecture up gradation to address all the three issues, so the processor

architecture is getting upgraded. So, you are getting version after version, and to take care of a all these three features.

(Refer Slide Time: 53:44)

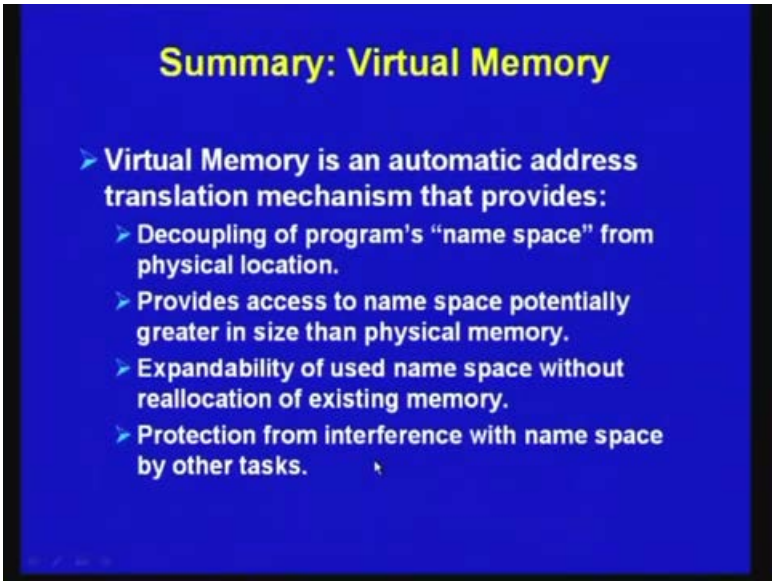


Finally to conclude, we shall discuss about the process virtual machines, a process virtual machines is sometimes it is called an application virtual machine, runs as a normal application inside an operating system and supports a single process, unlike the system virtual machine can support multiple operating systems. So, it is created when the process is started and destroyed when it exists, so it is purpose is to provide a platform independent programming environment, this is very important.

You are familiar with java, which provides you a java virtual machine that provides you a virtual independent, platform independent, programming environment that abstracts away details the underlined hardware or operating system. And allows a program to execute in the same way on any platform, so this is the basic notion of platform independent programming environment. So, a process virtual machine provides a high level abstraction that of a high level programming language, compared to the low level instruction set architecture abstraction of the system virtual machine.

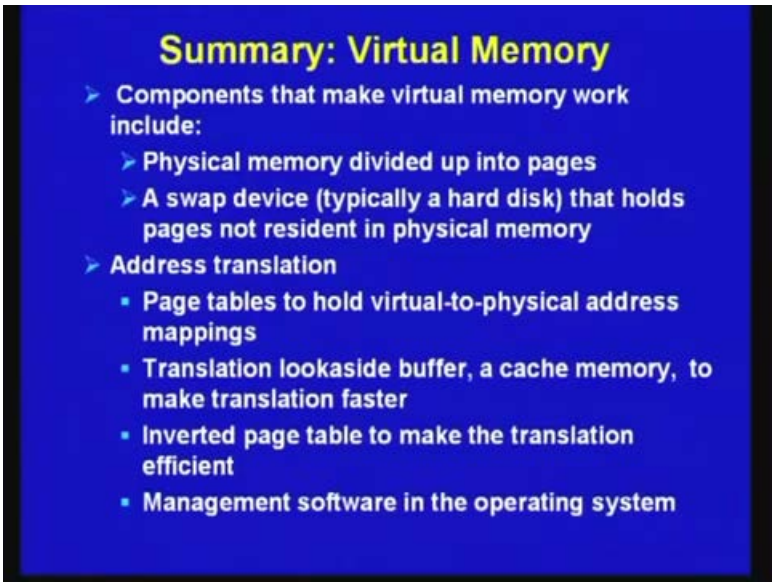
So, this type of virtual machine has become very popular with the java programming language, which is implemented using java virtual machines as I have already mentioned.

(Refer Slide Time: 55:05)



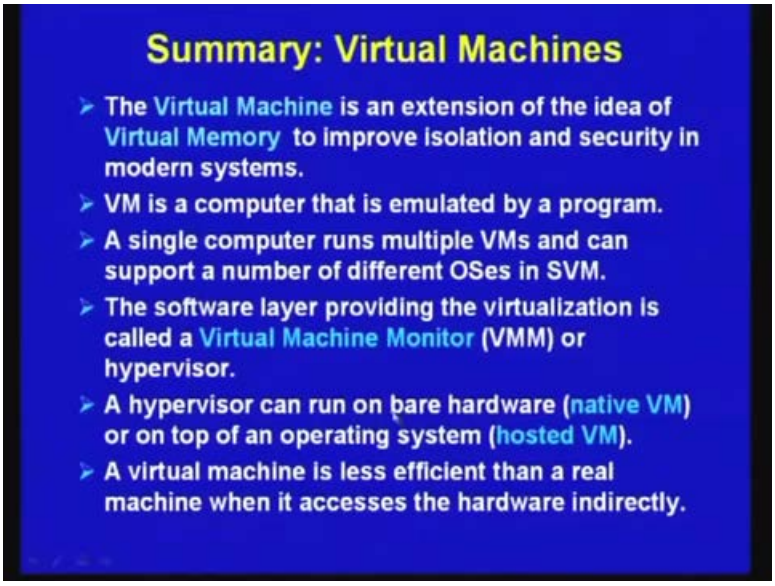
So, you can summarize virtual memory and virtual machine, as we have seen virtual memory is an automatic address translation mechanism, that it allows decoupling of programs name space from physical location. It provides access to name space potentially greater, in size than physical memory, rather unlimited memory size. Expandability of user name space without reallocation of existing memory, protection from interference with the name space of other tasks.

(Refer Slide Time: 55:42)



And the various components that allows it is a physical memory divided into phases, we have seen any use of phasing scheme, a swap device typically a hard disk that holds the pages not resident in the physical memory, we have seen that how paging is implemented. Then the address translation which does the mapping of virtual to physical memory, translation look aside buffer to improve the performance. Inverted page table to make translation efficient, we have discussed in detail and management software is done in the operating system.

(Refer Slide Time: 56:19)



Similarly, we can summarize the virtual machines, the virtual machine is an extension of the idea of virtual memory, to improve the isolation and security in modern systems which is becoming increasing complex. So, virtual machine is a computer that is emulated by a program as you have seen, a single computer runs multiple virtual machines and can support a number of different operating systems in system virtual machine monitor.

The software layer providing the virtualization is called the virtual machine monitor or hypervisor, a hypervisor can run on bare hardware, native virtual machine or on top of an operating system which is known as hosted virtual machine. So, a virtual machine is less efficient than a real machine, when it excesses the hardware indirectly, because of the emulation technique that is being used. So, with this we have come to the end of the

hierarchical memory organization using virtual memory, and also the extension of the idea to virtual machines.

Thank you.