Pınar Yücel
21802188

CS342 Operating Systems - Spring 2021
Project #3

In this implementation, every block in the shared segment has some overhead on their starting point. This overhead includes information about the segment which it belongs to. If a request of 128 bytes is made, assuming that the overhead is less than or equal to 128 bytes, 256 bytes will be allocated so that there is enough space for the overhead. There is also an overhead on the starting point of the shared segment which includes information about the shared memory in general. If the parameter of the sbmem_init function is 32768, a shared segment of 32768 plus general overhead size is opened.

Experiment 1 Description: In this experiment, shared segment size was 32768 excluding the general information overhead before it. Initially, allocation requests were made with request sizes 4000, 3000, 2000 and 1000 until no more available memory was left. Internal fragmentation in the following table refers to the amount of total internal fragmentation after the allocation requests are finished. After that, half of the odd numbered allocations were freed. First allocation got freed, second did not, third got freed, fourth did not and so on. After this deallocation process, the amount of external fragmentation was calculated.

Experiment 1 Conclusion: It can be seen from Table 1 that the external fragmentation amount is the same for 4000 and 3000 bytes of requests. Even though 4000 bytes of request is much more than 3000 bytes of request, they both get the same segment size, 4096. Since the segment sizes are the same in the memory, it makes sense that the external fragmentation is the same for those. However, the internal fragmentation amount is much larger for 3000 bytes of request. The reason for this is that there is more space between 4096 and 3000 (plus overhead). External fragmentation seems to be related with the request size. When request size decreases, there are more pieces. When there are more pieces, there are more overheads, therefore less external fragmentation.

Frag. Type / Request Size	4000	3000	2000	1000
Total Internal Frag.	672	8672	576	384
Total External Frag.	16336	16336	16288	16192

Table 1

Experiment 2 Description: In this experiment, only the values of the sizes of the requests were changed. Same steps were followed as in the first experiment. In Table 2, all of the request sizes correspond to the same nearest power of two, 1024. In Table 3, again, all of the request sizes correspond to the same nearest power of two, 256.

Experiment 2 Conclusion: With the following data, it can be more clearly seen that external fragmentation amount increases with increasing request sizes (external fragmentation amount is same for request sizes that correspond to the same nearest power of two). This makes sense because when the sizes of the segments are larger, there are less blocks. When there are less blocks, there are less overheads. When there are less overheads, more space is left empty. Even though the closeness of the request sizes of Table 2 to the nearest power of two is the same as the closeness of the request sizes of Table 3 to the nearest power of two, amounts of internal fragmentation in Table 3 are 4 times the amount of internal fragmentations in Table 2. Even though the internal fragmentation is the same for a single block, overall, there are more blocks when the request size is smaller, therefore the amount of total internal fragmentation is more.

It can be seen from all of the tables that even though there is enough space for 8192 minus overhead bytes of memory, such a request can not be satisfied because there is no such memory that is contagious because of external “fragmentation”. If a compaction mechanism were used in the implementation of the library, then it could have been possible to satisfy that request.

Frag. Type / Request Size	994	964	934
Total Internal Frag.	576	1536	2496
Total External Frag.	16192	16192	16192

Table 2

Frag. Type / Request Size	226	196	166
Total Internal Frag.	2304	6144	9984
Total External Frag.	15616	15616	15616

Table 3