Time Measurements

Amina Hamza

Software Technology

Department of Computer Science, Linnaeus University

Abstract—This is my report for Exercises 6 and 7 in Assignment 4 in the course 1DV507 – Programming and Data Structures. The task handled are below.

Experimental Setup--Experiments were done using Macbook Pro with an Intel Core i5 processor 1.60GHz with 8GB memory.

I started the experiment by creating an empty array with generic type string, then I initialized the array with different length depending on the test. I did the time measurement using the method System.currentTimeMills two times, first time before the experiment begun and second time after the experiment then I calculated the deference between the two times and finally I came with the real actual time it took the experiment. I did the experiment for each case five times.

```
long startTime = System.currentTimeMillis();
            // implmentaion
            long endTime = System.currentTimeMillis();
            int actualTime = (int) (endTime - startTime);
```

Table of mesurments

Case 1

	Time(in milliseconds)	length
String One Character	1000	86183
String 80 Characters	1001	1082
String Builder One Character	1105	150994943
String Builder 80 Character	1167	3765826

Case 2

	Time(in milliseconds)	length
String One Character	1000	102955
String 80 Characters	1000	1306
String Builder One Character	1048	182402741
String Builder 80 Character	1167	3774874

Case 3

	Time(in milliseconds)	length
String One Character	1000	104223
String 80 Characters	1001	1315
String Builder One Character	1050	188021716
String Builder 80 Character	1078	4839427

Case 4

	Time (in milliseconds)	length
String One Character	1000	104745
String 80 Characters	1000	1317
String Builder One Character	1047	191717150
String Builder 80 Character	1080	5093975

Case 5

	Time (in milliseconds)	length
String One Character	1000	104885
String 80 Characters	1000	1317
String Builder One Character	1048	189815310
String Builder 80 Character	1085	5006928

why StringBuilder is much faster than string concatenation using the + operator?

I think when we use the StringBuilder, we are not allocating memory every time we append something to the array, it is only done when the array is resized, but in the String we are allocating memory every time we add something to the string that is why it taking more time.