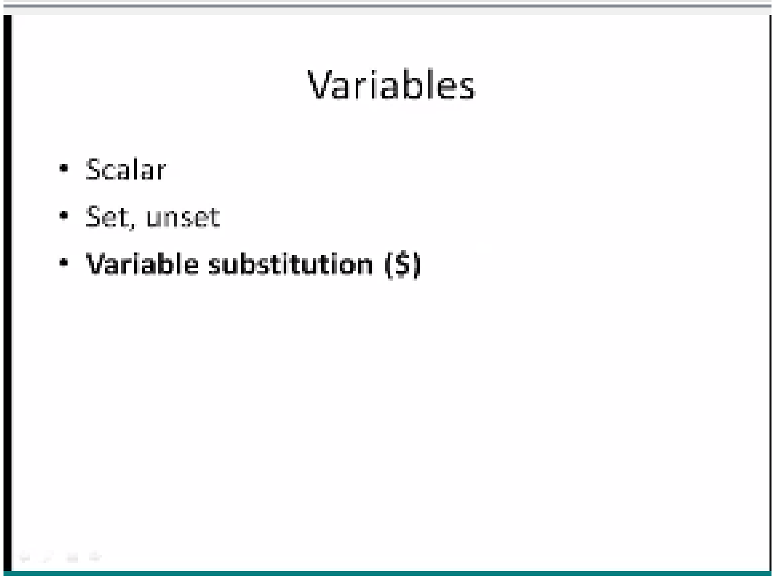
Tcl in Synopsys Tools

Hi everyone again welcome to the LPS class, we are going to continue our lecture on tickle if you remember like I mean in the last lecture we talked about the TK predominantly. We completed the DL we is a more advanced the tickle processes basically nice to how one can load in tickle has a dot a sells and called tickle functions from anywhere things like that that we talked about today. We are going to take another step in that direction but mainly we are going to talk about tickling Synopsys tools.

These are the tools that you use for your labs and also like I mean in real world we will be using a lot and synopsis is actually modified tickle a little bit for the better. So we will be learning about how to use tickling Synopsys tools and what kind of support that they point because tickle is not only their language but language of choice but actually it has a lot of good things basically. So today we will be talking mostly about tickling synopsis tools.

Which is mostly an extension of the existing people so I will be talking about a lot of the concepts that we already covered just to reinforce? So that you also know what it is?

(Refer Slide Time: 01:43)



Variables

- Scalar
- Set, unset
- **Variable substitution ($)**

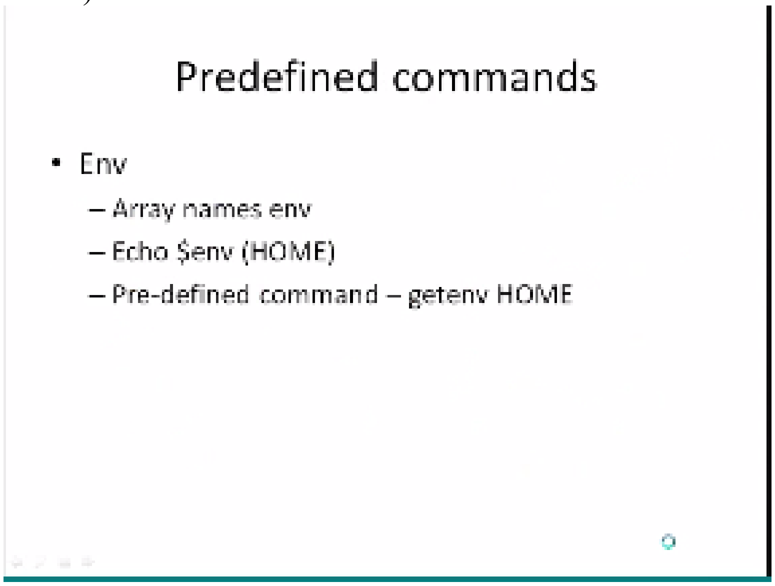
So number one thing as with any other tickle we have number of variables the variable names are mostly like it there. The scalar names this way then you can simply say any name it can become and variable name and you can assign a value to that variable by using the set command. So simply accept a1 will be same as a suitable a will have a value of 1. And then these variables these values can be integer, integer or floating and then we will soon see like them in other variable extensions.

These are some of the things that we already saw like I mean very first lecture of pickle and then one of the key concepts that we also talked about was the variable substitution. That is also valid so once you have this and then now the next case set B $A. So that D you will also be = 1. So those are the things basically that you can do and other ones like. The various other functions and in fact that I am also like supported like incremental or additions and whether the info exists kind of things.

Basically in progresses essentially like that is one thing which returns if the variable is true people exist it returns a 1. If the variable exists and if it is a it does not exist it does not exist in consider oh and then you can also have info bars and here you can give likes my Ponte X star. And this returns all the variables beginning with X and then the president rules are very similar to the regular little groups. That means that any start with an integer if and then as you go to further operations may be to this result states.

As an integer and then you gain divided by 2.0 now this becomes of mode answer the 1.0. So and then if you use the float and the first one the result will be a float. So those are those kind of things they need to be aware of you then one thing that synopsis provides is also set up three predefined variables.

(Refer Slide Time: 05:09)



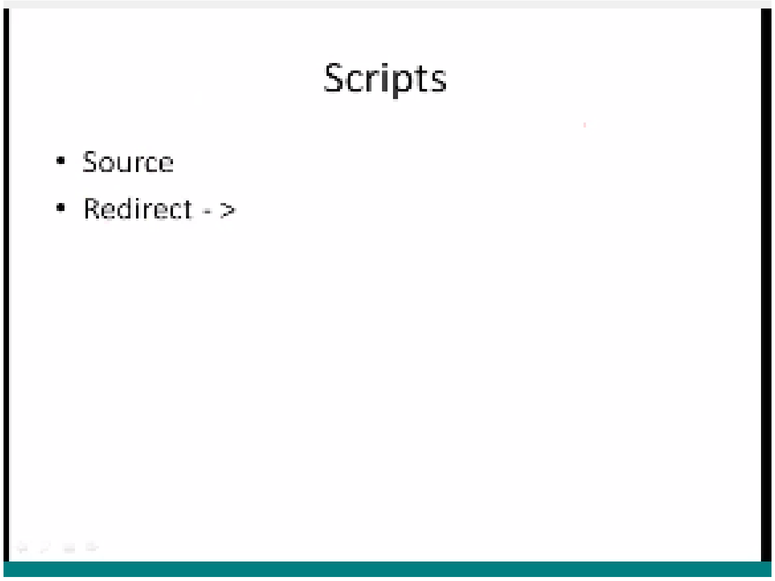
Predefined commands

- Env
  - – Array names env
  - – Echo $env (HOME)
  - – Pre-defined command – getenv HOME

One example is the Env, so Env is a variable that contains all the Environmental variables. So we can use commands like array names Env. This command will print out the list it contains elements even the correspond the name of the Environment and then to show like a particular Environment variable within that you can also use an echo command. Echo doll VMV home will print out whatever the value that is set as home. And the synopsis also provides a predefined command called get Env.

So the get Env command is very simple this is Env variable name and then it brought it dumps that particular variable. So home has whatever it is that is dumped out. So these are this is the nifty command that you can use. Now let us go into script basically.

(Refer Slide Time: 06:31)



Scripts

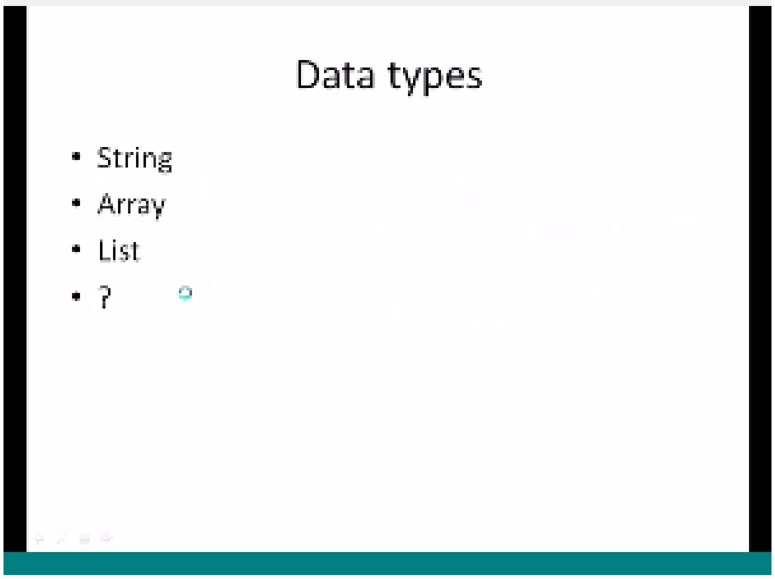
- Source
- Redirect - >

What is the script the script is nothing but it is a tickle. So it has a ticket commands, and synopsis commands. So these are only grouped together and so you can run like some synopsis command to get some data and then process it through using your failure tickle commands. And then also like apply that back into length not system using box so that becomes part of our script. And along with the commands and scripts essentially legging you know that already the hash if we use it that for comments.

And this can come anywhere in the line, and the only thing is after the hash everything in that line will be treated as a comment. The only exception is if you are extending the comment using the backs the \ arrow and use the \ arrow even the next line whatever it is that is treated a second as you know this is the continuation character and basically this and any kind of specific that will replace with moving so this has a new line to it that you proceed that we replace the sister blank and that is why this line will be then continuously been excellent.

And how do we run a script, we use the command called source so for example source my dot tickle. My article can be skipped the same things all these typical iron form with lots of comments and we simply do a source to run the script. So once you do the source you can redirect using greater than symbol. So this is all the standard ones that you already saw bots are these things. Now let us go to the next one which is the data types.

(Refer Slide Time: 09:28)



Data types

- String
- Array
- List
- ?

As we have seen in the regular finance it also supports the three basic types of data. Which are strings array and list, the string is nothing but a sequence of characters. And string can be within codes basically in embedding strings within quotes or you can have like this generic the strings of distance some letters. The strings are usually operated on by a string command, so string command takes those many arguments essentially.

So for example string and then we can say compare, here we do the first string the second string. So it compares these two and if they are same it generates a one so I send it to zero and then string - upper and then to name this will convert the string from lowercase top piece. So similarly there are many operations that are possible the main ones are format rag X Rex tub. Which is essentially it uses a regular expression to do a substitution also you can do scam like a screen.

Scam or one word and then string, string or it is a set of string manipulation functions and then string subst if you have subst which is for substitutions. So this will be a pure substitution a reggae sub will be based on a regular expression it can do a substitution. So these are the commands that synopsis will support let us serling list the list is nothing but an ordered group of elements so it is usually impossible rabbit.

And we can also use a list command create a list, list and then set of elements will be turned into a list and you know like I am in the command substitution on the [ ] command substitution. And then I also think that you remember in tickle there are grouping litigated, for those grouping all the with the braces and then more strict grouping with the arm coats. If it may be like doers but the main difference is that here when you have like a variable $X it does the variable substitution.

Whereas here it does not do any such situation they are preserved assets so this is stricter and this is looser okay. So based on that I mean you can actually group the list elements though so if you are anything under the inside the brace history class one object in the list and then the others will be subjected out also a list has index the dist L index you can do L index and then give a number and then to access any of the elements and it usually it starts with 0.

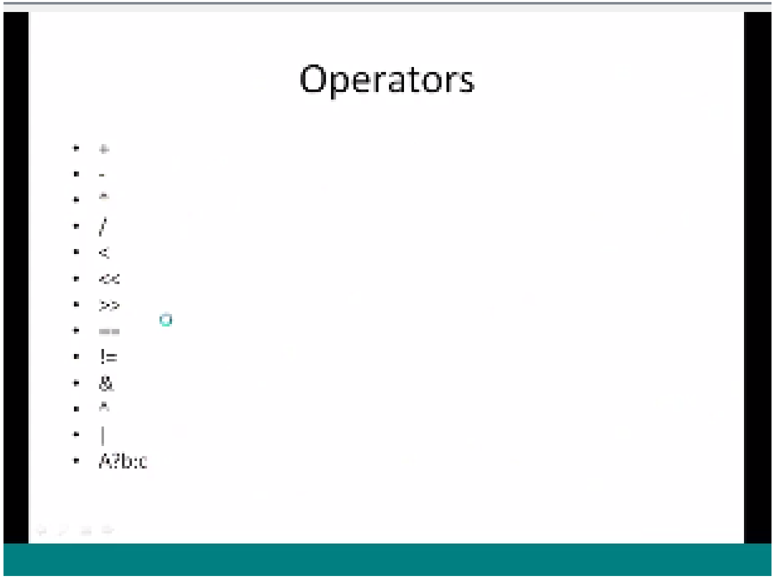
And then it goes to end so L index end will give the last element and let 0 will give you the first statement. So these things they do not change other list operations are contact join L append in Lindex. That I mentioned LLL insert then link or L link, L range arrange, it also legates a very useful operator. It is basically it extracts the elements in a list within a given range I will replace L search L sought and split VV went in much more detail in cover.

I think like pretty much you can use that so the tickling traitor within the synopsis tool can work with all these different steps. So now become to be a race that is this element here so array is the tickle it uses the associatively mobile basically. The index does not matter it can be real number it can be integer can be string whichever one that we can use. So typically an array name is we go in it is an array name followed by element.

And usually the array will have a two dimensional lookup table based dealing one is index number on line it is like a key value pair on think of it that way. Now tickle also supports the EXPR. So you can use the EXPL as well for evaluating any expression, so the I recommend particularly array and then similar to the, the list commands. And I have put a question mark here there is a fourth data type which is what is what makes synopsis rules with special.

Now we will talk about that towards the end of this lecture this is called selection. So collection is a data type in synopsis tools you can actually use this the very, very useful data time. It is kind of list, but it is even more advanced than a list. We will look at the term as to how to use it and how actually between uses.

(Refer Slide Time: 17:37)



Operators

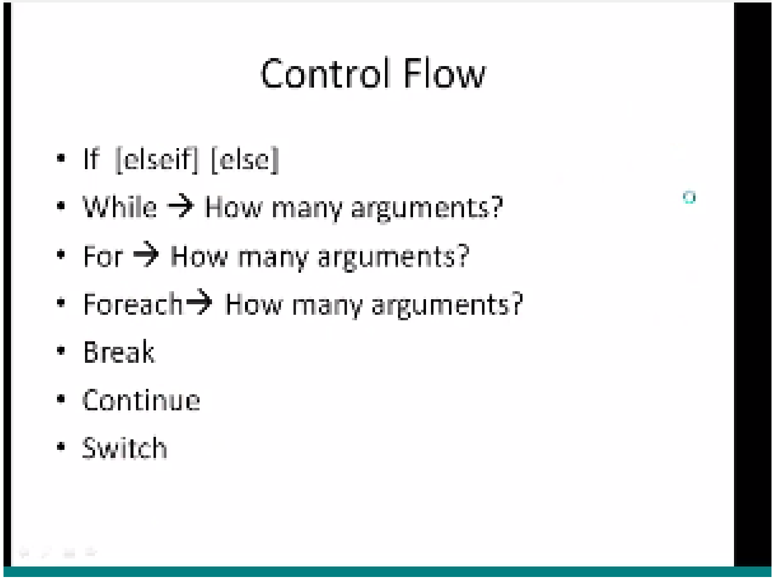
- +
- -
- *
- /
- <
- <<
- >>
- ==
- !=
- &
- ^
- |
- A?b:c

So operators essentially I mean because this mainly goes back to that what I was mentioning regarding the expressions. Expressions are, given by EXPR and then you can have like one argument which is enclosed in the in the braces inside that you can have like a B times C +B. Whatever you want, so maybe it was wrong at all something so the term substitution as well. So the operators essentially like knowing that you can use within the EXPR command are +, -, ×, ÷, <,> this is basically a shift operator.

The shifts the value to the left by so many bits so if you select A, BA it shifts B then the number of bits as to A. So it is to left by two and it is a shift right operator and this is the equal to as a logical operation and not equal to a logical operation. These are also linking, so these are the relational operation value but the desire bitwise term and this bitwise XR it was R and this is a conditional operator.

There if, if A is non view then take B otherwise take C so if you say like set X a question mark B 3 if a is 1 then X is B if a is 0 then X is okay. So the all the other operators basically you can also use in logical operations instead of bitwise and the logical operations typically use inside some control flow structure. So let us see like what are the control flow structures in the synopsis tickle.

(Refer Slide Time: 20:41)



Control Flow

- If [elseif] [else]
- While → How many arguments?
- For → How many arguments?
- Foreach→ How many arguments?
- Break
- Continue
- Switch

So number one is the if-else, if-else here you do not have to have these both these things it mean this be if so if as two arguments. One is the expression and then the second argument is you now you can see that I am putting greater for that term. They can work they are gook f12 in an else in case we specify the expression say expression 1. And then the execute script 1 and this and then we can give exposition to and then flipped to and then finally else. And then this good three so you can see that actually like it can flow nicely.

Because all the other faces fear so when the if comment evaluates the expression it is the expression result is not a zero then. If it is zero then it just goes to the next form and this else is can continue on and on now the while command. So quick question how many arguments do you think it takes? So the while command has two arguments this similar to the is the bare bone is command it is a while then condition or an expression in body. What is the subscription so we had two arguments for why how about far for is something that.

We studied also how many arguments you think far for it also another way of looping structure. So we looked at before four has actually three arguments actually four arguments, so the first argument is the initial expression, and then the second one is the termination, the third one is we in it expression. So what is this we will talk about that and then finally the body or the script itself so the Viet expression is basically the initial to the termination how to go about doing that?

So in a simple case like you can say initial can be set X 0 and then the termination condition can be all of X < or = 10. And then the unit expression will be increment X increment $ X and then you can have the board. So that that particular loop will be repeated 10 times now comes for each how many arguments do you think for each takes this one 1 2 3 4 automatics for this takes 3. How many do you think for each it takes actually 4 it takes also for it takes actually 3 arguments actually like this is 2 arguments?

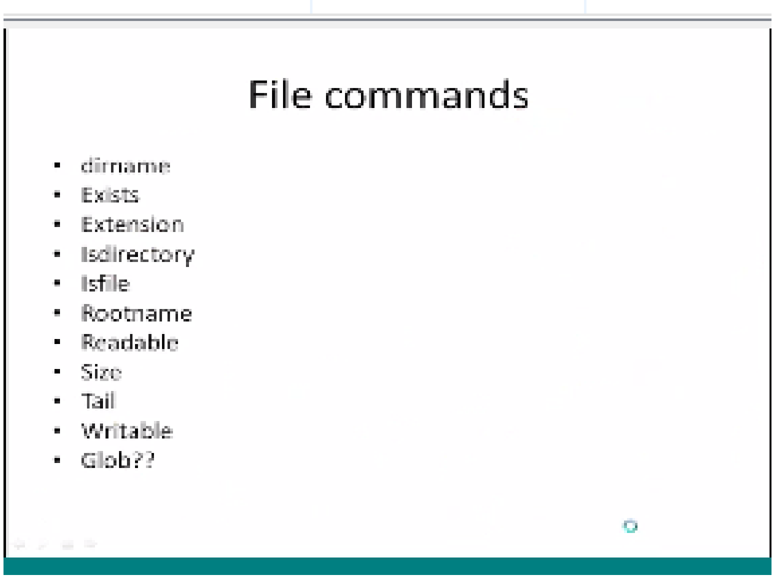
While takes 2 arguments for take 4 & 4 which takes 3 so what are the three arguments does it take. So it takes a variable then the input list and then the body so it picks up one variable at a time from the list and then runs the body for each one now there are two other actually three other controls three other control flow commands. Which can actually cause changes in the weight of all the so one such command is break the break command. Will cause the innermost loop to terminate as soon as it hits the break and these can be used to actually change.

The flow for, for each and now the next one is the continue cannot continue command actually causes the innermost 44 innermost loop to stop but it continues from where it left off it would not terminate. The whole thing so it causes the current iteration of the innermost loop to start to terminate but the next iteration will still continue. The Candida is kind of it is more safe or kind of command and compared to big but you may have situations where you won you can only use one or the other they are not interchangeable.

And sometimes like the big would not fit in there the continued fits in and vice versa. Now the last one is the switch command, the switch commands how many arguments it takes actually believe it or not it takes only two arguments. So switch as a variable and a whole bunch of value and then split. This entire thing is actually enclosed within one place so you can almost say that basically or given variable if it is of one pattern then do one script it is a different pattern then do seconds things. Like that I need progresses I mean until you have no more patterns and scripts the switch command also has three options.

That means we will talk about it can be exact match or a globe match D LOV this is not to be complicit block an operator this is a globe glob match is essentially kind of a loose match and then finally a vaguest explicitly. So these are the different options that provide or the switch command. Now let us talk about the basic file commands.

(Refer Slide Time: 29:03)



File commands

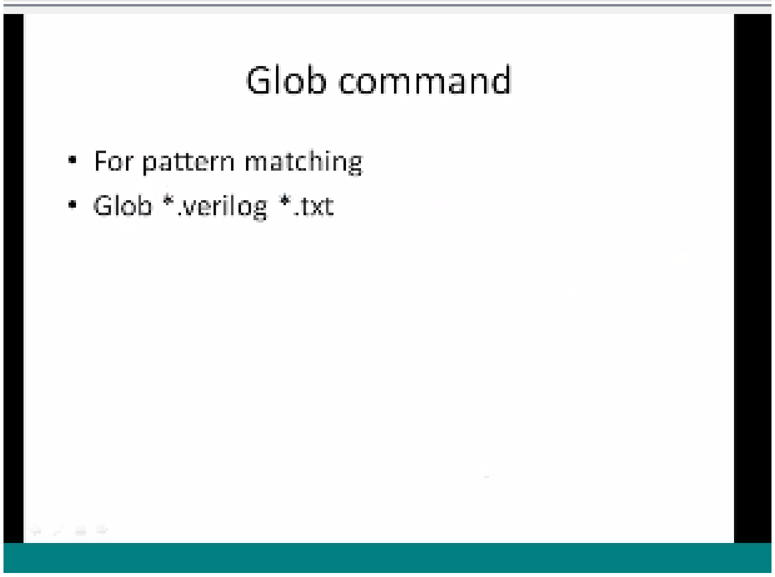
- dirname
- Exists
- Extension
- Isdirectory
- Isfile
- Rootname
- Readable
- Size
- Tail
- Writable
- Glob??

So before we talk about these additional commands basically CD and PWD I think you guys know that CD text and father change directory. And then PWD is the person working directory those are pretty much the same when you can use that inside synopsis so tickle command interpreter. And now let us look at the file commands and there are two of them basically like meaning like though we can use two different types of options here, in the file or glove and Bob you know that basically match the video expressions in this context.

So the file commands our directory name to get the directory part of the file and then you can also do file exists and then given a file name. Which returns 1 if the name is 0 if it does not exist file extension is give the extension part of the and /or text it returns the text and then each directory is a command that returns 1. If the file name so thread is a directory and 0 otherwise and each file also works the same thing the returns 1 if the name next to it is actually a file that exists otherwise it returns a 0 these are all like fire testing come on.

Then readable is another one which is written if the permission serves that such that the pipe can be available. I leave it and then the route name is another one which is use the root part of the and the file name. Size gives the size of I mean bytes and then file tail file name will return the finally some part of the front part string. And then you also have like writeable which are basically written someone is writable or 0 otherwise. Now the globe is actually like I put it put then basically the globe matching is something that we talked about in this context.

(Refer Slide Time: 31:53)

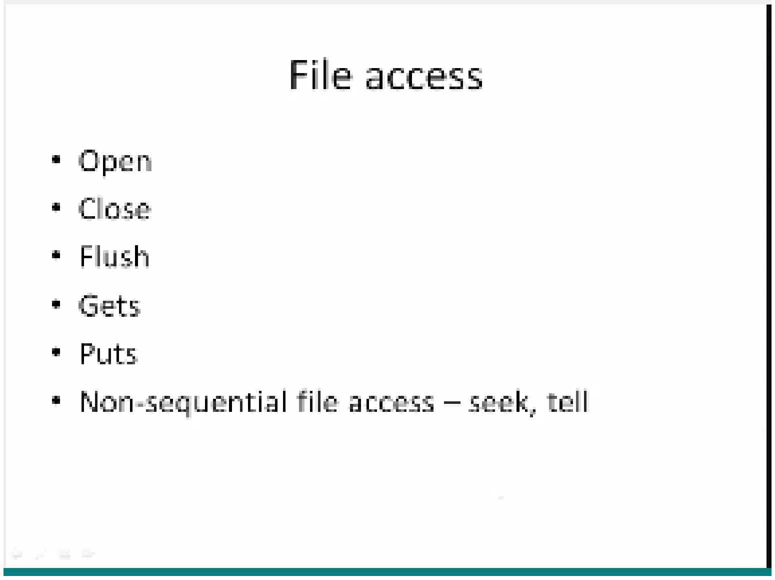


Glob command

- For pattern matching
- Glob *.verilog *.txt

It actually returns four files it is a globe command it is mainly for pattern matching and the reporting files. And globe you can say like start or day log and then start a text this will to the collection of very log and text files. Okay so far so good, you let us continue on.

(Refer Slide Time: 32:34)



File access

- Open
- Close
- Flush
- Gets
- Puts
- Non-sequential file access – seek, tell

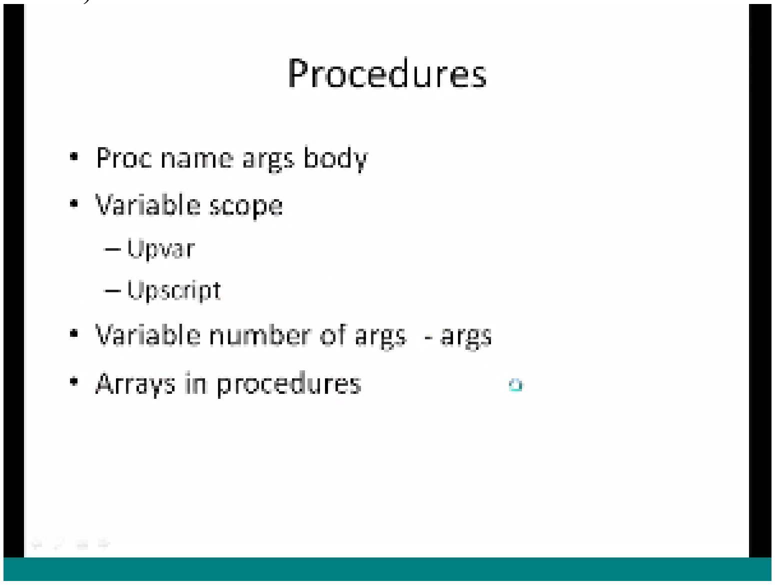
So file access commands essentially because for all can you open a file and access actually the first command is file open command or just going to open here, you can stay like what is a file and then what is access mode. The access mode can be R which is just really R plus it is used for reading and writing and the W it is a open only for hitting W class is open for reading and writing. And if the file exists it truncates it otherwise it will create the type then A is open the file for writing only.

And then the new data is appended to the side so the existing did ice-capped and then the new data is just getting into here the only requirement is. That the file name should exist equal create the body B file and a plus B is the same thing only for now you can use it for reading and writing and here. If the file does not exist it creates this file and then a new data is also like appended to that similar to the priest. And to close a file we just specify the close command the close followed by a file ID $file ID will close that particular file.

Then we also have another command called flush command literally means flushing so but the way that it is the flashing in happened. Whatever stored in the buffer gets written into the time at this point. Usually when it exits that is when it is written out in this case when you issue the flush command at that point it is written out into the main line. So no buffer is being yet, now we have other two commands also to get information from the put something into the file.

These are just send boots so the gates command essentially given the file ID in the variable it reads line by line into that you the put command. Essentially like whatever you find the line gets put into the fire and also there are two other commands for non sequential file. Access meaning you can go directly to a video file. Then start the format and those commands are seek and tell the tell command essentially it is used to obtain the current position of the pointer in this there as a seek command will this given an offset it goes to that particular location to be over there.

(Refer Slide Time: 36:53)



Procedures

- Proc name args body
- Variable scope
  - Upvar
  - Upscript
- Variable number of args - args
- Arrays in procedures

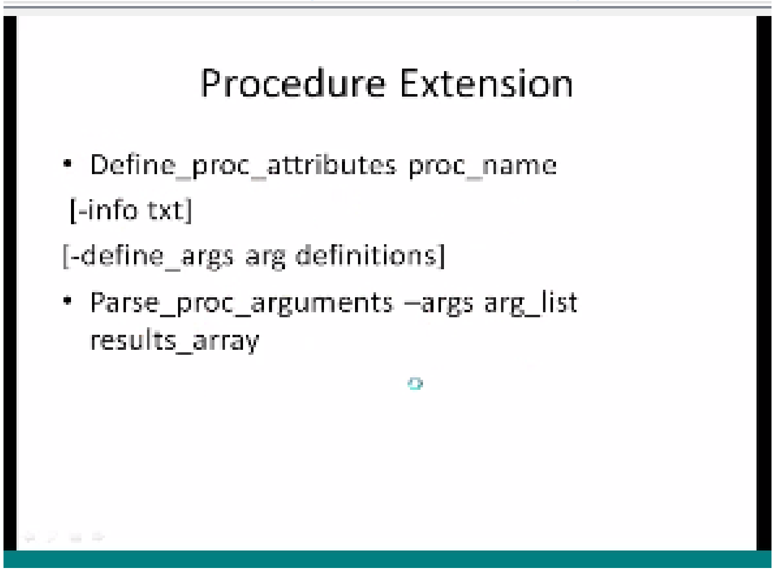
Now we turn to much more interesting stuff this procedure so I actually saw actually define what it is and then, then we can run these procedures instead of for the nature commands. And then the main command that creates a deep procedure is Brock. So and then Brock has three more arguments it is the name of the brock arguments and then the body. So we saw like I mean a lot of these think about procedures in previous lecture. And we also talked about the variable scope in the context of vertical over article and this is essentially the variable for determines.

Like how to access available so if you want to preserve the scope from a global variable into the procedure we use the keyword global. So that that particular variable is visible inside when a month or inside the procedure. Now if you want to pass a procedure that is available back to the calling process and without actually passing it as a result. We can do that using form and we can also do up script to upload the scriptures. These are some of the commands that we solve.

And we can also specify defaults for the arguments this right next to it as a single element position and then pulse will take it. Now coming to the variable number of arguments this is the tennis ARVs form in the procedure. So if one of the one of the arguments is this called an ARV then now tickle trace this as the variable number of arguments. So it will keep reading that $logs until it is fully exhausted. And arrays in procedures you can use arrays inside passages basically again.

The scope needs to be determined and then you can assign a scope and then it is it is quite pervasive. At the time and arrays can also be added in diseases now extension this procedure extension essentially.

(Refer Slide Time: 40:27)



Procedure Extension

- Define_proc_attributes proc_name

[-info txt]

[-define_args arg definitions]

- Parse_proc_arguments –args arg_list results_array

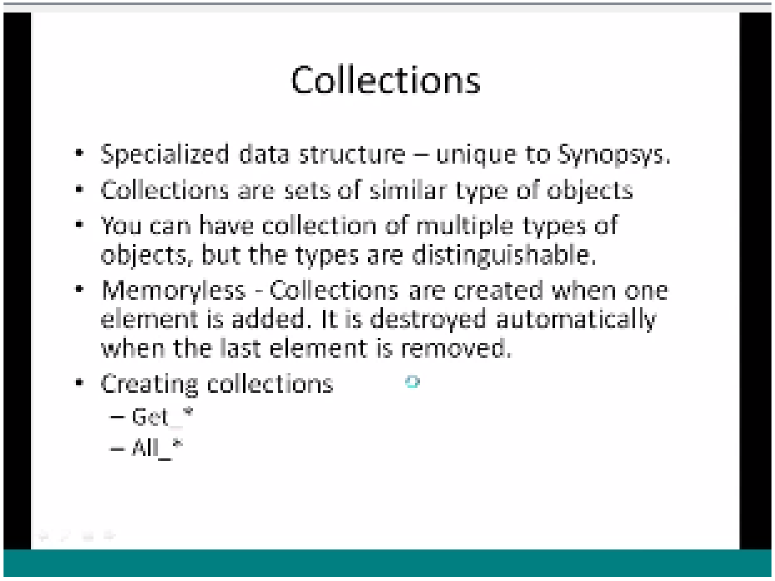
So cinalf sides actually support some built-in features that can actually enhance the whole table scriptwriting it is no longer feels like a script but it is a full-blown program. So two of such commands are defined Rock attribute POC name and then optionally a need for text or the second one is parse proc some arguments. Which has the - arguments key with the argument lists and then there are a result area which can make some salt you so the advantages of using these scripts will depend more script and the par script is that when somebody look with the help on this command.

It automatically produces a L you so the design frog command essentially, takes as I proclaim it can have in some influence information and our textual information with - info there are other things which are not very relevant which is the command group height body hidden permanent and do not ever or abbreviate. So I have not put the labels additional options you now the next one is the parse frog arguments the part product arguments has a arglist as one of the arguments and then result array.

So the results array will contain the parsed arguments essentially. So let us store in the result array and then the arc list is essentially Mr. Parkman's that needs to be passed procedure. The reason why you be used these two commands are mainly for that it makes precision more readable and also it gives the consistent messaging around the brock. The define brock comments but comments also goes with the design for attributes the command.

That actually gives more information about the all the attributes for the arguments you the parse proc arguments command essentially. That if you do not have this the particular script will not respond to the - help you can make it work by using a help procedure name and then - verbose but if you want to get the procedure added you need to define. The part parses the rock targets then once we define how we display.

(Refer Slide Time: 44:37)



Collections

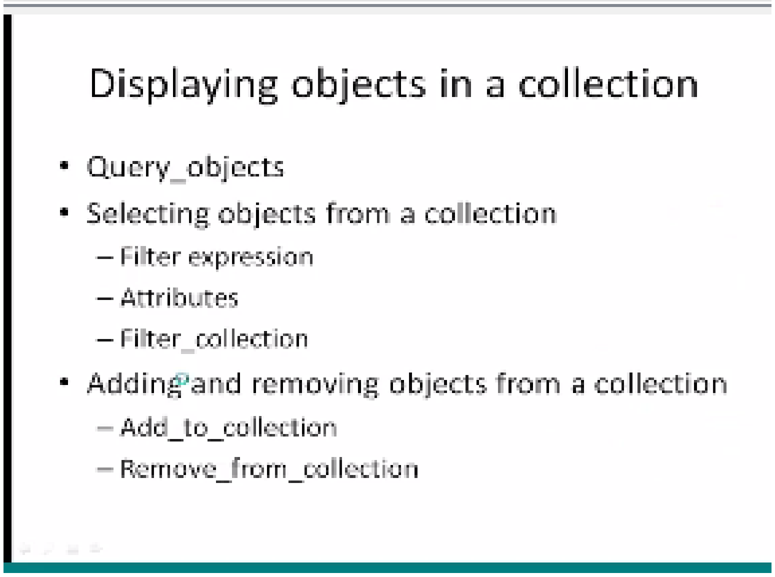
- Specialized data structure – unique to Synopsys.
- Collections are sets of similar type of objects
- You can have collection of multiple types of objects, but the types are distinguishable.
- Memoryless - Collections are created when one element is added. It is destroyed automatically when the last element is removed.
- Creating collections
  - Get_*
  - All_*

The procedure, for that we have several commands we can have info arks in full body. So these are some of the finance that we can use in procedure. So now we come to the collection which is the one other data structure which is unique synopsis the collections are set of similar type of objects. Can have collection of actually multiple type also say with the attribute select nets into a collection all the cells into attention and then you can have a single collection expendable next and sells but they are all grouped separately.

So if you have a name called X in the next and corresponding X in the name says they both are separate first of all inside the array itself and then we can separately destroy one of them with this the other one needs to be more prominent so if you remove any objects so it calls for like self so if you remove X it can be the cell or net and it could removes it from this cell see the X the net X is still retained inside the collection.

So other important thing to note about Corrections is Corrections or memory less so it is created when at least one element gets added and it is destroyed automatically. When the last element is removed. So by itself it does not have any kind of bearing on and for creating collections there are two commands one is get any get star is a tickle command or creating collections. And then all underscore stars is also another command.

(Refer Slide Time: 47:38)



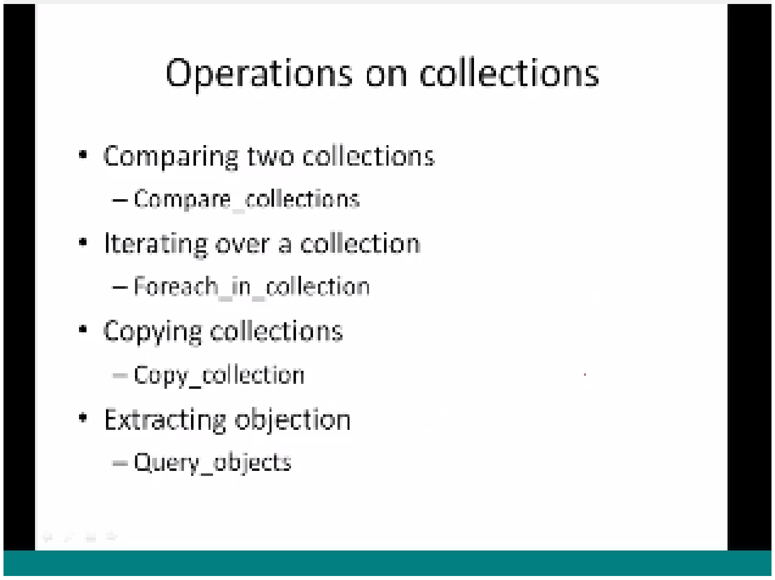
Displaying objects in a collection

- Query_objects
- Selecting objects from a collection
  - Filter expression
  - Attributes
  - Filter_collection
- Adding and removing objects from a collection
  - Add_to_collection
  - Remove_from_collection

So if you want to display objects in detection we use query object system and for selecting objects from my collection again filter expression is one attributes. So to select an object from a collection we can use a filter option or filter collection command. So this one is like the filter and then whole bunch of attributes. Now how do we add and remove objects from a collection we already saw like how to advocate a lemon using the yet or actually is the set commands but for adding and removing objects from a collection.

We have to use these two commands either add to collection or remove from connection so these are preset commands basically which will help add new objects into the collection or remove the existing ones and as I said basically the last element is remove the disarray cease to exist collection system and basically completely this time.

(Refer Slide Time: 49:33)



Operations on collections

- Comparing two collections
  - Compare_collections
- Iterating over a collection
  - Foreach_in_collection
- Copying collections
  - Copy_collection
- Extracting objection
  - Query_objects

So I mentioned earlier that the collections are very versatile and it is probably like the best thing happen here. So to compare two collections we use the compare collections and then if you want to trade over a collection use for each in collection see basically the common theme is the collection system between more and to copy collection basically we this useful copy command or copy underscore collection command.

For each collection also has just two arguments then for to copy detection use the copy collection form and the extracting objects should be it using the query matrix. So I think like I mean this pretty much is the overview of how we can use form tickle in synopsis tool of this is useful will probably like go through some examples in the next lecture thank you.