Dara Lim

CS4250

Homework6

1. The general characteristics of a subprogram are as follows:

- Each subprogram has a single entry point.

- The calling program is suspended during execution of the called subprogram.

- Control always returns to the caller when the called subprogram’s execution terminates.

2. Table showing advantages and disadvantages of using local static and stack-dynamic variables.

|  |  |  |
| --- | --- | --- |
|  | **Stack-dynamic local variables** | **Static local variables** |
| **Advantages** | -Support for recursion  -Storage for locals is shared among some subprograms | -More efficient  -Require no run-time over-head for allocation and deallocation.  -Direct addressing  -Allow subprogram to be history sensitive. |
| **Disadvantages** | -Allocation, Deallocation, initialization time.  -Indirect addressing.  -Subprograms cannot be history sensitive. | - Inability to support recursion.  - Storage cannot be shared with the local variables of other inactive subprograms. |

3. Describe the problem of passing multidimensional array as parameter.

If a multidimensional array is passed to a subprogram and the subprogram is separately, compiled

the compiler needs to know the declared size of that array to build the storage mapping function.

Programmer is required to include the declared sizes of all but the first subscript in the actual parameter. Disallow writing flexible subprograms.

4. Shallow binding is the environment of the call statement that enacts the passed subprogram. Most natural for dynamic-scoped languages.

Deep binding is the environment of the definition of the passed subprogram. Most natural for static- scoped languages.

5. Problem#5 (question a and b), p.414.

a. Pass by value.

The pass-by-value copies the values of the actual parameters in the formal parameters. The changes made in the formal parameters will not change the actual parameters.

Therefore, the values of the variables value and list will remain the same after all the function calls.

The values are as follows:

list[] = {1, 3, 5, 7, 9}, value = 2.

b. Pass by reference function call accepts the address of the actual parameters. Therefore, the changes made in the formal parameter will reflect in the actual parameters since the operations are performed using the addresses. The change in the values of the variables value and list is shown below:

. The first function call swaps the values of the parameters value and list[0] resulting in the following values:

value = 1, list [] = {2, 3, 5, 7, 9}.

.The second function call swaps the values of the parameters list[0] and list[1] resulting in the following values:

value = 1, list [] = {3, 2, 5, 7, 9}.

.The third function call swaps the values of the parameters value and list [value] resulting in the following values:

value = 2, list [] = {3, 1, 5, 7, 9}.

6. Problem#7 (questions a and b), pp.414-415

a. Pass by value: When parameters are passed by value to a subprogram, subprograms makes a copy of the parameters and any changes to the formal parameters do not reflect to the actual parameter. Therefore, the value of list array for pass by value call to function is = {1,3}.

b. Pass by reference: When parameters are passed by reference to a subprogram, an access path of the actual parameter is provided to the formal parameter. Now the formal parameters have access to the actual parameter. Any changes to the formal parameters are reflected to the actual parameter. Therefore, the value of list array after parameters are passed by reference is = {2, 6}.

7. Two arguments against the use of pass-by-name:

-When parameters are passed by name, all the occurrences of formal parameter in the definition of the called subprogram are replaced by the corresponding actual parameter. It is used as macros in C Language. Pass by name method is used in AlGOL to pass the parameters to the called subprogram.

-Pass by name can cause insignificant semantics in the program. If any local variable of the called subprogram has the same name as that of actual parameter, the readability of the subprogram is affected.

- Each instance of the formal parameter is replaced by the name of the actual parameter. If the actual parameter is an expression then the expression is evaluated every time an instance of formal parameter is found in the called subprogram.