Documentation for Small String Optimization (SSO)

By Daniel Manchevski

Small String Optimization (SSO) is a technique that has been used in std::string implementation to optimize for the common case where strings are small. The underlying idea behind SSO is to store small strings directly within the string object itself rather than allocating a separate block of memory on the heap.

My MyString class the implementation of SSO is done using union that can contain either a fixed-size array or a pointer (_fixedArray) to a dynamically allocated array (__dynamicArray), in addition to a _isDynamic flag, which serves the purpose of indicating which storage is being used.

In the event that the string is short enough to fit into the fixed-size array, the fixed-size array is used to store the string data, but if it is too long, the dynamically allocated array is used instead. It can enhance performance by reducing the number of memory allocations and deallocations needed for small strings.

To retrieve data from the arrays, we first check if the string is stored in the fixed-size array or the dynamic array by checking the value of _isDynamic. If the string is stored in the fixed-size array, we can simply read the characters from the array. On the other hand, if the string is stored in the dynamic array, we need to retrieve the characters from the dynamically allocated memory.

Writing to the arrays follows a different set of rules, where we always write to the dynamic array if the string is longer than the fixed-size array. Otherwise, if the string is short enough to fit into the fixed-size array, we write the characters to that array instead. Every time we write or append to the current array there is a check if it should move from a fixed-size array to a dynamic one.