MATHEMATICAL MODELING OF PROCESSES AND SYSTEMS

UDC 689.3.19/16 (045)

[1]**A. S. Yurchenko,**
[2]**N. F. Tupitsin,**
[3]**A. P. Kozlov,**
[4]**Yu. M. Kemenyash**

PERFORMANCE OF DYNAMIC MEMORY ALLOCATION ALGORITHMS

Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine

E-mails: [1]ayurchenko@yahoo.com, [2]nift@mail.ru, [3]ap_kozlov@ukr.net, [4]lindysik99@mail.ru

Abstract—*With the help of modeling there have been investigated 15 algorithms dynamic allocation of the non-paged memory, which, in addition to well-known, include four new algorithms, as well as three algorithms of memory compression.*

Index Terms—Dynamic memory allocation; first-fit; best-fit; fragmentation; simulation.

I. INTRODUCTION

Dynamic memory allocation is used in arranging the memory effectively, which is one of the most critical components of any computer system. It is memory manager who performs the memory allocation process. Two most important aspects of all the algorithms mentioned above are:

– its execution speed;

– effectiveness in storage utilization.

Virtual memory is a graceful solution of the problem of dynamic memory allocation (DMA). Over the past twenty years there has been two main approaches to implement virtual memory. These approaches are segmentation and paging memory allocation, which are reviewed and compared in [1]–[7], where reasons of their development are found. Besides these approaches, there is intensively studied “twins” memory allocation algorithm (featuring small system delays).This research work is continuation the work, described in [7], so all notions are used from there. With the help of modeling there have been investigated 15 algorithms dynamic allocation of the non-paged memory (DANM), which, in addition to well-known, include four new algorithms, as well as three algorithms of memory compression. In this study an attempt is made to analyze both existing and new algorithms DANM, consisting of segment allocation and allocation of memory by “twins” algorithms.

Thus this research work uses table of symbols, description of new proposed algorithms, terminology, the notions of external, internal, and full fragmentation, which are given in [7], where an overview of existing algorithms DANM is also given. Below, the term fragmentation (if its type is not specified) will denote the full fragmentation.

II. ABOUT SIMULATION THE RESEARCHED ALGORITHMS OF DANM

For the first time the program of modeling of the DANM was offered by D. Knuth [4]. In this program it is supposed that original variable value of TIME, characterizing the current time, is equal to zero, all area of memory is the free. This program works as follows.

P1. To move TIME per unit.

P2. To release all occupied segments in system which it is planned to release in case of the current TIME value.

P3. To calculate two values: S – the accidental size and T – accidental “lifetime”, basing on some distributions of probabilities.

P4. To allocate memory of size S to the next requirement, to release the occupied segments at the moment (TIME + 1). To return to P1.

Experiment of simulation comes to an end if memory can’t be allocated to the next requirement.

With the help of this program, algorithms of FIRST-FIT, BEST-FIT and BUDDY are researched. To the same program of modeling K. Shen and I. Peterson [5] are referring for researching of the algorithm of WEIGHTED BUDDY offered by them. However such program is too simplified, for example, this program is closed, when memory can not be given for next requirement. We will distinguish two modes of functioning of algorithm of main memory: saturated mode (external fragmentation takes place) and unsaturated (external fragmentation is not present). Consequently, using Knut’s program it is possible to simulate only unsaturated modes.

In [6] the algorithms of FIRST-FIT, BEST-FIT and NEXT-FIT are probed using the saturated mode.

The "lifetime" of segments (i. e, the time during which the occupied segment of any problem will be located in the allocated memory) is calculated from a probability distribution [6]. It should also be noted that the "lifetime" of the occupied segment will also be corrected by memory allocation algorithm.

Supposing that during work of one algorithm there is one number of occupied segments in memory, and during work other one is other number. Clear, that time of existence of segment in RAM during work of these algorithms is different. Although in [3] calculation temporal system costs is made, however it is by hand, and the conclusions got by Knut are enough rough. In this paper, algorithms DANM are investigated by the proposed program of simulation, which is based on the above-mentioned simulation programs, but is devoid of these disadvantages.

This program allows to research the unsaturated and saturated modes of functioning algorithms of DANM, calculates the average values of external, internal and full fragmentations of memory, utilization coefficient of memory, average number of free and occupied segments, the average time of operation of algorithm of allocation and releasing of memory, and also value of the generalized fragmentation.

"Lifetime" of a occupied segment in case of absence in memory of other segments, the size of this segment and the interval of time between the arriving requests were calculated on the basis of uniform distribution of probability with the corresponding parameters. It was supposed that one occupied segment of a certain size is provided only to each task (request). When processing requests of the CPU the cyclic discipline of RR (Round Robin) [7] was offered.

Whole program of simulation was written using the high-level language (HLL), including the algorithms of DANM. Then these algorithms were written in language of micro-commands for which the runtime of micro-commands is known. After writing the algorithms of DANM using two languages the branches of algorithms programs on HLL were calibrated by time with the help of the appropriate programs written in language of micro-commands. As for such programs it is possible to define exact time of execution of different branches of DANM algorithms.

After such calibration the simulation is executed using HLL, thus exact time of execution of different branches of algorithms of DANM on HLL is already known. As HLL the C++ was chosen, and as the language of micro-commands. Time of execution of micro-commands of register transfers type, reading from RAM to local memory of the CPU is equally in respectively $16\cdot10^{-8}$, $112\cdot10^{-8}$ s.

The allocated memory consists of M_0 – 32-bit words, memory is providing with accurate to the word. Lifetime (in seconds), the size (number of words) of the arriving requirements and an interval (in seconds) between their arrivals are calculated based on uniform distribution of probabilities with the corresponding parameters (B_1, B_2), (B_3, B_4), (B_5, B_6).

III. Results of Simulation of Algorithms of DANM from the Reserve of the Free Memory

The main objective of this work is to show the dependence of the generalized fragmentation on the average time of staying of a segment in RAM. For algorithms of FIRST-FIT, BEST-FIT, NEXT-FIT, WORST-FIT, BUDDY, WEIGHTED BUDDY, FIBONACCI BUDDY, SEGREGATED STORAGE, A4, A5 such results are summarized in Table I. In this and the following tables, in addition to already entered designations – f_e, f_i, f, K and Φ, there are used the new ones – t_n, t_0 and t_c, where by t_n and t_0 – temporary system costs for the allocation and release of memory respectively, and $t_c = t_n + t_0$.

TABLE I

RESULTS SUMMARIZED

Algorithms	$B_2(B_6)$					
	6.0	0.6	0.006	0.0006	0.00006	0.0000006
	0.4	0.04	0.0004	0.00004	0.000004	0.00000004
FIRST-FIT	0.17	0.1694	0.2010	0.4806	0.8861	0.9988
BEST-FIT	0.1314	0.1375	0.1756	0.3795	0.8269	0.9978
NEXT-FIT	0.1759	0.1765	0.2172	0.5279	0.9153	0.9990
WORST-FIT	0.2003	0.1960	0.2443	0.4890	0.8855	0.9987
A5	0.2263	0.2269	0.2465	0.4697	0.8657	0.9985
BUDDY	0.3403	0.3393	0.3551	0.4776	0.7985	0.9974
BUDDY WEIGHTED	0.3050	0.3053	0.3378	0.5777	0.9047	0.9989
BUDDY FIBONACCI	0.2705	0.2728	0.3024	0.4961	0.8709	0.9984
SEGREGATED STORAGE	0.1324	0.1347	0.1579	0.3695	0.8301	0.9977
A4	0.1153	0.1193	0.1500	0.3560	0.8043	0.9975

$M_0 = 4096$; $\Delta l_{min} = 9$; $B_1 = 0$; $B_3 = 5$; $B_4 = 50$; $B_5 = 0$.

Different characteristics of these algorithms, which are of interest, are given in Table II.

Analyzing the results given above, we receive that in case of $B_2 > 0.6$ s algorithms are located in ascending of values of the generalized fragmentation as follows: A4, BEST-FIT, SEGREGATED STORAGE, FIRST-FIT, NEXT-FIT, WORST-FIT, A5, FIBONACCI BUDDU, WEIGHTED BUDDY, BUDDY**.** Thus, smaller value of the generalized fragmentation will be have the algorithms with smaller value of full fragmentation of memory. This result has already been obtained with the help of calculations.

TABLE II

DIFFERENT CHARACTERISTICS OF ALGORITHMS

Algorithms	Characteristics							
	t_n	t_0	t_c	f_e	f_1	f	K	Φ
FIRST-FIT	$1.92\cdot10^{-4}$	$3.403\cdot10^{-5}$	$2.282\cdot10^{-4}$	0.1005	0.06944	0.1699	0.801	0.1700
BEST-FIT	$6.456\cdot10^{-5}$	$5.944\cdot10^{-5}$	$1.240\cdot10^{-4}$	0.06168	0.06970	0.1314	0.8686	0.1314
NEXT-FIT	$2.231\cdot10^{-4}$	$3.468\cdot10^{-5}$	$2.578\cdot10^{-4}$	0.1052	0.07055	0.1758	0.8242	0.1759
WORST-FIT	$1.637\cdot10^{-4}$	$3.382\cdot10^{-5}$	$1.975\cdot10^{-4}$	0.1552	0.04509	0.2003	0.7997	0.2003
A5	$1.006\cdot10^{-4}$	$5.045\cdot10^{-5}$	$1.511\cdot10^{-4}$	0.04522	0.1810	0.2262	0.7738	0.2263
BUDDY	$4.824\cdot10^{-5}$	$2.749\cdot10^{-5}$	$7.573\cdot10^{-5}$	0.05018	0.2901	0.3403	0.6593	0.3403
BUDDY WEIGHTED	$1.295\cdot10^{-4}$	$6.499\cdot10^{-5}$	$1.945\cdot10^{-4}$	0.1787	0.1262	0.3049	0.6951	0.3050
BUDDY FIBONACCI	$9.432\cdot10^{-5}$	$4.008\cdot10^{-5}$	$1.344\cdot10^{-4}$	0.08356	0.1869	0.2705	0.7295	0.2705
SEGREGATED_STORAGE	$7.379\cdot10^{-5}$	$3.818\cdot10^{-5}$	$1.120\cdot10^{-4}$	0.06763	0.06480	0.1324	0.8676	0.1324
A4	$5.395\cdot10^{-5}$	$4.923\cdot10^{-5}$	$1.032\cdot10^{-4}$	0.05829	0.5705	0.1153	0.8847	0.1153

$M_0 = 4096$; $\Delta l_{\min} = 9$; $B_1 = 0$; $B_2 = 6.0$; $B_3 = 5$; $B_4 = 50$; $B_5 = 0$; $B_6 = 0.4$.

In case of B_2 values, there are a lot of smaller 0.6 s, the order of analyzable algorithms layout by increase of their generalized fragmentation significantly changes. Thus smaller value of the generalized fragmentation will be had by algorithms in which smaller temporal expenses are inherent. In case of very small B_2 values all algorithms are poorly differed on values of the generalized fragmentation, these values are approximately equal to 1. The same result was received above.

Let us note that temporary system costs of algorithm of BEST-FIT are less, than at algorithm of FIRST-FIT. This result is opposite to the one received by Knuth. However it is necessary to consider that in [3] algorithms were simulated for that case when there was no external fragmentation of memory. In this work this case was also modeled (the utilization coefficient of memory was approximately equal 0.5).

That temporary system costs of algorithm FIRST-FIT is more, than algorithm's of BEST-FIT in case of the saturated mode, is possible to explain by the following. In the saturated mode the number of failures in memory provision increases. In algorithm of FIRST-FIT to make sure that memory can't be provided, it is necessary to review all list of the free segments, and in algorithm of BEST-FIT it is enough to address to the first free segment of the list which will be the biggest. Besides, as it was shown by modeling, the average number of free segments of algorithm of FIRST-FIT in the saturated mode is equal in 30, and at algorithm BEST FIT – 12.

In a unsaturated mode the memory is allocated in case of each request to the specified algorithms, failures generally aren't present. Temporary system costs of the algorithm of FIRST-FIT are less, than for algorithm of BEST-FIT. Because under unsaturated mode the system costs for maintenance of the list of the free segments starts to play an essential role, arranged in decreasing order of their segment's sizes.

As modeling shows, in case of memory utilization coefficient $K \approx 0.5$ the average numbers of the free segments at both algorithms are identical and equal 17.

In work [6] the big hope it was laid on algorithm NEXT-FIT, however results of modeling showed that temporary system costs of this algorithm aren't less then it was supposed, but more, than at algorithm of FIRST-FIT.

The intermediate algorithm of A5, offered in this work, has temporary system costs less, than FIRST-FIT. This is due to the fact that the last has more a average number of the free segments. It is easy to understand such phenomenon. The modeling has showed that the average number of free segments in the algorithm FIRST-FIT is equal to 30, and in

A5 – 10. However, temporary system costs of the algorithm A5 is greater than of the algorithm BUDDY. If to compare the three algorithms (FIRST-FIT, BUDDY, and A5) in magnitude of complete fragmentation, then, as it was expected, the biggest value of fragmentation has algorithm BUDDY, and the smallest – algorithm FIRST-FIT.

Now compare the algorithms of the family "twins". External fragmentation is the biggest in the algorithm WEIGHTED BUDDY, and the smallest – in the algorithm BUDDY. If to compare these algorithms by their internal fragmentation of memory, they are arranged (in order of increasing of this type of fragmentation) as follows: WEIGHTED BUDDY, FIBONACCI BUDDY, BUDDY. Using the total memory fragmentation the best algorithm is FIBONACCI BUDDY (for $B_2 > 0.6\,\text{s}$ such a situation occurs when these algorithms are compared using generalized fragmentation).

The algorithm WEIGHTED BUDDY, proposed as an alternative to BUDDY algorithm, has the largest temporary system costs among these algorithms. It was assumed that the lower value of internal fragmentation in algorithm WEIGHTED BUDDY in comparing with BUDDY is achieved with the help of a slight increase of temporary system costs (but this fact has not been tested by modeling).

Algorithm WEIGHTED BUDDY was proposed as the modification of BUDDY for further decreasing of internal fragmentation. However, as it was shown by simulation, internal fragmentation in WEIGHTED BUDDY is actually less than that of the algorithm BUDDY. But it is achieved by a significant increase in the temporary system costs. It turned out that the total memory fragmentation, although is less in algorithm WEIGHTED BUDDY than the algorithm BUDDY, but bigger than that in BEST-FIT. At the same time the temporary system costs in the algorithm BEST-FIT are less than in WEIGHTED BUDDY. Thus, the algorithm BEST-FIT is better than the algorithm WEIGHTED BUDDY both by temporary system costs and memory usage.

The algorithm BUDDY has the smallest temporary system costs, but also the worst memory usage, (among simulated algorithms in this study). Summarizing the analysis of the simulation results, shown in the Tables I, II, it should be said that the most preferred are the following algorithms (listed in decreasing order of their importance): A4, BEST-FIT, SEGREGATED STORAGE and FIRST-FIT.

The results of modeling the algorithms of providing a memory from reserve of free memory were obtained for B_4 = 200 and M_0 = 8192. The analysis of these results provides no difficulties.

Particular attention was paid to the influence of the value $\Delta l_{\min}$ on the different characteristics of the algorithms due to the lack of such researches in the literature, however $\Delta l_{\min}$ is non zero in DANM. The results of modeling algorithms FIRST-FIT, BEST-FIT, NEXT-FIT, WORST-FIT, SEGREGATED STORAGE, A4 (for other algorithms $\Delta l_{\min}$ value has no meaning) were obtained for $\Delta l_{\min}$ = 5.20. It is easy to see that the larger the value $\Delta l_{\min}$ is, the more internal fragmentation is, and hence the greater the value of the generalized fragmentation (B_2 > 0.6 s) is. Thus, for the value of the $\Delta l_{\min}$ there must be taken the possible smallest values. However, at values of B_2 > 0.6 s the value of the $\Delta l_{\min}$ should be selected the possible largest values.

IV. Results of Simulation of Compression Memory Algorithms

As it was said above, the three algorithms of memory compaction were simulated – A1, A2, A3. The results of the simulation are shown in Tables 3 and 4. From the analysis of Table 3 it is shown that the algorithms A1, A2, A3 provide minimal value of the external and full fragmentation of memory, but the largest value of the internal fragmentation is in comparison with algorithms of memory allocation from the reserve of a free memory. The last algorithms have a much smaller time system costs in comparison with the simulated algorithms. The value of the smallest generalized fragmentation is in algorithm A2, and most – in A3. These results were obtained for B_2 = 6.0 s; the average value of "lifetime" of the occupied segment is more than critical. Here are the results. If values of the average "lifetime" of the employed segment are larger than critical, algorithm A2 is better, it has a smaller value of the generalized fragmentation. Algorithms A1 and A2 are the algorithms of complete compression of memory when algorithms FIRST-FIT and BEST-FIT are working respectively. In [3], where the memory compression algorithms are analyzed, the preference is given to algorithms, which provides a partial compression. As such an algorithm, the algorithm A3 is taken. However, the results of simulation show this algorithm has the generalized fragmentation more than the algorithms Al, A2.

In Table 3 the results of modeling of algorithms Al, A2, and A3 are given for $\Delta l_{\min}$ = 5. 9 and 20, respectively. Hence it is easy to understand why these algorithms have less the generalized fragmentation at smaller values of $\Delta l_{\min}$. This fact is a consequence of changes in the internal memory fragmentation.

TABLE 3

RESULTS OF MODELING OF ALGORITHMS AL, A2, AND A3

Algorithms	Characteristics							
	t_n	t_0	t_c	f_e	f_l	f	K	Φ
$\Delta l_{\min} = 9,\ B_3 = 5,\ B_4 = 50$								
A1	$1.865 \cdot 10^{-3}$	$4.684 \cdot 10^{-5}$	$1.912 \cdot 10^{-3}$	$5.196 \cdot 10^{-3}$	$5.958 \cdot 10^{-2}$	$6.478 \cdot 10^{-2}$	0.9352	0.06538
A2	$1.693 \cdot 10^{-3}$	$1.118 \cdot 10^{-5}$	$1.704 \cdot 10^{-3}$	$6.761 \cdot 10^{-3}$	$4.480 \cdot 10^{-2}$	$5.156 \cdot 10^{-2}$	0.9484	0.05211
A3	$1.584 \cdot 10^{-3}$	$4.813 \cdot 10^{-5}$	$1.632 \cdot 10^{-3}$	$5.173 \cdot 10^{-3}$	$7.561 \cdot 10^{-2}$	$8.078 \cdot 10^{-2}$	0.9191	0.08130
$\Delta l_{\min} = 5,\ B_3 = 5,\ B_4 = 50$								
A1	$1.957 \cdot 10^{-3}$	$5.678 \cdot 10^{-5}$	$2.014 \cdot 10^{-3}$	$5.431 \cdot 10^{-3}$	$2.961 \cdot 10^{-2}$	$3.504 \cdot 10^{-2}$	0.9650	0.03569
A2	$2.074 \cdot 10^{-3}$	$3.144 \cdot 10^{-5}$	$2.105 \cdot 10^{-3}$	$5.307 \cdot 10^{-3}$	$2.437 \cdot 10^{-2}$	$2.968 \cdot 10^{-2}$	0.9703	0.03536
A3	$1.884 \cdot 10^{-3}$	$5.749 \cdot 10^{-5}$	$1.941 \cdot 10^{-3}$	$5.243 \cdot 10^{-3}$	$3.315 \cdot 10^{-2}$	$3.659 \cdot 10^{-2}$	0.9634	0.03723
$\Delta l_{\min} = 20,\ B_3 = 5,\ B_4 = 50$								
A1	$1.339 \cdot 10^{-3}$	$4.227 \cdot 10^{-5}$	$1.381 \cdot 10^{-3}$	$4.275 \cdot 10^{-3}$	$1.585 \cdot 10^{-1}$	$1.628 \cdot 10^{-1}$	0.8372	0.1631
A2	$1.214 \cdot 10^{-3}$	$2.949 \cdot 10^{-5}$	$1.243 \cdot 10^{-3}$	$4.188 \cdot 10^{-3}$	$1.736 \cdot 10^{-1}$	$1.778 \cdot 10^{-1}$	0.8222	0.1782
A3	$1.085 \cdot 10^{-3}$	$4.346 \cdot 10^{-5}$	$1.128 \cdot 10^{-3}$	$4.156 \cdot 10^{-3}$	$1.728 \cdot 10^{-1}$	$1.770 \cdot 10^{-1}$	0.8230	0.1773
$\Delta l_{\min} = 9,\ B_3 = 5,\ B_4 = 200$								
A1	$2.727 \cdot 10^{-3}$	$2.519 \cdot 10^{-5}$	$2.812 \cdot 10^{-3}$	$1.628 \cdot 10^{-2}$	$6.372 \cdot 10^{-3}$	$2.265 \cdot 10^{-2}$	0.9773	0.02357
A2	$2.617 \cdot 10^{-3}$	$3.330 \cdot 10^{-5}$	$2.650 \cdot 10^{-3}$	$1.596 \cdot 10^{-2}$	$8.221 \cdot 10^{-3}$	$2.418 \cdot 10^{-2}$	0.9758	0.02504
A3	$2.391 \cdot 10^{-3}$	$2.683 \cdot 10^{-5}$	$2.418 \cdot 10^{-3}$	$1.620 \cdot 10^{-2}$	$1.108 \cdot 10^{-3}$	$2.728 \cdot 10^{-2}$	0.9727	0.02806

$M_0 = 4096$; $B_1 = 0$; $B_2 = 6.0$; $B_5 = 0$; $B_6 = 0.4$.

Table 4 shows the values of the generalized fragmentation for algorithms Al, A2 and A3 (see for a comparison a similar dependence for A4). The table shows that the algorithms Al, A2 and A3 have the more meaning of critical "lifetime" of the occupied segment. But the values of the generalized fragmentation the algorithms Al, A2 and A3 have ess than the A4 at a mean "lifetime" of the occupied segment more bigger critical "lifetime". Also the dependences of the generalized fragmentation for the algorithms FIRST-FIT and A1 are given at M_0 – 8192. These relationships indicate that an increase in the size of the allocated memory increases critical "lifetime" of the occupied segment.

TABLE 4

THE VALUES OF THE GENERALIZED FRAGMENTATION FOR ALGORITHMS AL, A2 AND A3

Algorithms	$B_2(B_6)$					
	6.0	0.6	0.006	0.006	0.0006	0.00006
	(0.4)	(0.04)	(0.004)	(0.0004)	(0.00004)	(0.000004)
$M_0 = 4096$						
A1	0.06538	0.05824	0.1197	0.3204	0.8721	0.988
A2	0.05211	0.07121	0.1206	0.4067	0.8647	0.9856
A3	0.08129	0.8573	0.1256	0.4187	0.8789	0.9849
$M_0 = 8192$						
FIRST_FIT	0,1135	0.1143	0.1247	0.2069	0.5418	0.9255
A1	0.06789	0.07812	0.1820	0.6202	0.9390	0.9935

$\Delta l_{\min} = 9,\ B_1 = 0,\ B_3 = 5,\ B_4 = 50,\ B_5 = 0.$

V. SIMULATION OF ALGORITHMS A6 AND A7

Table 5 gives the average temporary system costs of algorithms FIRST-FIT and A6 for different coefficients of memory usage (K). From this table it is seen that if $K \sim 0.8$; 0.83 the algorithm A6 has less values of temporary system overhead than the algorithm FIRST-FIT. For values of $K \sim 0.45$; 0.55; 0.65; 0.75 the algorithm FIRST-FIT has the less time costs. These results can be interpreted as follows. For values of $K << 0.8$ large temporary system costs are inherent for algorithm A6, but at values of $K > 0.8$ – for algorithm FIRST-FIT. It confirms the assumption, made above, that for values of K greater than a certain boundary, the best algorithm is the algorithm A6.

In Table 6 the results of the simulation algorithm FIRST-FIT, A6 and A7 are shown. The modeling was performed as follows. For 1000 memory allocation the K is approximately 0.45, then the value of K – 0.83. The value of temporary system costs of these three algorithms were taken when the number of memory allocations were equal to 1000, 2000 and 3000. As seen from the results of modeling Table 6, the algorithm A7 adapts to changes of K, while it has the least temporary system costs compared with FIRST-FIT and A6.

TABLE 5

THE AVERAGE TEMPORARY SYSTEM COSTS OF ALGORITHMS FIRST-FIT AND A6 FOR DIFFERENT COEFFICIENTS OF MEMORY USAGE (K)

Algorithms	K					
	4.5	5.5	6.5	7.5	8.0	8.3
FIRST_FIT A6	$5.267\cdot10^{-5}$ $7.138\cdot10^{-5}$	$5.517\cdot10^{-5}$ $7.463\cdot10^{-5}$	$6.597\cdot10^{-5}$ $8.653\cdot10^{-5}$	$6.922\cdot10^{-5}$ $9.173\cdot10^{-5}$	$1.146\cdot10^{-4}$ $1.103\cdot10^{-4}$	$1.942\cdot10^{-4}$ $1.379\cdot10^{-5}$

$M_0 = 4096$, $\Delta l_{min} = 9$, $B_1 = 0$, $B_2 = 6.0$, $B_3 = 5$, $B_4 = 50$, $B_5 = 0$, $B_6 = 0.4$.

TABLE 6

THE RESULTS OF THE SIMULATION ALGORITHM FIRST-FIT, A6 AND A7

Algorithms	Number of memory requests		
	1000	2000	3000
FIRST-FIT	$5.267\cdot10^{-5}$	$1.0708\cdot10^{-4}$	$1.287\cdot10^{-4}$
A6	$7.138\cdot10^{-5}$	$9.518\cdot10^{-5}$	$1.0492\cdot10^{-4}$
A7	$6.598\cdot10^{-5}$	$9.194\cdot10^{-5}$	$1.0059\cdot10^{-4}$

$M_0 = 4096$, $\Delta l_{min} = 9$, $B_1 = 0$, $B_2 = 6.0$, $B_3 = 5$, $B_4 = 50$, $B_5 = 0$, $B_6 = 0.4$.

CONCLUSION

The concept of a generalized fragmentation is introduced. According to the proposed criterion the best algorithm is the algorithm with a smaller value of such fragmentation. With the help of modeling there have been investigated 15 algorithms DANM, which, in addition to well-known, include four new algorithms, as well as three algorithms of memory compression.

One of the proposed algorithms, combining the ideas of algorithms BEST-FIT and SEGREGATED STORAGE, is better these algorithms. They have a smaller value of the generalized fragmentation among the algorithms of memory allocation from a reserve of free memory.

The research also shows the expediency of application in DANM of the algorithms of complete (not partial) compression of memory, since such algorithms, although have a relatively large temporary system costs, but the value of the generalized fragmentation is less than any of the algorithms of DANM.

Given calculations are based on the rule of “fifty percent”, whose validity for algorithms FIRST-FIT and BEST-FIT was also verified by simulation.

The further development of this study, probably, is the study of DANM taking into account the exchange of information between the RAM and the secondary memory.

REFERENCES

[1] Dr. Taylor, (2003). *Deepak Agrawal Memory Management*. http://msoe.us/taylor/cs384/agrawald.pdf.

[2] P. J. Denning, “Virtual memory”, *Comput. Surv.* 28(4), pp. 213–216. 1996.

[3] P. J. Denning (2008, January). Virtual memory, *Naval Postgraduate School*, Monterey, California, http://denninginstitute.com/pjd/PUBS/ENC/vm08.pdf.

[4] D. E. Knuth, *The Art of Computer Programming*, vol. 1-4A Boxed Set Hardcover. March 3, 2011.

[5] K. K. Shen, and J. L. Peterson. “A weighted buddy method for dynamic storage allocation”. *Comm. ACM*, 17, no. 10, pp. 558–562. 1974.

[6] Carter Bays. “A comparison of next-fit, first-fit and best-fit”. *Comm. ACM*, 20, no. 3, pp.191–192. 1977.

[7] V. M. Sineglazov, A. S. Yurchenko, N. F.Tupitsin, and A. P. Kozlov, “The dynamic segment memory allocation algorithms”. *Electronics and Control Systems,* no. 2(44). Kyiv: NAU, pp. 114–120. 2015.

Received July 7, 2015

Yurchenko Alexander. Candidate of Engineering. Assosiate professor.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine.
Education: Moscow Phisics-thechnical Institute, Moscow, Russia (1975).
Research area: operating system, dynamic allocation memory.
Publication: 57.
E-mail: nift@mail.ru

Tupitsin Nikolay. Candidate of Engineering. Assosiate professor.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine.
Education: Moscow Phisics-thechnical Institute, Moscow, Russia (1975).
Research area: dynamic of flight, experimental methods of aerodynamic, aviation simulators.
Publication: 77.
E-mail: nift@mail.ru

Kozlov Anatoliy. Candidate of Engineering. Assosiate Professor.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kiev, Ukraine
Education: Kiev State University named T. G. Shevchenko, Kyiv, Ukraine (1965).
Research interests: capacitive transducers with non-uniform electromagnetic field. Capacitive meters of parameters small altitude of the flight aircraft. The use of capacitive transducers in automatic control small-altitude of the flight aircraft.
Publications: 49.
E-mail: ap_kozlov@ukr.net

Kemenyash Yuriy Sen. Teacher
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine
Education: National Aviation University, Kyiv, Ukraine (1996).
Research area: Identification of Complex Systems.
Publications: more than 16 papers.
E-mail: lindysik99@mail.ru

О. С. Юрченко, М. Ф. Тупіцин, А. П. Козлов, Ю. М. Кеменяш. Моделювання динамічних алгоритмів розподілу пам'яті

Досліджено п'ятнадцять алгоритмів динамічного розподілу несторінкової пам'яті за допомогою моделювання, до яких крім відомих алгоритмів включено чотири нових алгоритми, а також три алгоритми стиснення пам'яті.
Ключові слова: динамічний розподіл пам'яті FIRST-FIT; BEST-FIT; фрагментація; моделювання.

Юрченко Олександр Сергійович. Кандидат технічних наук. Доцент.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Московський фізико-технічний інститут, Москва, Росія (1975).
Напрям наукової діяльності: операційні системи, динамічний розподіл пам'яті.
Кількість публікацій: 57.
E-mail: nift@mail.ru

Тупіцин Микола Федорович. Кандидат технічних наук. Доцент.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Московський фізико-технічний інститут, Москва, Росія (1975).
Напрям наукової діяльності: динаміка польоту, експериментальні методи аеродинаміки.
Кількість публікацій: 77.
E-mail: nift@mail.ru

Козлов Анатолій Павлович. Кандидат технічних наук. Доцент.
Кафедра комп'ютерно-інтегрованіх комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Київський державний університет имени Т. Г. Шевченка, Київ, Україна (1965).
Напрям наукової діяльності: ємнісні перетворювачі з неоднорідним електромагнітним полем. Ємнісні прилади вімірювання геометричних параметрів мало висотного польоту повітряного судна. Використання ємнісніх перетворювачів в системах автоматичного керування мало висотним польотом повітряного судна.
Публікації: 49.
E-mail: ap_kozlov@ukr.net

Кеменяш Юрій Михайлович. Старший викладач.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна (1996).
Напрям наукової діяльності: ідентифікація складних систем.
Кількість публікацій: більше 16 наукових робіт.
E-mail: lindysik99@mail.ru

А. С. Юрченко, Н. Ф. Тупицин, А. П. Козлов, Ю. М. Кеменяш. Моделирование динамических алгоритмов распределения памяти

Исследованы пятнадцать алгоритмов динамического распределения нестраничной памяти с помощью моделирования, в которые кроме известных алгоритмов включены четыре новых алгоритма, а также три алгоритма сжатия памяти.

Ключевые слова: динамическое распределение памяти; FIRST-FIT; BEST-FIT; фрагментация; моделирование.

Юрченко Александр Сергеевич. Кандидат технических наук. Доцент.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Московский физико-технический институт, Москва, Россия (1975).
Направление научной деятельности: операционные системы, динамическое распределение памяти.
Количество публикаций: 57.
E-mail: nift@mail.ru

Тупицин Николай Федорович. Кандидат технических наук. Доцент.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Московский физико-технический институт, Москва, Россия (1975).
Направление научной деятельности: динамика полета, экспериментальные методы аэродинамики.
Количество публикаций:77.
E-mail: nift@mail.ru

Козлов Анатолий Павлович. Кандидат технических наук. Доцент.
Кафедра компьютерно-интегрированых комплексов, Национальный авиационный университет, Киев, Украина
Образование: Киевский государственный университет имени Т. Г. Шевченко, Киев, Украина (1965).
Направление научной деятельности: емкостные преобразователи с неоднородным электромагнитным полем. Емкостные устройства измерения геометрических параметров мало высотного полета воздушного судна. Использование емкостных преобразователей в системах автоматического управления мало высотным полетом воздушного судна.
Публикации: 49.
E-mail: ap_kozlov@ukr.net

Кеменяш Юрий Михайлович. Старший преподаватель.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Киевский авиационный университет, Киев, Украина (1996).
Направление научной деятельности: идентификация сложных систем.
Количество публикаций: более 16 научных работ.
E-mail: lindysik99@mail.ru