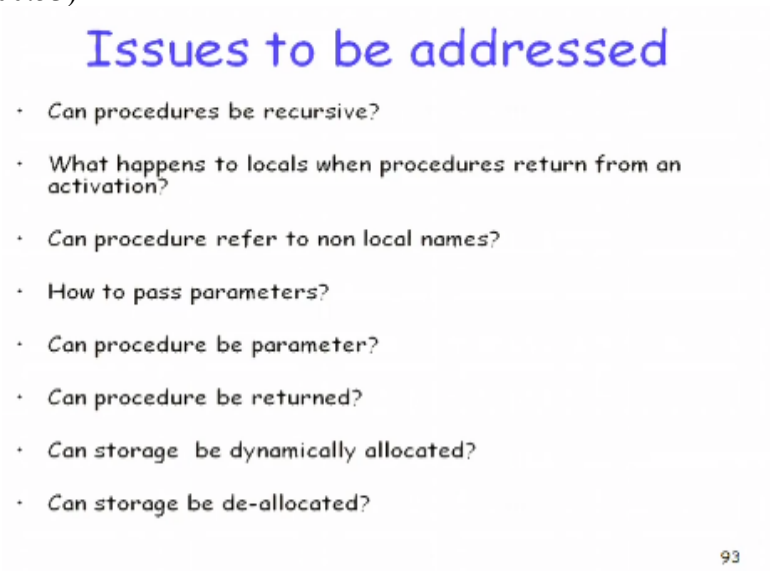
Indian Institute of Technology

Kanpur
NPTEL

National Programme
On
Technology Enhanced Learning
Course Title
Compiler Design
Lecture – 26

by…
Prof. S. K. Aggarwal
Dept. of Computer Science and Engineering.

So let us start with discussing where we left previous class what we start looking at what are the important issues we look at lay out now you want to deigned what are the important therefore in this file I am going to do this issues you can say that always have impact design.
(Refer Slid Time: 00:53)



So first issue are want to first question in the while designing because whether I have so we I just issues but this point of time thank you another important question is that when I return from a procedure what happens globally if you have no common sea of c-like languages what happens local variable when you will return from a procedure or a function in two things we are saying X or – X now both cannot be use simantenously they left has it so they are discarded, so whether contradiction.
So here to take a ,we do not see those variables suppose this procedure is called once again what happens so that means discard so I do not overwrite the memory location but they discarded

exploiting code is different issue like programming language is not safe – X. so what it means is that so when I say available has been initialized yeah if I desired a variable or noise question what happens when I allocate the middle available and say X is going to in this particular location.

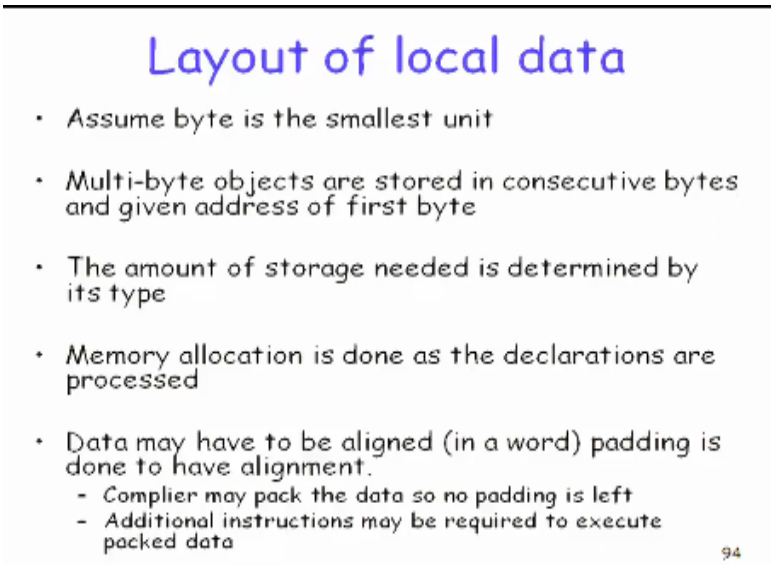
Do I override everything with zero ,zero ,zero bytes why I say this has been allocated there will be some Δ which I think it is two are implementation okay, but we are talking about programming language or implementations so question is from point of view programme language what we say locals and we return from procedure programming language this is no offer available implementations here and have been kind of miscalculations but program language is very clear.

That locals are long a axis symbol if it is calculations that our issue can procedure and if you can what re that mean okay , how do I past parameters they are many matters of passing arguments envier know what is the right method passing arguments so I can design system okay, next question is function of procedure itself so far passing out arguments are expression I can pass a function procedure of the parameter or can I return from point to procedure few more arguments.

What about storage of decoction of storage are some languages sometime when I is rule but if it is do then you also worry about manages this and then we also have relocation so again the question that come is can be decollated or then going to be okay, so these are various questions we have to worst before we get design of a all of these question is going to be impact design and slowly you will see that some point of time.

I am talking about sometimes system or on the fourth generations system are all these question we will get okay, now first look at.

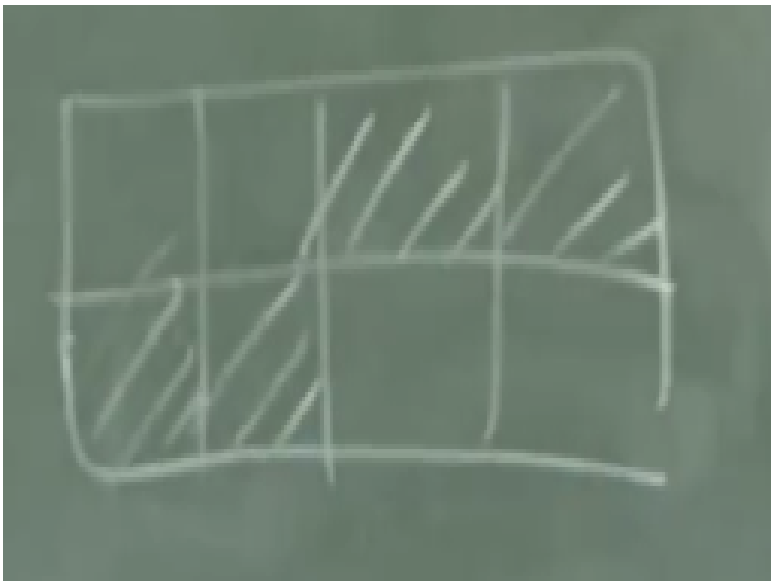
(Refer Slide Time: 04:38)



For the data is been located on the activation local data been that this is the data within a procedure which I am not going to allocate for which I am not going to allocation complications I need to have my core prepare or activation for this okay and what they are going to assume is that smallest unit data now everything is going to be if terms of bytes so I can allocate one byte 2 byte and I am not trying to allocate every bytes.

I can allocate 4 bytes I will allocate 8 bytes so on And the reason for this being is that when I talk about words what should typically either have to bytes older versions have two bytes what all your machines would have either four bytes or 8 bytes and we want to make sure that desired location happens of course and there is an efficiency reason social try to axis free data which is write on the I can do it much faster and if it is not allow what is going to be happen in the same data so delighting the situation like this then I have to consecutive words .

(Refer Slide Time: 05:53)

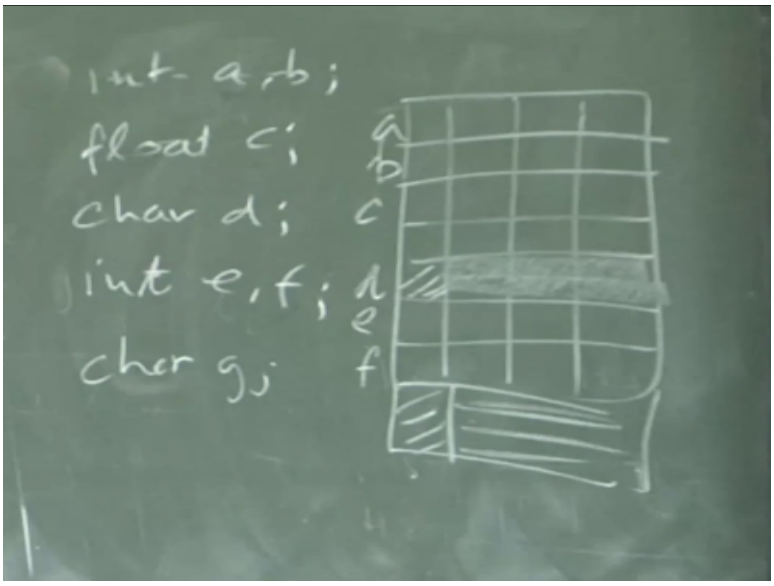


Each having 4 bytes and my data is in these two bytes okay I cannot single instructions so I need two instruction in therefore this is going to be talk more efficient so we are not going to multi-byte object are told and this is ,so I am going to say that multi-byte data I will, start from here whether you have addressing left to right or to right to left then so I will keep these four contributed location and then I try to axis this and the amount of storage so If I see the shorting or 1bytes to this and if I have double procedure numbers.

I may locate so timing is going to determined the and how go 5 and that is an so programming language say that always programming language depending upon the questions so going to get implantations will determine whether is going to be two bytes so in some cases you look at shortage same along and somewhere at again other way so this also become implementations so always memory location is also going to be done as equations of boxes now this is an important issue.

Because what happens is that as I am watching my program and these declarations suppose I say something like this if I say.

(Refer Slide Time: 07:32)



It may B then I have floats C and then I have say can D and I again have let us say so this is how I have written my declaration no in which order I am going to allocate space to this normally in the order if I am not worried about optimize additional space that happens when some compilers but normally what I do is I first allocate space for A then allocate space for B and then for C and D and in this one no one in this order making sure that I can allocate space as I am part of but what may happen in this case is that if I try to align it on word boundaries then I may end up wasting space.

Okay so I have to insert what we know in this example so let us assume that each integer is going to take four bytes this will take eight bytes character takes one byte and again is taking byte and I have a machine which is four byte machine world is four bytes so how will it look for another vision and location will go something like this, like this will get associated with A and then associated with B now two words associated with C now when D comes what will happen this space will go to B.

Which is a single line now when E comes to I start a location from here now the reason I will not allocate from here is that if I start allocating from here then E will fit in these three byte and byte and accessing this all right look E here and I put F here but then you will see that this is my side in bytes on most machines most large kind of general machine this not too much overhead I do not have memory problem but imagine a situation did you say that E problem which fourth a space.

Care out to course problem course right so in that case you have to use different strategies and one possible strategy is like to take all these declarations and sort them in the order of thanks and once you sort them in order of bytes t what do you do then we will say that let me start with the

largest byte and keep on then located what will happen that case C will get the first space thing then at the first this and then A and B and D and F he gets bytes and then these device will come at the back.

But suppose you have multiple declarations for not now suppose I say I will say G if normally in this situation what will happen is this bytes will get a location and expanding but if I sorted then I could have used the bytes and only two bytes instead of six bytes so depending upon final requirements you may do this but compilers also always do not try to save the space very compact space requirement imagine that languages should say that locate a bit okay, a data you are not implementing okay but missing anything yes but posture has a greater size called back their way of pulling data idea is that I will be able to allocate one bit for every day time the piece if I said that I have an array of size 36 then I can fit it in one word and each bit is going to be allocated for each other.

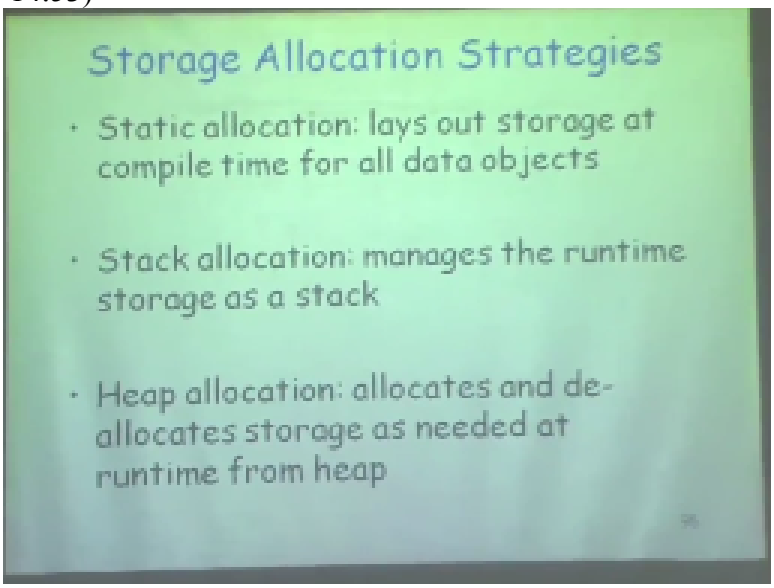
Now this is my decision as far as access is concerned how will you find out what is the bit value how will you find the out in certain a location what end of quote should I but still requires position access okay and then look at and now most posture complain I going to do not worry about it okay ,we have just going to allocate one so program I say data I am trying to say but suddenly complier that is found that memory is not a particular ,machine so I slope sown the course.

And therefore one force every element of the regarded okay so these kind of procedures compiler so any time in C you says I want this variable to be what are that be suppose you declare we will every data item you can give such declaration what will happen ,what will, compiler do physical limitation of how many registers your question may have okay if you tries to everything all the time so it will be moving something memory that is going to be in efficient what is program, advance and allocate whatever is only okay either complier many times.

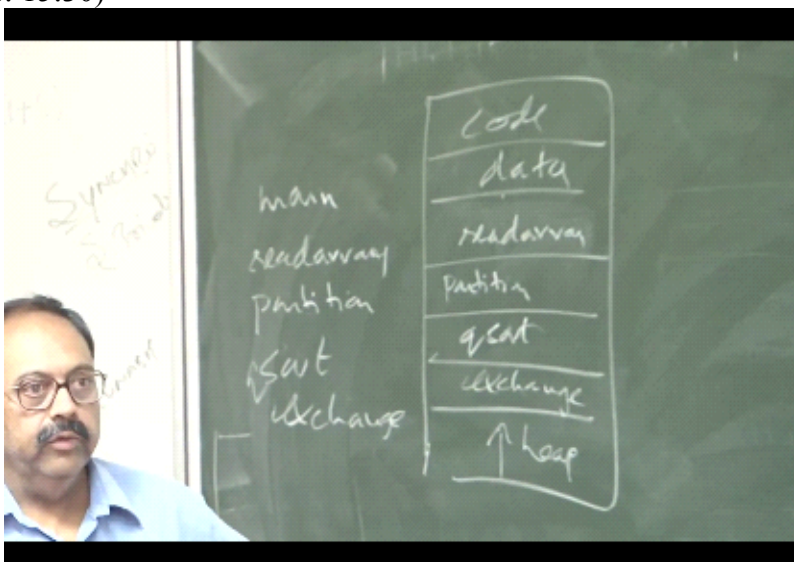
We need you take symbols when people are going beyond then coming languages is specified as long as functionally is dead because many times programming time when you try to decide about the efficiency issues which are really beyond then so for example, when somebody is act any of get up it programmer again leaving it to programmer and the designer saying that this is efficient way of implementing complier it was not happy because of the machine okay, so we have to worry about that so we have to pack data.

So that no Packing is left and additional instructions may be required to kind situation many times you do not want it unless it is really essential on us because there the space requirement is spaces critical now what are the storage allocation strategies so static allocation is one now what

is that static allocation said the connotation is saying that this activation record certainly has to be bound to a procedure and has to be new in space and then another allocation strategy we have seen what activation record has on the stack that is the second strategy.
And the third strategy obviously that you put it on E they can keep is obviously going to be that as and when you require some space you put it on it but there is no guarantee that you can access it either by a fixed location or by going anywhere.
(Refer Slide Time: 14:53)



And this is again a requirement coming from the language size. So let us look at all these strategies and see in which case each one is beneficial so static allocation so what happens in static allocation is something very in dusty okay anyone knows which language uses static allocation of activation records. So first box that education is that you recall I showed you.
(Refer Slide Time: 15:30)



That we had a support then we have code and we have data here then we had stack going and stack is the one because fitting of my activation because then I have keep going here now suppose I say that I have mail program and I had so let me go back to fit for example so the main program was solved and then using that I had a redirect and partition sort and maybe add change right these are the functions we add okay.

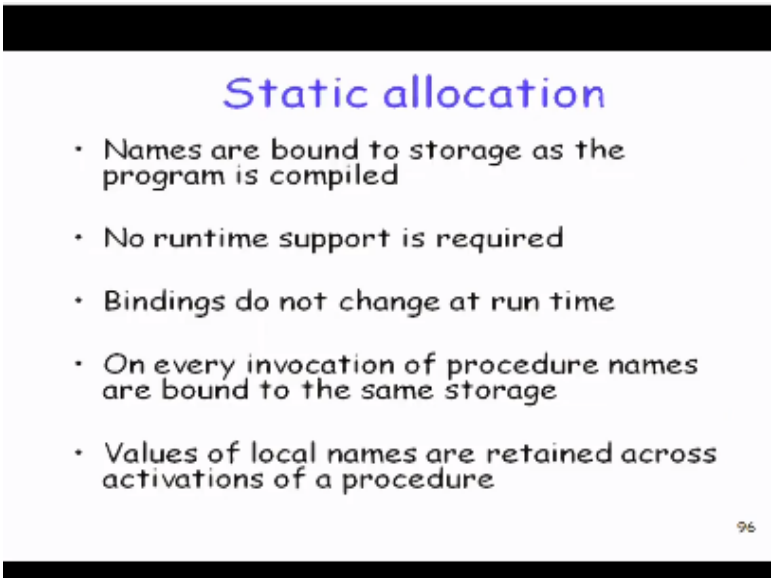
Know what statically allocation as we do you say that do not worry about this back part and let us say that mail is already the global data then I have an activation of redirects I had activation corresponding to partition and these are the active agents which are already to each of activation what is the advantage of this and what is the disadvantages so what is that one image of this model or does it happen on clear advantages.

So what is the disadvantages that compile-time I know at this of each of the activation record okay were us inside at one time I will determine okay so if I am thank you axis is the normal variable I know right away I do not have to have any pointers on the symbol table I know right away that this vehicle is going to be located in some instances there what happens when I say that I am my control is going from suppose.

I say my control is going from quits or who needed and then V direct definition what will do for this case so first function was in mail program by direct there after redirect the information is wanted are doing started location I be a located what will do insatiable location. I am okay with what you are doing static a location nothing it remains there that means if I convey once again and all the local variables will be available to me and the old values will be available.

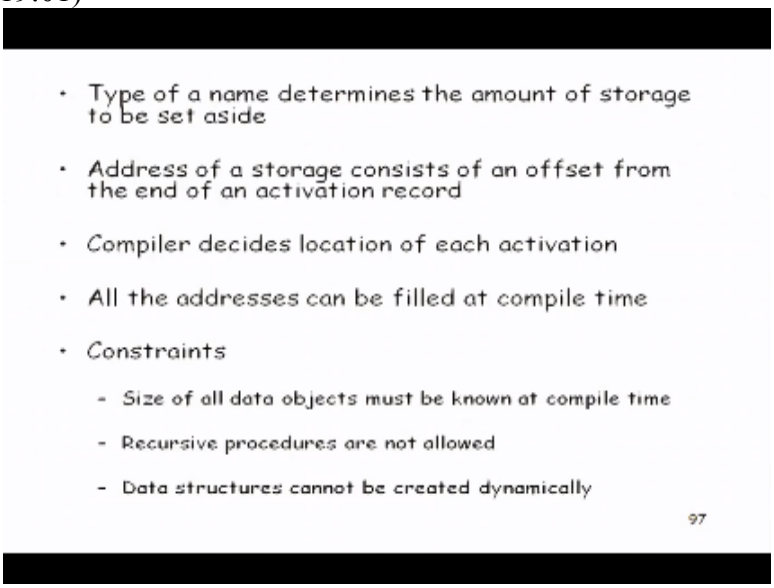
So I can keep track of a local variable how many times this function has which is not possible in a static allocation so names are bound to storage great for any runtime support which I need removing the stack allocation and when the profile in my program is equals these bindings do not change that means every time I call me Derek this is going to be the bomb fine this is where my reader is always done and for every invocation the procedure name is going to be bound to the storage,

Refer Slide Time: 18:18)



That means every time I call reduce a read array of quick sort of partition on quizzes binding to this location but this puts a very severe constraint what is that constraint I cannot have the cushion right so value of local names are again across activation okay and type of name,

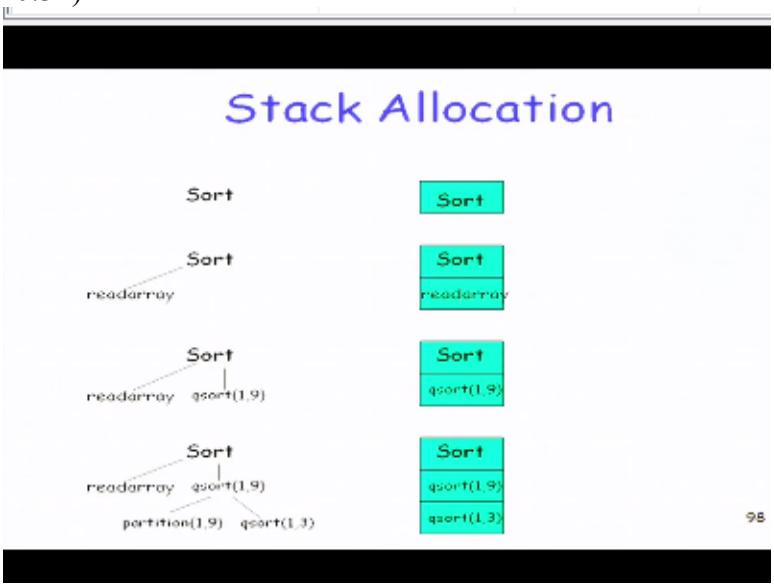
Refer Slide Time: 19:01)



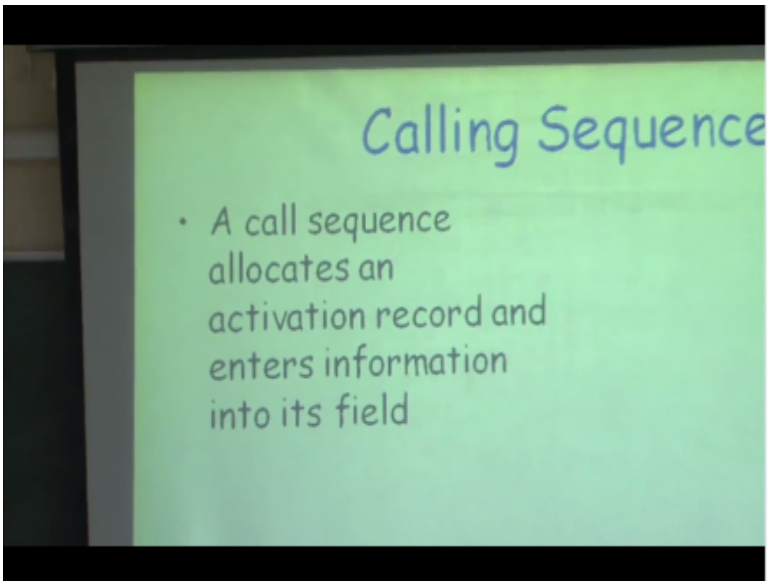
Determines obviously the amount of storage that has to be kept aside so some state is going to be sign of data object must be so because of poor generation register education almost everything happened after the first compiled opinion. So first compiled the people who are able to functionally do something it whatever they talk about protect that point of time similarly I mean you look at column - major is going tom more efficient or column-major is the one.

They picked up and later on standard provision what happens that your location is going to be where I say that these are not to be allocated space at compile time but I will know the layout of

activations of each of these at compile time but actual space allocation will happen only when program started so what happens here so typically when I say that,
Refer Slide Time: 20:34)

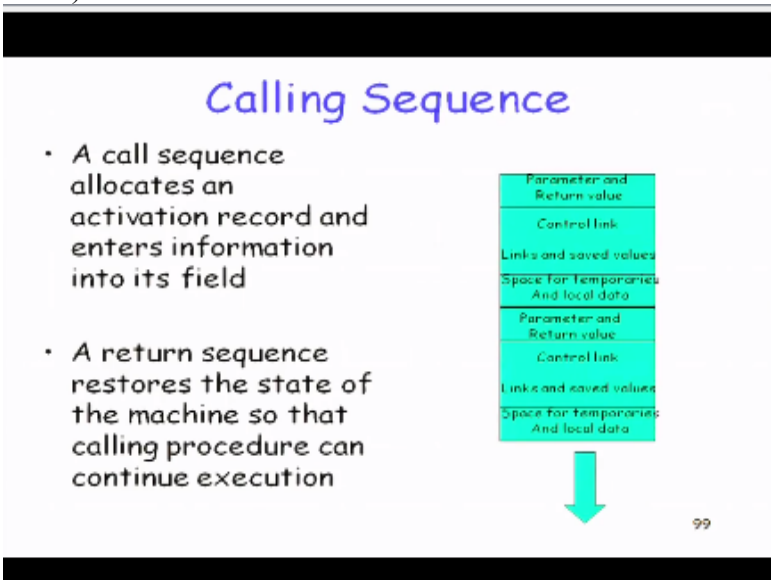


I have made program of the sort program this activation goes on the stack and then if so thought all three read array then this activation goes on the stack and after this a free read array finishes and mi- online then activation of delay rays come from the stack and it is off on items on the stack unlike here where we they will always remain and then you find somewhere here in my country and this obviously is node at the top then the only nodes which will be able on my static assault with sort of alignment with sovereignty and read array partitions Online had already be so then what happens in stagnant location but what is the fall sequence because inspect hello patient what is going to happening.
Refer Slide Time: 21:19)



That somebody will have to not put data this data is not going to be known to me as compile this data is will be initialized at runtime to generate additional code for managing my activation score so what happens is that I have all sequence which is going to allocate activation information in this video so I actually show you who was and after we have looked at all these digits that of the port for initializing an activation record looks and return sequences when I am going to rest from the save of the Machine office automation.

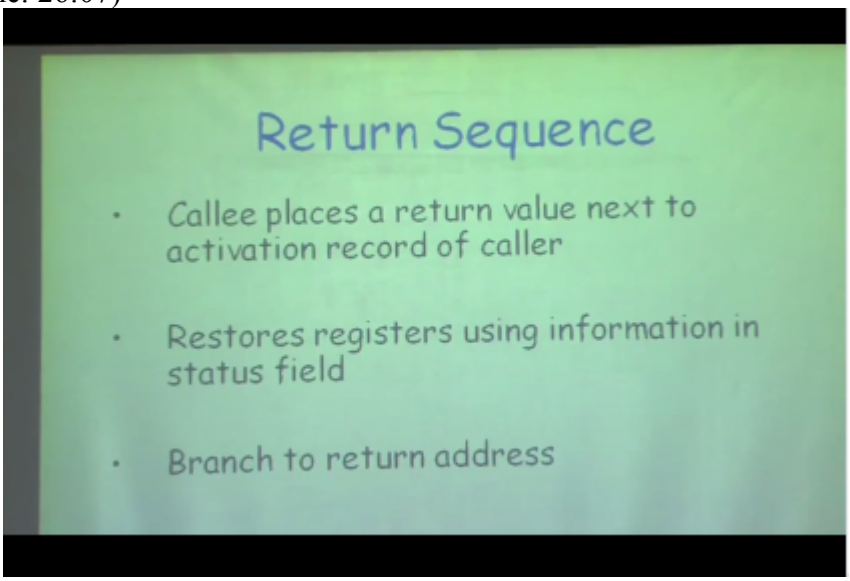
Refer Slide Time: 21:54)



So okay this suppose I have now two activations and this is where my stack is growing so stack is going in this direction this is the activation value because this must also have any go fight someone so this is like a snapshot of to quantitative activations in my step so I have these so you

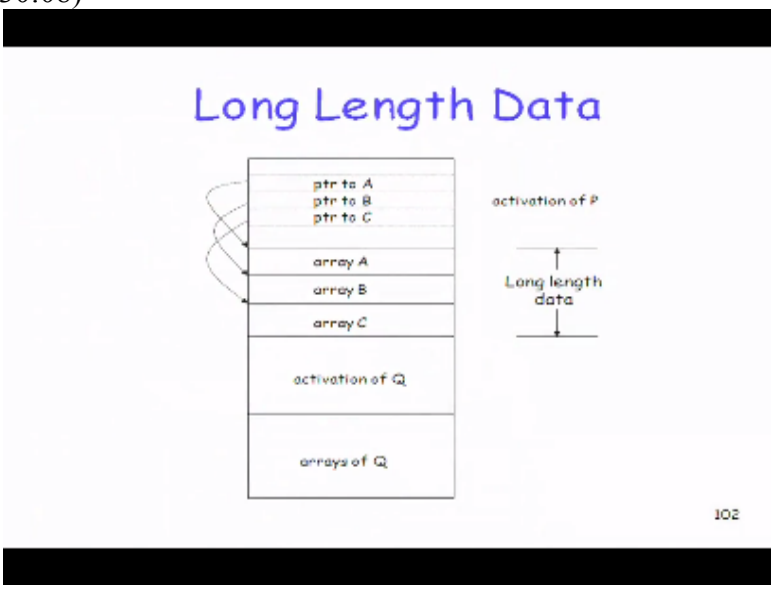
can see that these are exactly same only the size of data may differ and I have this control lanes I have excess links and that I have space for temporary and return this.
One now first question that comes is if this is the caller and this is the body to freeze in this data so when I say that who fills in this data I now am going into detail of saying that if some data has to be initialized is a job of the caller should that poor with part of the caller sequence or part B see here show that Part B should that poor be part of the policy now starting from here a so this is the orders activation record this is the normal caller.
Starts in now who fits in this data parameter min return back all the space for the return rate who fills in for is in this particular activation who are still missing so that must have been a activation of which reach like this obviously you say a calls me that a knows what the family does with you without you it so this must have been filled by the order of this caller similarly the controlling weather control has to go back it must have you filled it but if you come to this spot it this must have been filled by the caller.
Which says what are the links and what is the machine skaters crying I am saving and of a temporary space I want so typically what happens is that this point is going to be filled by the column and this part for the activation of the body is going to be filled by the pond in the same manner as this will partner with so many both in this structure it does not mean that all of the activation is to be filled in by this particular procedure.
So therefore not you have to see that can i generate code my code sequence when I say call sequence and return this sport must be part of the code sequence this spot must be part of the top all of sequence and this bar code must be part of it functionally you can see that this is where I have information from the caller and this is a sense so this is what we want actually start doing for code generations that will spark beginning on this day so this is okay.
So there is the calling responsibility so next look at what happens then actually we go for generating code so what is my all sequence what is going to be when a collector that it has to supply all the heaven so see this in that all this water park has to make that only family there ends up then we are there so I have all of the kinetics of the actual parameter then all and also stores the return address in other branches which is circling to the bodies activation record is telling only that this is the variable.
And then caller is going to save all the register values and call is going to save all the register value when this information before it starts it and once it has done that that is going to initialize its own local data and it's going to that anything so there are certain responsibilities in the call sequence which have to be done by the procedure routine books and responsibilities have to be

taken fighter closely here which gets in here and what is the return sequence then now what is going to place the return value in execution.
Refer Slide Time: 26:07)



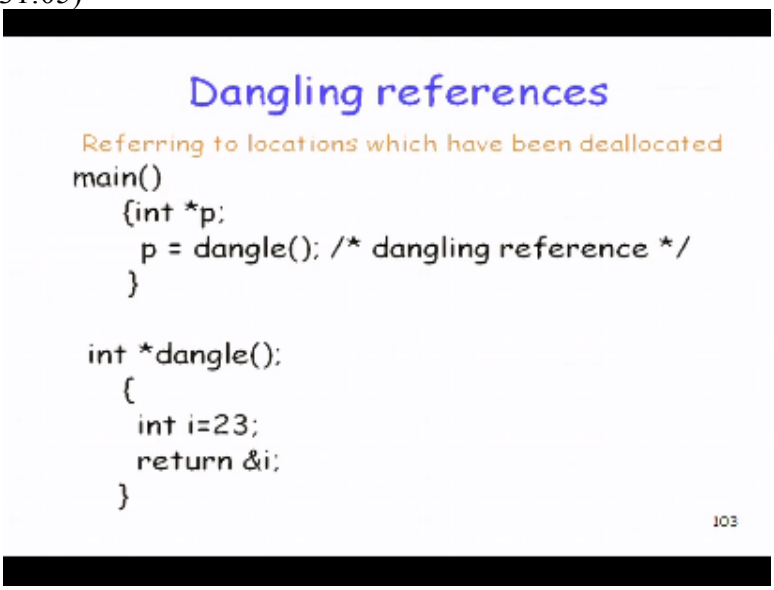
Now if you see something very interesting is happening here funny when I say that I have these excavations. So suppose I have situation where a cos B this is a and this is B and sequences they cause B now offer me finishes here assumption is that this mode of the this activation is no longer accessible to it but if you see this instruction this is saying all these listening the return value next mission body is missing the return value here and is not putting it in a why do that a is putting in location immediately next foot 5 is not fitting.
Somewhere else register nothing stops me from doing that I can put this any advantage of this is that volley when I say that I am below the elbow fitting for this fish water in system I can use it for something else that does not mean that I am going to initiative of the big sequence is 15 it means when I put my return value here if this is my stack pointer so earlier my stack pointer was here when I be allocated my stack pointer becomes this and then I say whatever is in the stack pointer plus one morning early one that is okay.
So before this code of a resumes execution it is going to now so I can very safely now why do I keep in the location here and not anywhere else the reason is very simple anywhere else is going to be arbitrary because every time for every activation record that offset is to be different here it is saying that offset is always going to be stack point what was fun exciting irrespective of which procedure is being so then what we do that once it places than you hear it has source the register values branches to the return address and different addresses for the control signals.
And then the caller is going to copy written value activation forces back the collar we have some poor sequence where this value will be called for feedback and then execution this time now

suppose that when I am talking about so when I am talking about activation record one issue our content this and that is if I go back to my website ever tried to and allocate space to this what will happen to my Activision for my activation record will just become billeted now I got one that exhibition report will be very weak and one is the reason for index or what is the reason for not having a big activation.
Record change and the simple thing is that normally what will happen is that I have a pointer which is suspect concepts and then I have a pointer which can be pinpointed so I can have pointers at two sites and I want all my data which I am accessing should be in a small range 0.0% one if the distance is lost and what will happen often you take will change even if it fits in a page what can happen then offset is not in therefore it becomes slow so normally what we do is that when I have large bonds over data in the last local data then it would it activation.
Refer Slide Time: 30:08)



So what I can do is I have this activation of P and then if suppose I have these arrays a b and c which are set on there is I push this is outside the activation record but now I am two additional bookkeeping for my stack pointer that means my band my stack pointer was this normally when I increment that pointer is a set point is going to be stack pointer plus size of activation now in this case it will be stack pointer plus size of local data plus size of activation for additional bookkeeping has to be done I have to remember that I have created some local data here.
But my activation capacity and another related thing that comes is that you reference so basically sometimes fearless and they try to access a location which has already BPM so what may happen is a situation like this and the common situation you see that we have a local variable and then you return able of this local variable.

Refer Slide Time: 31:05)



And as soon as you return control technique and you what happens is that this space as p pl okay now suppose after some time we will reserve this pointer and then you make another procedure call and the same space is located to the next function. And what will happen in that case when I try to access this data will have how many people unless you know of stack education happens we know fear is what is wrong with this but once you understand stack allocation.

And immediately you know this trying to gain access to something which is knocked over trees this also becomes a loophole where all these viruses and stack overflow and so on menu correctly but as a programmer in the checks and balances which are possible that you can push your program through these data may be safe or you are trying to return the but there is something even as a programmer.

You should not do and you can explain it only by that inbox - it's like a location if you show it to us in programmer who does not know the runtime model then you find that they do not even understand what is wrong so I think today I have to stop here and today we need to take a import rear array okay and one of you will have to volunteer come forward and do this must be that will what is here for the course be back.

Acknowledgment
Ministry of Human Resources & Development
Prof. Phalguni Gupta
Co-ordinator, NPTEL IIT Kanpur
Satyaki Roy
Co Co-ordinator, NPTEL IIT Kanpur
Camera
Ram Chandra
Dilip Tripathi

Padam Shukla
Manoj Shrivastava
Sanjay Mishtra
Editing
Ashish Singh
Badal Pradhan
Tapobrata Das
Shuubham Rawat
Shikha Gupta
Pradeep Kumar
K.K Mishra
Jai Singh
Sweety Kanaujia
Aradhana Singh
Sweta
Preeti Sachan
Ashutosh Gairola
Dilip Katiyar
Ashutosh Kumar
Light& Sound
Sharwan
Hari Ram
Production Crew
Bhadra Rao
Puneet Kumar Bajpai
Priyanka Singh
Office
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari
Saurabh Shukla
Direction
Sanjay Pal
Production Manager
Bharat Lals
an IIT Kanpur Production