

Experiments:

For the C# implementation. After 3 trials of loading the bigbible.txt into my text buffer. Below are my test results.

<Trial> <Static Time> / <Dynamic Time>

1. 5.39min / 5.03min

2. 5.80min / 4.68min

3. 5.82min / 4.55min

Memory usage:

<Trial> <Static Time> / <Dynamic Time>

1. 72mb / 65mb

2. 72mb / 65mb

3. 72mb / 65mb

Result (C#)

- For C# the static length string loading time is about 17.66% faster than the dynamic length string.
- The static length string memory usage is about 10.219% more than the dynamic length string.
- From the IDE ram usage graph, I can see the heap garbage collection occurs 6 times for the dynamic length string, and 12 times for the static length string.

Conclusions (C#)

- From this experiment, we can see the static length string is much more efficient than the dynamic length string. The reason could be the dynamic length string requires more memory allocation to create each line. The default capacity for the dynamic string is 10 and the maximum capacity is 1024. The static string has a default capacity of 550.
- The reason why C# is much slower than the plain C language could be. I didn't use pointer for most of the parameter passing, so the IDE will need to create too many new parameters and let the garage collector clean lots of unnecessary memory locations during the load time.