**Özgür Öney**

**21101821**

**CS342 – 2**

**Assoc. Prof. Dr. Özcan Öztürk**

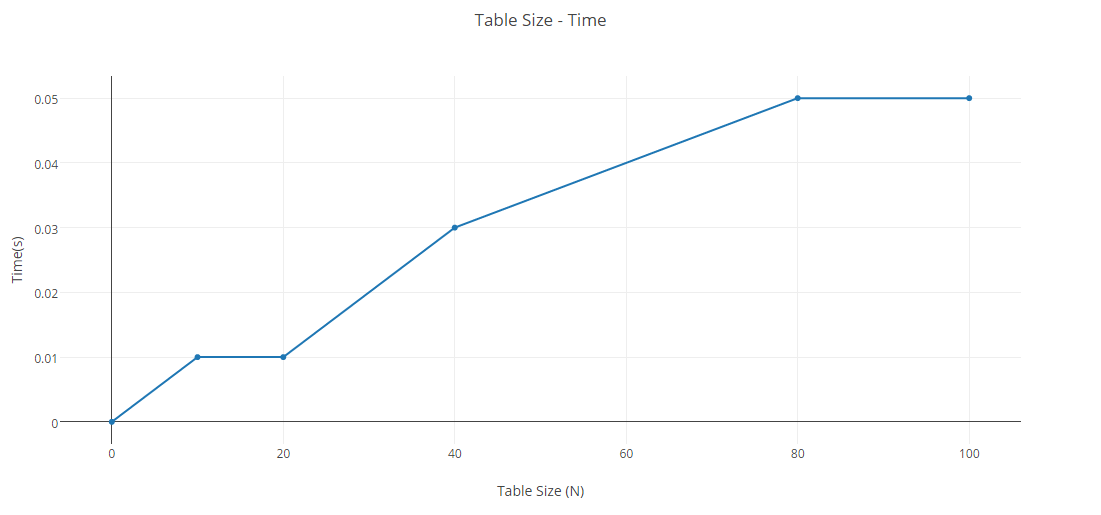
**Project 3 Report**

At part B, I have implemented a tread-safe hash table structure. To do such organization, I have created a struct named SynchControl, which holds a mutex in it and created for every single bucket on the hash table. Whenever the table is initialized with number taken from command line, SynchControl module is also created to keep mutex. Manipulations done on a particular bucket is protected with it. For example, whenever user requests to do some insertion on a single bucket, deletion and retrieval on it is prohibited.

Graphs given below demonstrate different scenarios while I tested hash table implementation. Necessary explanations are also provided for each graph. Numbers determined by considering given limits for N (N>10 & N<100)

***Table A:*** Constant number of deletions and insertions while table size is increased.

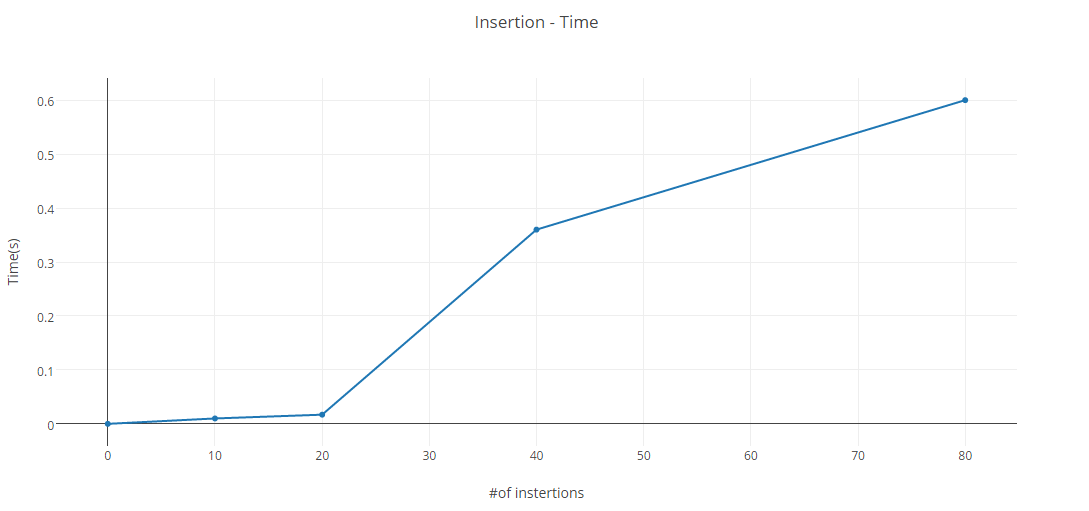
|  |  |
| --- | --- |
| **Table Size** | **Time(s)** |
| 10 | 0.01 |
| 20 | 0.01 |
| 40 | 0.03 |
| 80 | 0.05 |
| 100 | 0.05 |



As can be understood from the figure, as long as we increase the number of buckets in hash table, time needed to do fixed number of deletions & insertions is increased. However, although the computer used for testing is same, graph does not behave linearly; which could be a result of background processes running on the computer.

***Table B:*** Needed time for insertion when number of buckets (N) is the same.

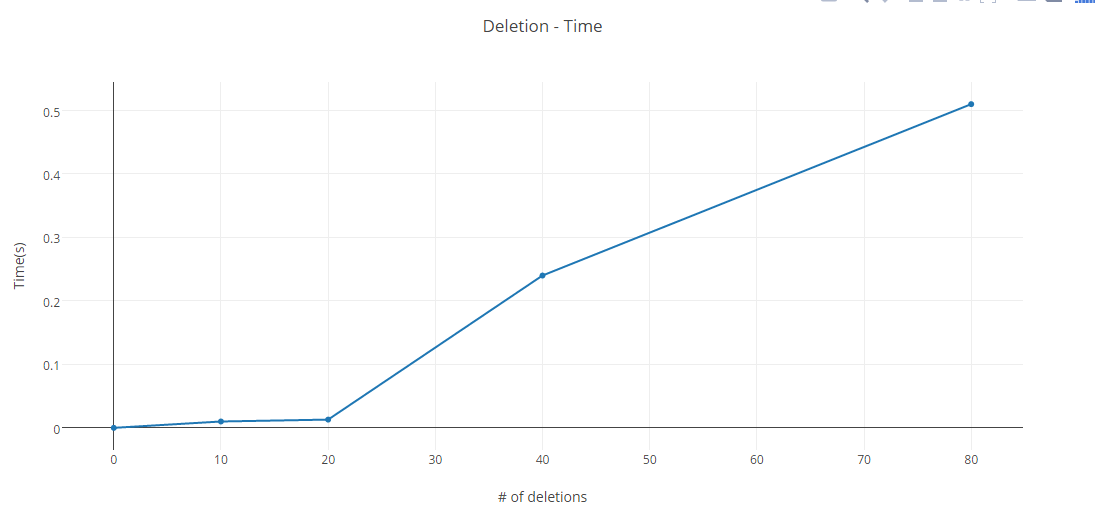
|  |  |
| --- | --- |
| **# of insertion** | **Time** |
| 10 | 0.01 |
| 20 | 0.017 |
| 40 | 0.36 |
| 80 | 0.6 |



As can be understood from the figure, time increases in parallel manner when number of insertion increases. To exemplify this, imagine a scenario where we want to insert both 2 and 12 at mod10, which indicates that we are aiming to bucket #2. Hereby, we have to wait for the insertion of 2 before, just after that 12.

***Table C:*** Needed time for deletion when number of buckets (N) is the same.

|  |  |
| --- | --- |
| **# of deletion** | **Time** |
| 10 | 0.01 |
| 20 | 0.013 |
| 40 | 0.24 |
| 80 | 0.51 |



As can be understood from the figure, time increases in parallel manner when number of deletion increases. To exemplify this, imagine a scenario where we want to delete both 3 and 13 at mod10, which indicates that we aim to use bucket #3. Hereby, we have to wait for the deletion of 3 before, just after that 13 can be deleted. (No need to say that they should be in the list of bucket #3).