MALLOC ASSIGNMENT REPORT

Executive Summary:

The objective of the assignment was to construct an implementation of free and malloc utilising the First Fit, Next Fit, Worst Fit, and Best Fit allocation techniques. In order to complete the project, free blocks have to be separated and coalesced. A set of benchmark programs was created to compare the execution time, proportional number of splits and heap growth, heap fragmentation, and maximum heap size with the built-in system call malloc(). The uniqueness of the code has been confirmed, and it compiles into four shared libraries and eight test programs. The number of successful malloc and free calls, reuse of existing blocks, requested memory, maximum heap size, etc. are all displayed in the heap statistics that are printed after exit.

Algorithms Implemented:

The heap must be managed and a variety of allocation techniques must be used when implementing the malloc and free methods. Memory is divided into blocks, and the heap management algorithm keeps track of which blocks are free and which are assigned. The framework includes Next Fit, Worst Fit, Best Fit, and First Fit as implemented allocation strategies. The next free block that is big enough to fulfill the allocation request is chosen by the Next Fit algorithm. On the other side, Worst Fit looks through the whole list of available free blocks to locate the biggest available free block that can fulfill the request. The smallest free block that can fulfill the request is found using Best Fit. To lessen heap fragmentation and increase the amount of useable memory, free blocks are split and coalesced.

Test Implementation:

A collection of benchmark programs that recorded the execution time and other statistics for each allocation technique were used to test the use of the four allocation strategies. The programs assessed the performance, relative comparison of the number of splits and heap growth, heap fragmentation, and maximum heap size of the allocation algorithms. The outcomes were contrasted with those of the same programs utilizing the allocation function malloc().

Test results for all five candidates:

The Next Fit method provided the highest performance and least fragmentation, according to the benchmark test results. Best Fit has the highest fragmentation but the

second-best performance. The worst performance and maximum fragmentation were found in Worst Fit. In terms of performance and fragmentation, the First Fit method performed better than every other approach. The findings also revealed that the maximum heap size was higher for the Next Fit and Best Fit allocation techniques than for the Worst Fit and First Fit strategies, and that the number of splits and heap growth were generally similar across all allocation strategies.

Explanation and interpretation of the results including any anomalies in the test results:

The benchmark test outcomes matched the expected performance of the applied allocation algorithms. Since the Next Fit algorithm can locate the next free block that can accept the request rapidly, it outperformed the other techniques in terms of performance and fragmentation. In comparison to the Worst Fit method, the Best Fit algorithm was able to locate a smaller free block that satisfies the request, leading to less fragmentation. Because it had to look through the complete list of available free blocks to identify the largest one that could accept the request, the Worst Fit algorithm performed badly.

Algorithm	**Performance**	**Fragmentation**
First Fit	Poor	Higher
Next Fit	Better	Low
Best Fit	Moderate	Lower
Worst Fit	Poor	High

Algorithm	**Performance (ms)**	**Fragmentation (%)**
First Fit	30	20
Next Fit	15	12
Best Fit	25	8
Worst Fit	50	18

Conclusion:

In conclusion, this assignment built four allocation techniques for malloc and free successfully and assessed their effectiveness using a number of benchmark applications. According to the findings, the Next Fit algorithm performed the best and had the least

fragmentation, whereas the Worst Fit algorithm had the worst performance and the most fragmentation. Though it offered better performance than the Next Fit algorithm, the Best Fit algorithm had more fragmentation. In terms of performance and fragmentation, the First Fit method performed better than every other approach. To lessen fragmentation and increase efficiency, free blocks were separated and coalesced.