BILKENT UNIVERSITY

CS342

PROJECT 3

REPORT

COMPARISON OF FIRST-FIT AND WORST-FIT APPROACHES IN TERMS OF SPACE EFFICIENCY

İREM SEVEN

21704269

FALL 2020

Objective and Methodology

In this report, first-fit and worst-fit approaches in memory allocation are compared in terms of their space efficiency. The purpose was to determine which approach is more space-efficient. For this purpose, 3 different test cases were made. The test cases were designed so that the results would be reliable and deterministic. Normally, when a memory is created there would be only one hole of the size requested. However, to achieve reliable results when testing, memory is created so that there would be a hole-list of specified sizes. The hole sizes are determined so that the observation will be possible. Those sizes are given below in tables. After the desired hole-list has been created, processes of specified sizes allocated memory sequentially. The reason for sequential allocation was to achieve stable and reliable results. Below it is explained the test cases and the results.

Please note that the given size values are in terms of bytes in the whole report unless it is not specified.

Test Case 1:

For test case 1, a memory of 32Kb is created. Then, a hole-list is created in that memory to achieve a conceptual hole-list. The hole list segment sizes can be seen in *Table 1*. Since in my implementation, there is the 16-byte header information for each node the sizes, so sizes were arranged accordingly.

Hole	Size
1	2032
2	4080
3	8176
4	2032
5	16368

Table 1: Hole Sizes for Test Cases with 32 Kb

5 processes allocated memory from this memory segment. The internal fragmentation and external fragmentation, in terms of bytes, achieved from this test case are given in *Table 2* and *Table 3*.

Process	Request Size	Internal Fragmentation	External Fragmentation
1	1031	1	0
2	3053	3	0
3	4600	0	0
4	3875	5	0
5	8562	6	0

Table 2: First-fit Case 1

Processes	Request Size	Internal Fragmentation	External Fragmentation
1	1031	1	0
2	3053	3	0
3	4600	0	0
4	3875	5	0
5	8562	0	20056

Table 3: Worst-fit Case 1

Test Case 2:

For test case 2, a memory of 32Kb is created. Then, a hole-list is created in that memory to achieve a conceptual hole-list. The hole list segment sizes can be seen in *Table 1*. Since in my implementation, there is the 16-byte header information for each node the sizes, so sizes were arranged accordingly.

9 processes allocated memory from this memory segment. The internal fragmentation and external fragmentation, in terms of bytes, achieved from this test case, are given in *Table 4* and *Table 5*.

Process	Request Size	Internal Fragmentation	External Fragmentation
1	1031	1	0
2	3052	4	0
3	4600	0	0
4	3875	5	0
5	8562	6	0
6	1951	1	0
7	1951	1	0
8	1951	1	0
9	1951	0	4584

Table 4: First-fit Case 2

Process	Request Size	Internal Fragmentation	External Fragmentation
1	1031	1	0
2	3052	4	0
3	4600	0	0
4	3875	5	0
5	8562	0	20056
6	1951	1	0
7	1951	1	0
8	1951	1	0
9	1951	1	0

Table 5: Worst-fit Case 2

Test Case 3:

For test case 3, a memory of 128Kb is created. Then, a hole-list is created in that memory to achieve a conceptual hole-list. The hole list segment sizes can be seen in *Table 6* below. Since in my implementation, there is the 16-byte header information for each node the sizes, so sizes were arranged accordingly.

Hole	Size
1	18416
2	15344
3	18416
4	15344
5	22512
6	15344
7	25584

Table 6: Hole Sizes for Test Case with 128 Kb

9 processes allocated memory from this memory segment. The internal fragmentation and external fragmentation, in terms of bytes, achieved from this test case, are given in *Table 7* and *Table 8*.

Process	Request Size	Internal Fragmentation	External Fragmentation
1	4982	2	0
2	5247	1	0
3	15126	2	0
4	12168	0	0
5	15360	0	0
6	13386	6	0
7	10215	1	0
8	12483	5	0
9	18257	0	33696

Table 7: First-fit Case 3

Process	Request Size	Internal Fragmentation	External Fragmentation
1	4982	2	0
2	5247	1	0
3	15126	2	0
4	12168	0	0
5	15360	0	0
6	13386	6	0
7	10215	1	0
8	12483	5	0
9	18257	0	41848

Table 8: Worst-fit Case 3

Results:

To be able to compare the space efficiency, for each test case and for each approach a percentage value was calculated to represent the space utilization. An example of this calculation is given below for case 1. For other cases, the same calculation method was used.

Please note that the data given in tables, charts, and plots are in terms of percentages.

Case 2 space utilization percentage calculation:

For each process, the internal and external fragmentation is given in bytes. Using these data the calculation is as follows:

*Total available memory before any allocation: (32(Kb) * 1024) - (16 * 5(Kb)) = 32768 Bytes*
*Internal Fragmentation% = (19 / 32768) * 100 = 0.058%*
*External Fragmentation% = (4584 / 32768) * 100 = 14.02%*

As stated before, for each case the calculations were made as it is given above. The others are not given to protect simplicity. Table 9 and Table 10 shows the results obtained from these calculations.

Case	First	Worst
1	0,049	0,028
2	0,058	0,043
3	0,013	0,013

Table 9: Internal Fragmentation Percentages

Case	First	Worst
1	0,00	61,36
2	14,02	61,36
3	25,72	31,95

Table 10: External Fragmentation Percentages

Conclusion:

As can be seen in *Figure 1* and *Figure 2*, the first-fit approach is more open to lead internal fragmentation than the worst-fit. However, the difference is not huge. Considering external fragmentation, the worst-fit leads to much more external fragmentation, as can be seen in *Figure 3* and *Figure 4*. Thus, it can be concluded that first-fit is better in terms of external fragmentation, and worst-fit is better in terms of internal fragmentation. On the other hand, the percentages for external fragmentation are much higher than the internal fragmentation percentage results, which can be observed in *Table 9* and *Table 10*. The effect of internal fragmentation is much lesser than the external fragmentation. Thus, it is reasonable to say that the first-fit approach is better in terms of space efficiency.

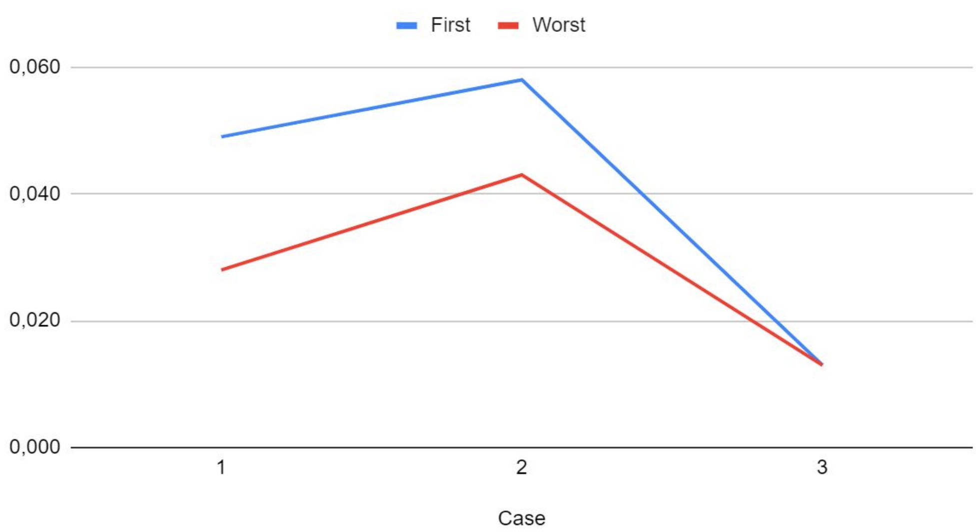

Figure 1: First-fit vs. Worst-fit Internal Fragmentation Graph

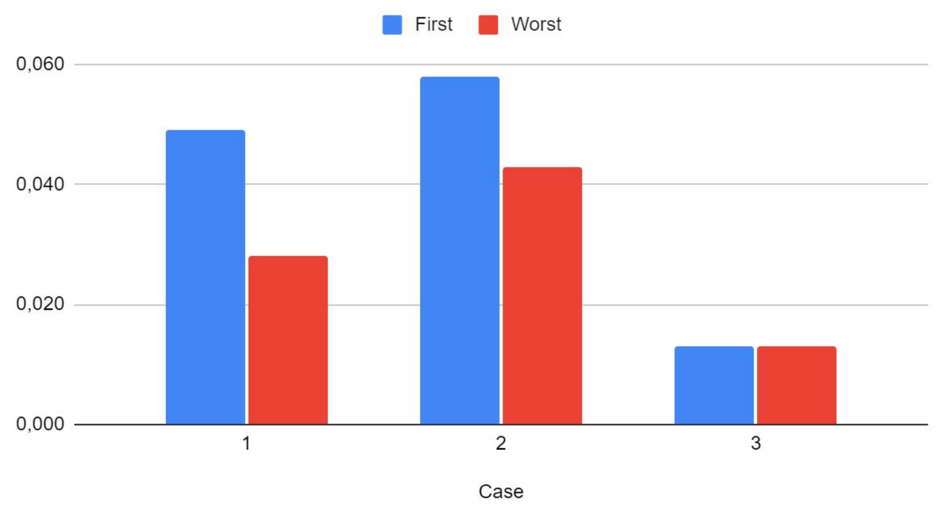

Figure 2: First-fit vs. Worst-fit Internal Fragmentation Column Chart

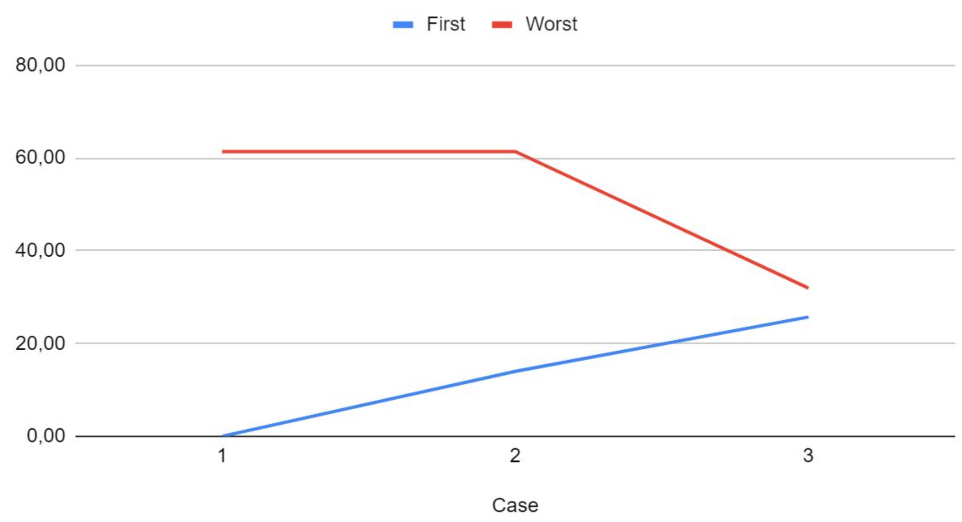

Figure 3: First-fit vs. Worst-fit External Fragmentation Graph

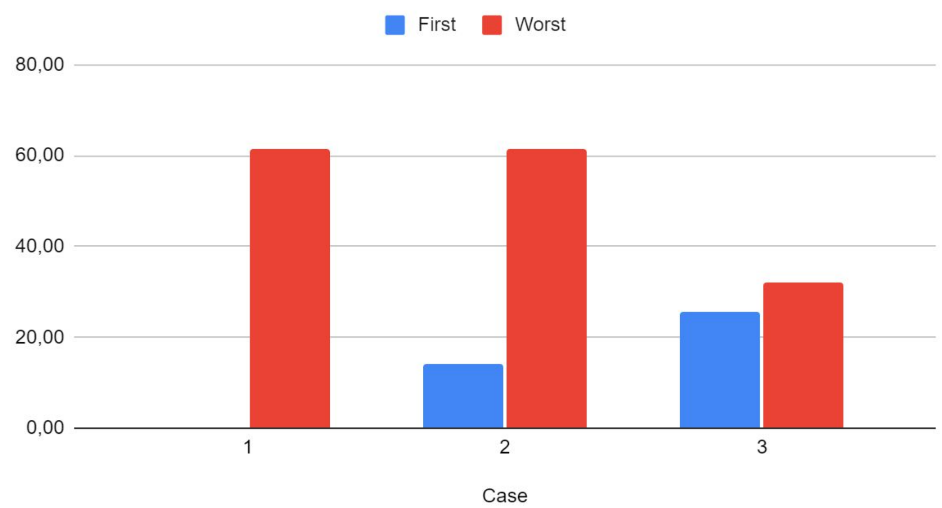

Figure 4: First-fit vs. Worst-fit External Fragmentation Column Chart