RUN-TIME ENVIRONMENTS

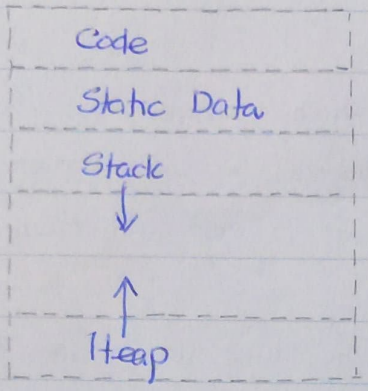
- The design of the run-time support, which manages the allocation and deallocation of memory locations, is influenced by the semantics of procedures.
- Each execution of a procedure is referred to as an "activation" of the procedure.
- The "lifetime" of an activation of a procedure P is the sequence of steps between the first and last steps in the execution of the procedure body, including time spent executing procedures called by P.
- We can use a tree, called an "activation tree", to depict the way control enters and leaves activations.
- We can use a stack, called a "control stack" to keep track of live procedure activations.

Scope of a Declaration

- The scope rules of a language determine which declaration of a name (identifier) applies when the name appears in the text of a program.
- The portion of the program to which a declaration applies is called the "scope" of that declaration.
- An occurrence of a name in a procedure is said to be "local" to the procedure if it is in the scope of a declaration within the procedure; otherwise, the occurrence is said to be "nonlocal".

Storage Organization

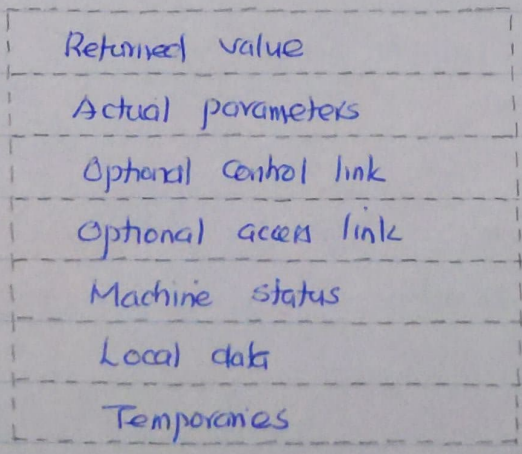
- The run-time storage, obtained as a block of storage from the operating system, might be subdivided to hold:
  1. the generated target code
  2. data objects
  3. control stack to keep track of procedure activations, and
  4. a heap hold other information, such as the ones through dynamic allocation.



- The size of the generated target code is fixed at compile time, so the compiler can place it in a statically determined area.
- The size of some of the data objects (like global variables) may also be known at compile time, and these too can be placed in a statically determined area.
- When a call occurs, execution of an activation is interrupted and information about the status of the machine, such as the value of the program counter and machine registers, is saved on the stack.
- A separate area of run-time memory, called a "heap", holds all other information (like the ones through dynamic allocation).

Activation Records

- Information needed by a single execution of a procedure is managed using a contiguous block of storage called an "activation record", consisting of the collection of fields shown below.



- Temporary values, such as those arising in the execution of expressions, are stored in the field for temporaries.
- The field for local data holds data that is local to an execution of a procedure.
- The field for machine status holds information about the state of the machine just before the procedure is called.
- The optional access link is used to refer to nonlocal data held in other activation records.
- The optional control link points to the activation record of the caller.
- The field for actual parameters is used by the calling procedure to supply parameters to the called procedure.
- The field for the returned value is used by the called procedure to return a value to the calling procedure.

Parameter Passing

- When one procedure calls another, the usual method of communication between them is through nonlocal names and through parameters of the called procedure.
- Differences between parameter passing methods are based primarily on whether an actual parameter represents an r-value, an l-value, or the text of the actual parameter itself, based on which they are called
  - call-by-value
  - call-by-reference,
  - call-by-name

  respectively.

- Example : Parameter passing

```
program main()
    var y : integer;
    procedure p(x : integer);
        begin
            x := x + 1;
            write(x, y);
        end;
    begin
        y := 1;
        p(y);
        write(y);
    end.
```

- Output :

call by value : (2, 1, 1)
call by reference : (2, 2, 2)
call by value-result : (2, 1, 2)
call by name : (2, 2, 2).