**Assignment 2 - Marking Sheet**

<table>
<thead>
<tr>
<th>Item</th>
<th>Comments</th>
<th>Mark</th>
</tr>
</thead>
</table>
| **Correctness**       | [+1] for each benchmark for each number of threads (2-7) that runs successfully (6 benchmarks x up to 6 points each)  
                        [+2] for total memory use < 20x kheap (indicator that free() is working)  
                        [+2] for handling large (> 4K) allocations                                                                 |  / 40 |
| **Performance**       | **Sequential speed pts (maximum 8)** = \[
\sum_{all\; bench} \frac{T_{libc}}{T_{a2alloc}} \times 8
\]  

**Scalability pts** = scalability of threadtest, linux-scalability, Larson and phong  
(maximum 12)  

**False sharing pts** = scalability of cachescratch and cache-thrash  
(maximum 12)  

**Fragmentation pts (maximum 8)** = \[
\sum_{all\; bench} \frac{\min_{mem}}{\;_{mem\; used}}
\]  

scaled such that  

\[=> 0.1 \rightarrow 1, \; \geq 0.3 \rightarrow 2, \; \geq 0.5 \rightarrow 3, \; \geq 0.6 \rightarrow 4, \; \geq 0.7 \rightarrow 5, \; \geq 0.8 \rightarrow 6, \; \geq 0.9 \rightarrow 7, \; \geq 1.0 \rightarrow 8\]  |  / 40 |
| **Style**             | Example deductions:  
                        [-2] for many short or otherwise cryptic variable names  
                        [-2] for poor or misleading comments  
                        [-2] poorly organized code, etc.                                                                 |  / 10 |
| **Writeup**           | Example deductions:  
                        [-1] for > 2 grammatical errors  
                        [-1] for > 2 spelling errors  
                        [-1] for not discussing alternatives  
                        [-1] for no references  
                        [-3] for no design  
                        [-3] for no analysis  
                        [-2] for analysis with no or very few numbers  
                        [-1] for no discussion of fragmentation or memory use in performance analysis |  / 10 |
| **TOTAL**             | Other deductions:  
                        [-2] for using more than 1 KB of globals                                                                 |  / 100 |

**NOTE:** allocator performance cannot be fairly assessed if the allocator is not correct.  
(e.g., failing to synchronize correctly could yield a sizeable speed or scalability advantage in the cases where it doesn’t cause the program to crash) Nonetheless, we will attempt to assess partial performance scores where possible.