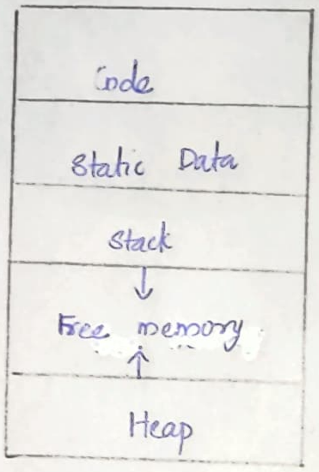
Storage Organization

The executing target program runs in its own logical address space in which each program value has a location.
The management and organization of this logical address space is shared between the compiler, operating system and target machine.
The operating system maps the logical address into physical addresses, which are usually spread through memory.

Typical subdivision of run time memory.



Code area : Used to store the generated executable instructions, memory locations for the code are determined at compile time.

Static Data Area : Is the locations of data that can be determined at compile time

Stack Area : Used to store the data object allocated at runtime

Heap : Used to store other dynamically allocated data objects at runtime

Storage Allocations Strategies

The different storage allocation strategies are :

Static allocation

- In static allocation, names bound to storage as the program is compiled so there is no need for a run-time support package
- Since the bindings donot change at runtime, every time a procedure activated, its runtime name bounded to the same storage locations. Therefore, values of local names retained across activations of a procedure. That is when control returns to a procedure the value of the local are the same as they were when control left the last time
- From the type of a name, the compiler decides amount of storage for the name and decides where the activation records go. At compile time, we can fill in the address at which the target code can find the data it operates on.

Stack Allocation :

- All compiler fors languages that use procedures, functions or methods as unit of user functions define actions manage at least part of their runtime memory as a stack runtime stack.
- Each time a procedure called, space for its local variables is pushed onto a stack, and when the procedure terminates, space popped off from the stack.

Heap Allocation

- Heap allocation parcels out pieces of contiguous storage as needed for activation records or other objects.
- Pieces may be deallocated in any order, so over the time the heap will consist of alternate areas that are free and in use.
- The record for an activation of procedure r is retained when the activation ends. Therefore the record for the new activation q(1,9) cannot follow that for s physically.
- If the retained activation record for r is deallocated, there will be free space in the heap between the activation records for s and q.