Wind

Software library for C++

Subhajit Sahu

Function Names

The function naming strategy is important for easy usability and understandability of the library as an API. It should be tried to remove unnecessary features, and map requisite feature to proper function names. Since functions perform an action, their names should be verbs, and as much as possible fall into the following names:

|  |  |
| --- | --- |
| Name | Use |
| Begin | Prepare a module before first use |
| End | End the use of a module |
| Create | Alternative to constructor, where it might need to be avoided |
| Destroy | Alternative to destructor where it might need to be avoided |
| Optimize | Take time to perform certain optimization |
|  |  |

Heap

Dynamics memory allocation is performed from a heap, and in Windows, each process can have multiple heaps. Each process also has a default heap of its own. It was also observed that memory allocation of large blocks a memory takes a lot of time. However, following are the objectives of dynamic memory allocation schemes:

* Heap allocation functions should support both C and C++ style functions (with C style functions containing the actual code, and C++ functions simply calling them).
* Care must be taken with constructors which otherwise might lead to unexpected results.
* Heap functions must use basic verbs for function names, if available.
* Since memory allocation is a costly operation, functions should constrain the programmer to put less stress on dynamic allocation.
* Heap functions must return both pointer to memory and size actually allocated, as return value.

|  |  |
| --- | --- |
| Function | Use |
| Begin() | Prepare the heap module before first use |
| End() | End the use of the heap module |
| Create() | Create a new heap. Alternative to constructor. |
| Destroy() | Destroy a created heap. Destructor unused. |
| heap() | Create a new heap (constructor). |
| Optimize() | Optimizes a heap so that further operations are faster. |