Memory Allocation Investigation: FirstFit and BestFit

COSC1114 Operating Systems Principles
Assignment 3

Karina Kresnadi
s3895755

Introduction

To investigate and analyse the performance of memory allocation strategies, we are creating solution designs with two linked lists to manage memory.
The FirstFit method iterates through the free list and allocates the first chunk that is large enough to hold the requested size.
The BestFit method searches for the best-fit chunk in the free memory pool. If a best fit is found, it allocates the memory from that chunk. If no suitable chunk is available, it allocates a new chunk from the system.
The Address, Total Chunk Size, Total Used Chunk size, Total Allocated Size, Total Size, Total Free Size, Allocation requests, Deallocation requests and Total Runtime are recorded in the output of the program.

Methodology

Experimental Setup:

Input file: datafile20, datafile100, datafile250, datafile500, datafile1000 with 5 variations for each datafile size.
Fixed partition sizes of 32, 64, 128, 256, and 512 bytes.

Datafile size 20:

FirstFit (20)	1	2	3	4	5	Avg
Total Allocated Size	3712	1024	1568	5248	3584	3027.2
Total Used Size	2317	809	1128	4105	2397	2151.2
Total Free Size	128	1408	1280	64	640	704
Allocation Requests	9	6	7	13	12	9.4
Deallocation Requests	6	9	8	4	5	6.4
Total Runtime (µs)	1848	1703	2409	2106	2386	2090.4
BestFit (20)	1	2	3	4	5	Avg
Total Allocated Size	3712	1024	1568	5248	3584	3027.2
Total Used Size	2317	809	1128	4105	2397	2151.2
Total Free Size	128	1408	1280	64	640	704
Allocation Requests	9	6	7	13	12	9.4
Deallocation Requests	6	9	8	4	5	6.4
Total Runtime (µs)	1826	1659	1812	2149	2254	1940

Datafile size 100:

FirstFit (100)	1	2	3	4	5	Avg
Total Allocated Size	13504	13632	14592	12384	16672	14156.8
Total Used Size	9901	10215	10574	8942	12465	10419.4
Total Free Size	2816	2368	3456	512	896	2009.6
Allocation Requests	44	44	46	37	50	44.2
Deallocation Requests	31	31	31	32	28	30.6
Total Runtime (µs)	1751	15450	1700	9815	2355	6214.2
BestFit (100)	1	2	3	4	5	Avg
Total Allocated Size	13248	13440	14592	12384	16288	13990.4
Total Used Size	9901	10215	10574	8942	12465	10419.4
Total Free Size	2560	2560	3456	512	1280	2073.6
Allocation Requests	43	44	46	37	50	44
Deallocation Requests	31	31	31	32	28	30.6
Total Runtime (µs)	2587	1635	1520	1615	2095	1890.4

Datafile size 250:

FirstFit (250)	1	2	3	4	5	Avg
Total Allocated Size	24288	30592	38528	18880	27776	28012.8
Total Used Size	17493	22495	27891	13475	20068	20284.4
Total Free Size	7744	2848	800	12096	6016	5900.8
Allocation Requests	95	91	107	90	97	96
Deallocation Requests	86	83	73	97	84	84.6
Total Runtime (µs)	2494	2068	35554	3287	10730	10826.6
BestFit (250)	1	2	3	4	5	Avg
Total Allocated Size	24288	30240	38496	18624	27776	27884.8
Total Used Size	17493	22495	27891	13475	20068	20284.4
Total Free Size	7744	2720	832	12352	5504	5830.4
Allocation Requests	95	91	107	90	96	95.8
Deallocation Requests	86	83	73	97	84	84.6
Total Runtime (µs)	3099	3000	2226	1540	3280	2629

Datafile size 500:

FirstFit (500)	1	2	3	4	5	Avg
Total Allocated Size	67680	57568	60768	67488	62688	63238.4
Total Used Size	48307	42490	45777	49032	46801	46481.4
Total Free Size	6304	4288	1920	2560	2080	3430.4
Allocation Requests	203	184	178	200	184	189.8
Deallocation Requests	157	163	164	155	161	160
Total Runtime (µs)	11056	14547	10678	11771	11213	11853
BestFit (500)	1	2	3	4	5	Avg
Total Allocated Size	67680	56224	60512	67232	62176	62764.8
Total Used Size	48307	42490	45777	49032	46801	46481.4
Total Free Size	5536	5120	2176	2560	2592	3596.8
Allocation Requests	201	184	178	200	184	189.4
Deallocation Requests	157	163	164	155	161	160
Total Runtime (µs)	3919	2757	3351	2958	3192	3235.4

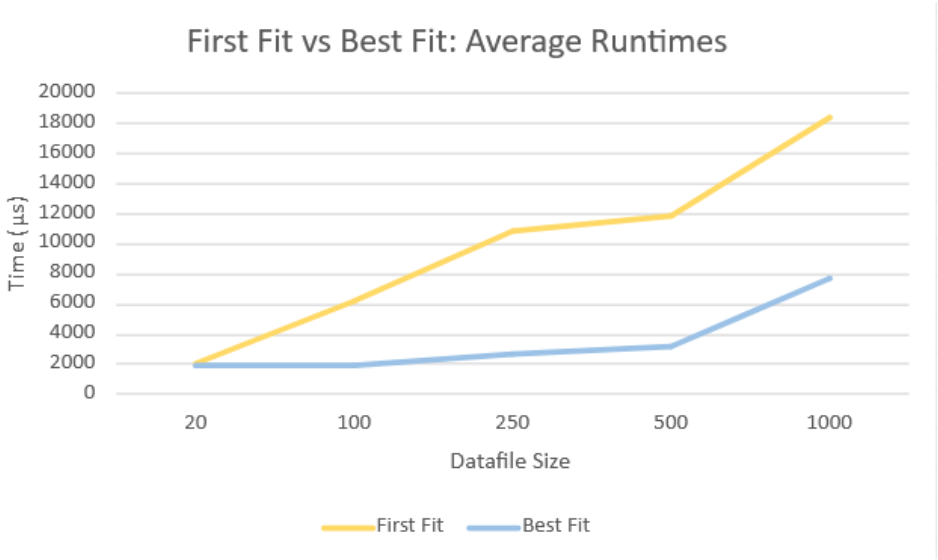
Datafile size 1000:

FirstFit (1000)	1	2	3	4	5	Avg
Total Allocated Size	111328	103264	137504	118112	111424	116326.4
Total Used Size	82259	77065	102054	89115	84106	86919.8
Total Free Size	6464	8064	5440	7104	13696	8153.6
Allocation Requests	345	340	392	358	357	358.4
Deallocation Requests	338	340	311	331	341	332.2
Total Runtime (µs)	14036	15128	34183	15586	13468	18480.2
BestFit (1000)	1	2	3	4	5	Avg
Total Allocated Size	110624	102624	136544	117152	111424	115673.6
Total Used Size	82259	77065	102054	89115	84106	86919.8
Total Free Size	6400	7712	5888	7936	13440	8275.2
Allocation Requests	345	339	391	358	357	358
Deallocation Requests	338	340	311	331	341	332.2
Total Runtime (µs)	4567	9677	4655	10392	9190	7696.2

Average Runtimes (µs):

Avg Runtime	20	100	250	500	1000
First Fit	2090.4	6214.2	10826.6	11853	18480.2
Best Fit	1940	1890.4	2629	3235.4	7696.2

Figure 1: Graph of Average Runtimes



Results and Analysis

From the observed experimentation, the Best-Fit algorithm has faster runtime and similar or less total allocated size compared to the First-Fit algorithm. As the scale of data files increases, the differences in the performances between the first-fit and best-fit methods is noticeably significant.

The Best-Fit method consistently outperforms the First-Fit strategy, which can be attributed to reduced fragmentation. The Best-Fit strategy aims to allocate the smallest suitable free chunk, which leads to fewer, larger, and less fragmented free spaces. In contrast, First-Fit generally allocates smaller chunks and leaves more fragmented free spaces.

The biggest con for Best-Fit is that it is more complex to implement due to the need to maintain and search through a list of free memory chunks to find the best fit for a given allocation request. The First-Fit algorithm, on the other hand, is relatively simpler, as it typically allocates the first chunk of memory that is large enough to

satisfy the request. In the experiment with Datafile size 20, the performance difference was approximately 150 microseconds (μs). Thus, for less demanding applications that prioritize simplicity, First-Fit would be the preferred strategy.

As software applications continue to grow in complexity and scale, the importance of selecting the right memory allocation strategy becomes increasingly evident. As such, based on the findings in this report, the best memory allocation method is the Best-Fit algorithm.