Bilkent University Computer Science

CS342

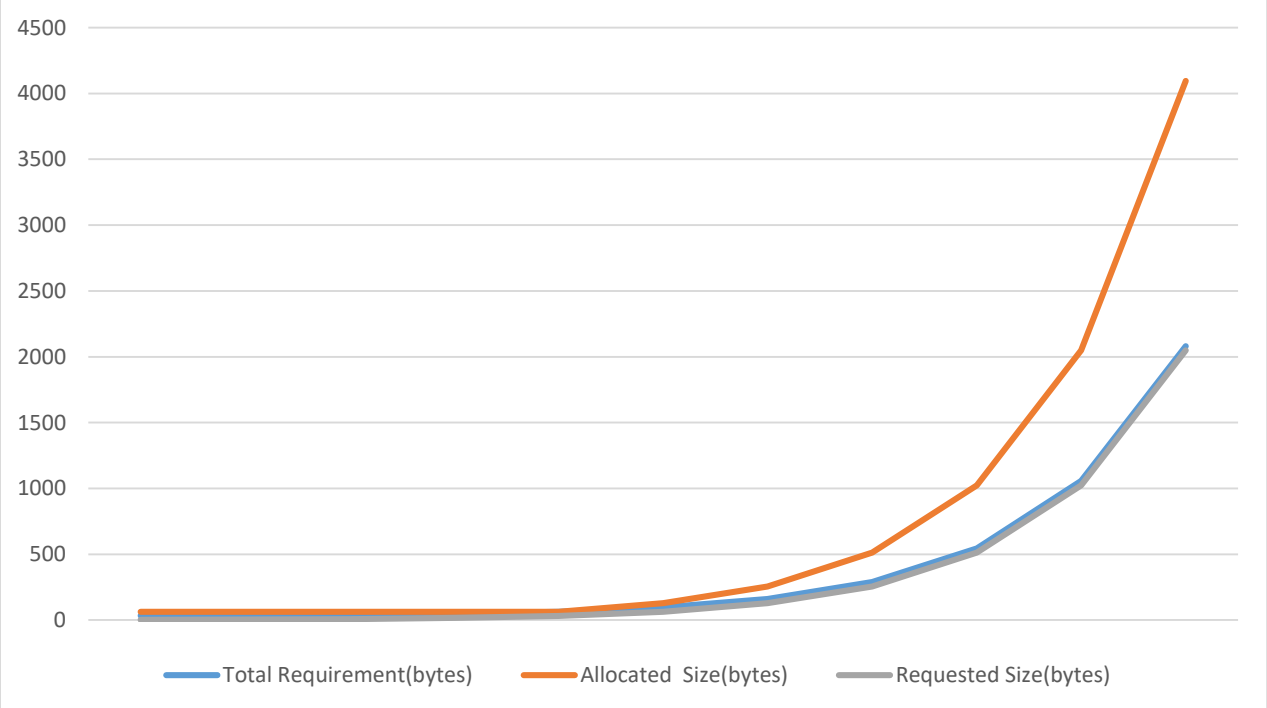
Project 3 Report

Turan Mert Duran 21601418 Section 1

Radman Lotfiazar 21600450 Section 2

Memory Management

Requested Size(bytes)	Allocated Size(bytes)	Total Requirement(bytes)
2	64	34
4	64	36
8	64	40
16	64	48
32	64	64
64	128	96
128	256	160
256	512	288
512	1024	544
1024	2048	1056
2048	4096	2080



In this experiment we have chosen different bytes of size allocation for memory and see the how much memory is allocated for each requested bytes of size. While doing this, we used buddy memory management algorithm for them to store. Buddy algorithm works like that, for each requested size of allocation, it finds firstly what is 2 power of the most close and bigger or equal value of requested value. It allocates size always power of 2. Therefore, let assume that if requested size is 80 bytes, and there is nothing to store with it like (address pointer, size int, etc.), closest number that is bigger than requested (80) and power of two, 128 bytes is allocated by the algorithm. In our experiment, with each allocation we stored linked list nodes with memories to keep track of them. We stored end and start points of memory, process id and next memories' address. So, our struct needed 32 bytes even if requested byte size is 1. According to this, we took the requested size and added our node size (32) bytes and find closest number that is bigger or equal to requested and power of 2. Therefore, we allocated number of size that is bigger than (32 + requested size) and closest power of 2. In our experiments we saw lots of internal fragmentations. As we chose requested byte size close to power of two, it allocates much more than we need that causes fragmentation. For example, when we requested 2 bytes of size for memory, it needed 32+2 = 34 bytes but allocated 64 bytes. So 64-34 = 30 bytes goes for fragmentation. However, there would be such cases where there is not any fragmentation like requested 32 bytes. When we requested 32 bytes, we need 32+32(node size) = 64 bytes which satisfies power of two and bigger or equal to needed size. So, there is no fragmentation occurred in this trial.

It can be acquired that buddy algorithm is useful in terms of fragmentation when total need of bytes is close to power of two. However, if it pass the power of two with a little bit offset it causes occurrence of lots of fragmentation.