UDC 689.3.19/16 (045)

[1]V. M. Sineglazov,
[2]A. S. Yurchenko,
[3]N. F. Tupitsin,
[4]A. P. Kozlov

THE DYNAMIC SEGMENT MEMORY ALLOCATION ALGORITHMS

Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua, [2]ayurchenko@yahoo.com, [3]nift@mail.ru, [4]ap_kozlov@ukr.net

Abstract—*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. Fifteen algorithms of dynamic allocation of the non-paged memory are considered, 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

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], [2] where reasons of their development are found. Besides these approaches, there is intensively studied "twins" memory allocation algorithm (featuring small system delays).

At present time, in computing systems the emphasis is on the paged memory allocation, although there are a very good systems with a "clear" segmentation. To fill the gap in the literature concerning dynamic allocation of the non-paged memory (DANM), in this study an attempt is made to analyze both existing and new algorithms of DANM, consisting of segment allocation and allocation of memory by "twins" algorithms.

This study uses the notion of external, internal, and full fragmentation, which is given in [1], 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.

In the process of solving problems on computers, it's needed to allocate random access memory (RAM) for data, commands, results of intermediate calculations, and so on. We assume that for solving of any problem only one segment is enough, and for different tasks sizes of the corresponding segments will be different. Before solving the problem with the help of some algorithm of providing RAM, there will be allocated such area of memory, into which there can placed a segment of this problem. When the task solution ends, memory, occupied by it. is given to the reserve of free memory (i.e. memory, which may be used for other tasks) using an algorithm of releasing of used memory. The process of impact of some area of random access memory into the free memory reserve will be called memory release. All investigated in this study algorithms DANM belong to the allocation of random access memory, so the word "memory" or "allocated memory" will mean random access memory.

In addition, for ease of explanation as a synonym to the word "task" will be used the word "request".

II. DYNAMIC SEGMENT ALLOCATION OF MEMORY AND RULE OF "FIFTY PERCENT"

For the dynamic segment allocation D. Knuth [3] proved the "fifty percent" rule, which sets the ratio between the number of occupied and free segments. According to this rule, if the memory system tends to the equilibrium state, in which system has an average of N occupied segments, than the average number of free segments M is approximately $p\dfrac{N}{2}$, where p is the probability, that another memory request is provided due to the free segment, the difference between the size of which and required larger than $\Delta l_{\min}$ words ($\Delta l_{\min}$ is introduced into [1] and is equal to the maximum acceptable internal fragmentation on a occupied segment and the minimum size of the free segment).

With the help of this rule for $p = 1$ were obtained some estimates of fragmentation in dynamic segment allocation. In this study, a number of estimates for the General case of $p \neq 1$ will be received.

If we denote M_0 is the size of the allocated memory in words, and S_0 is the average size of the used segment (the law of the allocation of occupied segment and the rule of determination of S_0 are supposed to be the same as in [3]), so for the relative losses of the internal f_i, external f_e, and complete f_1 fragmentation when $p \neq 1$ have the following expression:

$$f_i = N(1-p)\frac{\Delta l_{\min}}{2M_0}, \quad (1)$$

$$f_e = \frac{NpK_{av.}S_0}{2M_0}, \quad (2)$$

$$f_1 = f_e + f_i, \quad (3)$$

where K_{av} is the ratio between the average size of the free segment and S_0, based on the "fifty percent" rule:

$$M_0 = NpS_0 + N(1-p)\left(S_0 + \frac{\Delta l_{\min}}{2}\right) + \frac{N}{2}pK_{av.}S_o = N\left[S_0 p\left(1+\frac{K_{av.}}{2}\right) + (1-p)\left(S_0 + \frac{\Delta l_{\min}}{2}\right)\right]. \quad (4)$$

Expressions for the relative losses on external and complete fragmentation can be easy obtained from qualitative arguments:

$$f_1 = \frac{M_0 - NS_0}{M_0}, \quad (5)$$

$$f_e = f_1 - f_i = \frac{(M_0 - NS_0) - N(1-p)\frac{\Delta l_{\min}}{2}}{M_0}. \quad (6)$$

The other expression is also true for f_e:

$$f_e = \frac{M_0 - pNS_0 - (1-p)N\left(S_0 + \frac{\Delta l_{\min}}{2}\right)}{M_0}, \quad (7)$$

which after simple transformations becomes (6). Substituting $K_{av.}$ from (4) in (2) it is easy to show that expressions (2) and (3) becomes (5), (6), respectively.

III. DANM ALGORITHMS

The totality of the segmented memory allocation algorithms can be divided into two classes: algorithms of the memory allocation from free memory reserves and algorithms of reallocation of used memory. To the first is included the following: FIRST-FIT (the first suitable), NEXT-FIT (the first suitable with the "wandering pointer"), BEST-FIT (the most suitable), WORST-FIT (the least suitable). Algorithms for memory reallocation consist of algorithms performing as moving the occupied segments in the random access memory, and as their displacement in the secondary memory.

Algorithms DANM contain also a group of three algorithms: BUDDY ("equal twins"), WEIGHTED BUDDY ("weighted twins"), FIBONACCI BUDDY ("Fibonacci–twins"). The algorithm which combines the ideas of BUDDY and FIRST-FIT algorithms is SEGREGATED STORAGE algorithm (separate storage). An overview of these algorithms is given in [5]–[10].

In the this study the results of the DANM algorithms modeling are described. The need of such research is due to the fact that the literature does not have simple answer for the question which algorithm DANM is better.

Four algorithms of providing the memory from the reserve of free memory are modeled: FIRST-FIT, BEST-FIT, NEXT-FIT and WORST-FIT, which share the same feature. If the difference between the sizes of the found segment and required is larger than $\Delta l_{\min}$ words, this segment is split into two, one of which is providing the request, and the second is included in the free segments list.

Otherwise whole segment is given to the request. In the memory release algorithms which work with the described algorithms, the upper and lower segments are checked with respect to the released one. If one of them (or both) is free (are free), it is (or they are) removed from the list of free segments, then the neighboring free segments (if they exist) combine together with released in one resulting segment, and the latter is included in the list of free segments.

In this study, three compression algorithm memory are also modeled. The first algorithm (**algorithm A1**) works as the algorithm FIRST-FIT, however, when the memory can not be provided, full compression is performed. The second algorithm (**algorithm A2**) is similar to the first one, but it provides memory as the algorithm BEST-FIT. The third algorithm (**algorithm A3**) performs partial compression of memory. When the memory can not be provided using the FIRST-FIT, compression is performed until it's formed a free segment of size larger than required.

Three "twins" group algorithms are also modeled: BUDDY, WEIGHTED BUDDY: FIBONACCI-BUDDY. The SEGREGATED STORAGE algorithm is modeled with the following values of k and a_i, $k \geq i \geq 0$:

$$k = 3, \qquad a_0 = 0, \qquad a_1 = \frac{B_4 - B_3}{3},$$
$$a_2 = \frac{2}{3}(B_4 - B_3), \qquad a_3 = M_0, \tag{8}$$

where B_3, B_4 are the minimum and maximum requested sizes respectively. Besides these 11 DANM algorithms in this work there are proposed and investigated four new algorithms – A4, A5, A6 and A7.

Algorithm A4. As already mentioned, the algorithm SEGREGATED STORAGE combines the ideas of the algorithms FIRST-FIT and BUDDY. It would be interesting to use the ideas of the algorithm SEGREGATED STORAGE for algorithm BEST-FIT.

For this purpose, the algorithm A4 is offered, which requires three lists (the number of lists may be different) of free segments. In these lists free segments are ordered in descending order of their sizes, wherein the list i ($3 \geq i \geq 0$) will have all free segments sizes of which are larger than a_{i-1} and less than or equal to a_i where a_{i-1} and a_i are defined in (8) with $k = 3$. When it is needed to provide the memory size of *S*, list *i* is selected, for which $a_i \geq S > a_{i-1}$. Further there is the search of suitable free segment in list *i* like in the work of algorithm BEST-FIT. If list *i* doesn't have the suitable free segment (the list was empty), then $i < 3$ occurs the transition to the list $i + 1$, with which we act as with the list *i*. If all the lists from i to k are empty, then the memory could not be provided to request.

When in one of the lists the suitable segment was found, it is removed from the corresponding list, while the remaining part of the segment is the difference between the size of the appropriate segment and size of requests (if it cannot be neglected, and to give all the segment to request) is formalized as a new free segment and is included in the list of free segments with the corresponding a_i and the segment of the necessary size is provided to request. When releasing memory the neighboring lower and upper segments are analyzed by their physical addresses (with respect to the considered one). If one of them is free, it is removed from the corresponding list. If both segments are free, they are both removed from the lists. Next, considering segments are combined into one, and the resulting segment is entered into the appropriate list. When both neighbors (with respect to released) segments are occupied, the union does not occur, and released segment is entered into the corresponding list.

This algorithm should have the same values of external, internal, and full fragmentation of memory as the algorithm BEST-FIT has, system costs of the proposed algorithm should be less than that of algorithm BEST-FIT.

Algorithm A5. Since the algorithms BUDDY and FIRST-FIT are not quite satisfactory (the first of them have small temporary system costs, but large losses on internal and external fragmentation, and the second has great system delay when making best use of memory), compromise algorithm A5 is proposed, similar to the FIRST-FIT, but with one feature.

Let to the request of size *S* the suitable free segment size S_2 ($S_2 > S$) has already been found. Then when $S_2 - S > K_0 S$ the founded free segment is split into two, with sizes *S* and ($S_2 - S$), first is given to request and the second is included in the list of free segments. Otherwise, all found segment is allocated for the request. Value $K_0 = 1$, while the possible values K_0 are from the interval (0, 1).

It is expected that this algorithm will have a better memory usage than the BUDDY algorithm, and temporary system costs are less than FIRST-FIT.

Algorithms A6, A7. One of the major disadvantages of algorithm FIRST-FIT is a lengthy search for a suitable segment in the list of free segments. Moreover, the search is performed even when the list has no suitable segment.

The main idea of the proposed algorithm segmental memory allocation is the elimination of search in the list of free segments, when there is no suitable segment. To achieve this, the algorithm introduces a variable $S_{\max}$ which stores the size of the largest free segment and is constantly adjusted by this algorithm. When a new request is coming, first of all the memory $S_{\max}$ is compared with the size of the received request. If $S_{\max}$ more size of request, it is looking for a suitable segment, otherwise the search is not performed. A more detailed description of the proposed algorithm (algorithm A6) looks as follows.

The Algorithm A6. Let *S* be the size of the required memory, S_1 and A_1 respectively, the size and address of the considered current free segment when searching; the variable FREE points to the first free segment; S_2 and B_1 – the variables necessary for the algorithm A6.

A6.1. If $S < S_{\max}$ (the list has suitable segment), then the transition to the A6.2, otherwise the algorithm has finished his work, unsuccessfully.

A6.2. A_1 := FREE (the variable A_1 is assigned the address of the first free segment).

A6.3. If $B_1 \neq 0$, then go to A6.4, otherwise – to A6.7.

A6.4. If S < S_1, then there is a giving of memory to the request, otherwise go to A6.7.

A6.5. If $S_1 = S_{\max}$, $B_1 : = 0$, go to A6.8, otherwise to A6.6.

A6.6. Successful exit from the algorithm A6.

A6.7. If $S_1 > S_2$, then $S_2 := S_1$.

A6.8. A_1 := address of next free segment in the list.

A6.9. If $A_1 = 0$ (was considered the last free segment in the list), $S_{\max} : = S_2$, a successful exit from the algorithm A6, otherwise go to A6.3.

This algorithm A6 performs simultaneously a search of new, with the largest size, and the right one of free segments. Search for free segment with the largest size is performed when to request is allocated the memory with the help of segment sized $S_{\max}$, if this happens, the search is performed till the end of the list. Additional operation in procedure of release of memory would be following. Let S_{rel} is the size of the released segment, if $S_{rel} > S_{\max}$, then $S_{\max} := S_{rel}$ otherwise to do nothing.

The algorithm of A6 is an algorithm of FIRST-FIT this superstructure which increases temporal system costs of provision of memory. However they are justified if the probability (P_{fail}) of an unsuccessful output from algorithm of A6 is rather great. This probability is defined by the ratio of number of unsuccessful outputs (failures) to number successful ones, measured for a long time. Thus it is supposed that the system of DMA is in an equilibrium state.

We will compare algorithms of FIRST-FIT and A6. Values of external, internal and full fragmentations at them are identical at any moment, however temporal system costs are different. If the probability P_{fail} is small, additional temporal system costs in algorithm of A6 will be very noticeable in comparison with algorithm of FIRST-FIT, therefore

$$T_{A6} > T_{\text{F-F}}, \tag{9}$$

where T_{A6} and $T_{\text{F-F}}$ – the temporal system costs for providing memory N_1 (rather big) to requirements by using the algorithms of A_6 and FIRST-FIT.

On the contrary, if the probability P_{fail} is great, the operation of algorithm of A6 quite often will be reduced to checking of $S < S_{\max}$ and to an exit from algorithm of A6 without search in the list, therefore

$$T_{\text{F-F}} > T_{A6}, \tag{10}$$

By modeling it's possible to determine the critical value P_{fail} in case of which equality of $T_{\text{F-F}} = T_{A6}$. takes place. With the probability P_{fail}. it is possible to connect memory-utilization factor K, determined by the ratio of the size of used memory to the size of all allocated one. The larger K, the greater the probability P_{fail} is.

Since it is impossible to answer clearly, what algorithm is better: FIRST-FIT or A6 (in case of small values of K it's better FIRST-FIT, in case of big – A6), it would be good if the system of DMA itself in case of values K, smaller some critical K_3, was set up for algorithm of FIRST-FIT, in case of values K, larger than K_3 – for algorithm of A6. As such algorithm here it is offered the following.

Algorithm of A 7

A7.1. If $K < K_3$, than works algorithm of FIRST-FIT, an exit from A7, otherwise go to A7.2.

A7.2. If the algorithm of FIRST-FIT worked before it, to appropriate $S_{\max}$ value of all size of the allocated memory.

A7.3. The algorithm of A6 is working, an exit from algorithm of A7.

For coordination of algorithms of FIRST-FIT and A6 in the integrated algorithm of A7 it is necessary to make little change in A6, when the value of all memory is appropriated to the $S_{\max}$. If thus it appears that in the list of the free segments there isn't suitable one, it is necessary to provide an exit from algorithm of A6. Transition from algorithm of A6 to FIRST-FIT in algorithm of A7 is carried out quite simply.

IV. About criterion of comparing different algorithms of DANM

Algorithms of DANM are usually evaluated according to the main characteristics: temporary system costs for algorithm execution and value of fragmentation of the memory arising by its operation . For known nowadays algorithms of DANM – FIRST-FIT, BEST-FIT, NEXT-FIT, WORST-FIT, BUDDY, WEIGHTED BUDDY, FIBONACCI BUDDY and SEGREGATED STORAGE – the estimates according to these characteristics are not systematized, the literature sources give (even for the most known algorithm of FIRST-FIT and BEST-FIT) contradictory results [3], [4]. The reason is that for comparing of different algorithms there is no common criterion considering both named characteristics. The real operation represents the attempt to fill this gap.

Time of the central processor (CP) consists of time intervals, spent directly for the solution of tasks, and the intervals of time, necessary for provision and release of the memory occupied by different tasks. Let's assume that fragmentation of memory during intervals of time for provision and releasing of memory is equal to the size of all memory. Let during rather long interval of time (0, *T*) there executes the solution of many tasks, and let E_i – value of fragmentation of memory in interval (t_i, t_{i+1}), and in time intervals (t_j, t_{j+1}) the allocation or release of memory happens. Then taking into account the designations given above we will determine the value of the generalized fragmentation of memory in the following way:

$$\Phi = \frac{\sum_i E_i(t_{i+1} - t_i) + M_0 \sum_j (t_{j+1} - t_j)}{M_0 T}, \qquad (11)$$

where the first sum is executed on all intervals of time, during which the problems were solving, the second sum – in all intervals in which allocation or release of memory was solved.

As criterion for comparing of different algorithms of DANM expression (11) is offered. When comparing two algorithms the best will be algorithm with smaller value of Φ. It can shown that

$$\Phi = \frac{ET + M_0 t}{M_0(T + t)}, \qquad (12)$$

where *T* is the average time of a segment being in memory in case of absence in it other occupied segments, *t* is the average time necessary for one allocation of memory and one releasing, *E* is average fragmentation of memory without tempory system expenses.

As a rule, for algorithms of DANM the following statement is correct: from two algorithms the one having smaller value of fragmentation of memory will have bigger temporal system expenses. Let's look, what conclusion can be drawn from this statement if to compare algorithms on values Φ.

Let

$$\Phi_1 = \frac{E_1 T + M_0 t_1}{M_0(T + t_1)} \quad \text{and} \quad \Phi_2 = \frac{E_2 T + M_0 t_2}{M_0(T + t_2)}.$$

Be values of the generalized fragmentation for the first and second algorithms respectively, and let $E_1 > E_2$, $t_1 < t_2$ then $\Phi_1 > \Phi_2$ under

$$T > \frac{(M_0 - E_1)t_2 - (M_0 - E_2)t_1}{E_1 + E_2} > 0,$$

in case of

$$T > \frac{(M_0 - E_1)t_2 - (M_0 - E_2)t_1}{E_1 + E_2} > 0,$$

the following inequality holds $\Phi_1 < \Phi_2$.

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. 15 algorithms DANM are considered, which, in addition to well-known, include four new algorithm, 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 then 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] Denning P. J. (1996). Virtual memory, Comput. Surv. 28(4), pp. 213–216.

[2] Denning P. J. (2008). VIRTUAL MEMORY Naval Postgraduate School, Monterey, California January, http://denninginstitute.com/pjd/PUBS/ENC/vm08.pdf.

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

[4] Shore, J. E. (1975). On the external storage fragmentation produced by first-fit and best-fit allocation strategies. Comm. ACM, 18, no. 8, pp. 433–440.

[5] Buddy memory allocation. http://en.wikipedia.org/wiki/Buddy_memory_allocation

[6] Tertiary Buddy System for Efficient Dynamic Memory Allocation DIVAKAR YADAV AK SHARMA @ http://www.wseas.us/e-library/conferences/2010/Cambridge/SEPADS/SEPADS-09.pdf

[7] Hsu, H. T. (1976). A note on weighted buddy systems for dynamic storage allocations. 16, pp. 378–382.

[8] Marchand, A.; Balbastre, P.; Ripoll, I.; Masmano, M. and Crespo, A. "Memory Resource Management for Real-Time Systems". *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference* on, On page(s): pp. 201–210.

[9] Survey Report on Memory Allocation Strategies for Real Time Operating System in Context with Embedded Devices http://www.ijera.com/papers/Vol_2_issue3/GR2311511156.pdf

[10] MS Johnstone "Fragmentation Problem: Solved?" *In Proc. of the Int. Symposium on Memory Management (ISMM'98)*, Vancouver, Canada. ACM Press.

Received 22 April 2015

Sineglazov Viktor. Doctor of Engineering. Professor.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant.
Publications: more than 500 papers.
E-mail: svm@nau.edu.ua

Yurchenko Alexander. Candidate of Engineering. Assistant 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: 56.
E-mail: nift@mail.ru

Tupitsin Nikolay. Candidate of Engineering. Assistant 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: 76.
E-mail: nift@mail.ru

Kozlov Anatoliy Pavlovich. 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: 48.
E-mail: ap_kozlov@ukr.net

В. М. Синєглазов, О. С. Юрченко, М. Ф. Тупіцин, А. П. Козлов. Алгоритми динамічного сегментного розподілу пам'яті

Введено поняття узагальненої фрагментації. Згідно введеному критерію алгоритм з меншим значенням цієї величини вважається кращим. Розглянуто п'ятнадцять алгоритмів динамічного розподілу несторінкової пам'яті, до яких крім відомих алгоритмів включено чотири нових алгоритми.

Ключові слова: Динамічний розподіл пам'яті first-fit; best-fit; фрагментація; моделювання.

Синєглазов Віктор Михайлович. Доктор технічних наук. Професор.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 500 наукових робіт.
E-mail: svm@nau.edu.ua

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: 56.
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: 76.
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 наукових інтересів: Ємнісні перетворювачі з неоднoріднім електромагнітнім полем. Ємнісні прилади вімірювання геометричних параметрів мало висотного польоту повітряного судна. Використання ємнісніх перетворювачів в системах автоматичного управління мало висотним польотом повітряного судна.
Публікації : 48 .
E-mail: ap_kozlov@ukr.net

В. М. Синеглазов, А. С. Юрченко, Н. Ф. Тупицин, А. П. Козлов. Алгоритмы динамического сегментного распределения памяти

Введено понятие обобщенной фрагментации. Согласно введенному критерию алгоритм с меньшим значением этой величины считается лучшим. Рассмотрены пятнадцать алгоритмов динамического распределения нестраничной памяти, которые помимо известных алгоритмов включают четыре новых алгоритма.
Ключевые слова: динамическое распределения памяти. first-fit ; best-fit , фрагментация, моделирование.

Синеглазов Виктор Михайлович. Доктор технических наук. Профессор.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Киевский политехнический институт, Киев, Украина (1973).
Направление научной деятельности: аэронавигация, управления воздушным движением, идентификация сложных систем, ветроэнергетические установки.
Количество публикаций: более 500 научных работ.
E-mail: svm@nau.edu.ua

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:56.
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:76.
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).
Область научных интересов: 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: 48
E-mail: ap_kozlov@ukr.net