

ВИЗНАЧЕННЯ ПАРАМЕТРУ ЕФЕКТИВНОСТІ МЕТОДІВ РЕАЛІЗАЦІЇ ДИНАМІЧНОГО РОЗПОДІЛУ ПАМ'ЯТІ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Дана стаття присвячена визначенню параметра ефективності роботи програмної реалізації методів динамічного розподілу пам'яті. Параметр роботи програмної реалізації динамічного розподілу пам'яті залежить від трьох основних показників роботи комп'ютерної системи. Такими показниками є: час роботи програми, об'єм необхідної пам'яті та значення параметра фрагментації. В результаті роботи було визначено, що між характеристикою ефективності та зазначеними вище показниками роботи комп'ютерної системи спостерігається обернено пропорційна залежність.

Ключові слова: фрагментація, ефективність роботи комп'ютерної системи, динамічний розподіл пам'яті, виділення/звільнення динамічної пам'яті, час роботи програми, об'єм використовуваної пам'яті.

ANATOLIY STEPANOVICH VASURA, ANNA IGORIVNA LISOVENKO
Vinnitsa national technical university

DETERMINATION OF A PARAMETER PERFORMANCE OF THE METHODS IMPLEMENTATION OF A DYNAMIC MEMORY ALLOCATION IN A COMPUTER SYSTEM

Abstract – This article contains defining performance parameter of a dynamic memory allocation in a computer system. The performance parameter depends on a time of a work program, a size of memory and a fragmentation of memory. The performance parameter has inverse relationship on a time of a work program, a size of a memory and a fragmentation of memory.

Keywords: a fragmentation, performance of a computer system, a dynamic memory allocation, allocation/deal location dynamic memory, a time of a work program, a size of memory.

Актуальність. Сучасні комп'ютерні програми не можуть мати необхідний об'єм пам'яті, так як вхідні дані можуть мати різний розмір. Крім того, такі об'єктно-орієнтовані мови як C++ часто використовують для своєї роботи динамічну пам'ять [1].

Зазвичай операційні системи надають системній бібліотеці підпрограми та фізичний простір пам'яті під виділення та звільнення динамічної пам'яті для програм. Динамічне виділення та звільнення пам'яті може становити близько 60% від загального часу виконання програми. Збільшення попиту програм на динамічну пам'ять призвело до пошуку більш ефективних алгоритмів розподілу, що мінімізують час роботи програми та об'єм використовуваної пам'яті. Другим гострим питанням, є надання програмі, що реалізує виділення динамічної пам'яті, необхідного розміру фізичної пам'яті. Третім питанням є необхідність малої фрагментації фізичної пам'яті при роботі програми виділення динамічної пам'яті.

Ефективність роботи комп'ютерної системи залежить від багатьох параметрів і факторів. У випадку визначення ефективності роботи програми реалізації виділення динамічної пам'яті у комп'ютерній системі мають бути задіяні параметри, що визначені вище – час роботи, використовуваний об'єм пам'яті та параметри фрагментації.

У роботі [2] описуються параметри, які визначають час роботи алгоритмів виділення та звільнення пам'яті. У роботі [3] визначено необхідний об'єм пам'яті таких алгоритмів. У роботах [4, 5] наведено математичні вирази для визначення фрагментації пам'яті, а в роботі [6] наведено числові значення фрагментації цих алгоритмів.

Параметр і характеристика, що визначають в сукупності ці показники (час роботи, об'єм використовуваної пам'яті та значення фрагментації) для визначення ефективності роботи алгоритму виділення та звільнення динамічної пам'яті в комп'ютерних системах, є відкритим питанням.

Метою даної статті є визначення математичної залежності між показниками часу роботи, використовуваного об'єму пам'яті, значення фрагментації та ефективністю роботи алгоритму виділення та звільнення динамічної пам'яті.

Постановка задачі. Ефективність роботи комп'ютерної системи являє собою показник, який залежить від багатьох характеристик. У випадку визначення ефективності роботи алгоритму та програми розподілу динамічної пам'яті мають бути задіяні всі три показники, що вказані вище – час роботи програми, об'єм використовуваної пам'яті та значення фрагментації. Визначення математичної залежності між цими показниками та ефективністю необхідно виконати шляхом аналізу впливу кожної з цих характеристик на значення ефективності роботи комп'ютерної системи.

Рішення задачі. Рішення задачі виконаємо наступними кроками. На першому кроці проведемо аналіз кожної з цих характеристик та визначення їхнього впливу на ефективність роботи. На другому кроці сформуємо залежність, яка буде визначати математичний зв'язок між характеристиками часу роботи програми, об'єму використовуваної пам'яті і значення фрагментації та ефективністю роботи програми розподілу динамічної пам'яті.

1 крок. Відповідно до першого кроку, необхідно проаналізувати вплив характеристики «час роботи» на «ефективність роботи» програми динамічного виділення пам'яті та визначити чисельний показник, який буде його відображати. Зрозуміло, що при роботі програми даного типу головним показником, який визначає час роботи є кількість запитів програмних додатків на виділення/звільнення

динамічної пам'яті. Запити на виділення/звільнення динамічної пам'яті є головним та критичним часовим показником. Ефективність роботи комп'ютерної системи визначається результативністю роботи програмного забезпечення при застосуванні ресурсів комп'ютерної системи. Очевидно, що в тому випадку, коли кількість запитів на виділення/звільнення динамічної пам'яті буде зменшуватися, результативність роботи комп'ютерної системи та її ефективність буде підвищуватись. По цій причині, можна сформулювати висновок: чим менше кількість запитів і час роботи, тим більше буде ефективність алгоритму розподілення динамічної пам'яті в комп'ютерній системі.

Об'єм пам'яті, що необхідний для роботи програми розподілу динамічної пам'яті, визначається кількістю байт, які задіяні у фізичній пам'яті. Не зважаючи на те, що об'єм пам'яті комп'ютерної системи має тенденцію до збільшення, задіяний об'єм фізичної пам'яті певною програмою є одним з основних та головних показників, який впливає на результативність та ефективність роботи всієї комп'ютерної системи. У випадку, коли певний програмний додаток займає значний об'єм пам'яті, робота комп'ютерної системи буде сповільнюватися через неможливість в певний момент часу завантажити інший програмний додаток (особливо в багатозадачних системах). Якщо в операційній системі реалізовано сторінковий спосіб організації пам'яті і виникає необхідність завантажити в оперативну пам'ять програмний додаток, розмір якого перевищує вільний простір пам'яті, операційна система має виконувати вивантаження вільних сторінок пам'яті на жорсткий диск. Дана операція займає значний час. Аналізуючи це, можна зробити висновок, що ефективність роботи програмного додатку в комп'ютерній системі обернено пропорційна обсягу пам'яті, що його займає.

Фрагментація – це явище утворення у фізичній пам'яті невикористаних блоків, і ці блоки неможливо використовувати в подальшому при роботі системи. В роботах [4, 5] наведено класифікацію фрагментації та визначено їхню кількісну оцінку. Згідно цієї роботи, існує два типи фрагментації – зовнішня та внутрішня.

Зовнішня фрагментація – це поява між блоками виділеної фізичної пам'яті пустих ділянок пам'яті, які неможливо використати. Кількісний показник зовнішньої фрагментації визначається параметром «ступінь зовнішньої фрагментації» і задовольняє виразу:

$$S = \max\{x_i\}, \sum_{i=1}^m x_i > S, \quad (1)$$

де S – необхідний розмір пам'яті;
 $\{x_i\}$ ($i = 1, 2, \dots, m = N/2$) – послідовність розмірів вільних сегментів.

Внутрішня фрагментація – це утворення всередині блоків пам'яті вільних невикористаних ділянок. Внутрішня фрагментація виникає по причині округлення розміру запитуваної пам'яті. Для визначення впливу внутрішньої фрагментації на ефективність комп'ютерної системи необхідно проаналізувати причини її утворення та наслідки, які виникають після її формування. Згідно роботи [7], у тому випадку, коли необхідно виділити блок пам'яті B , розмір цього блоку має бути більшим або дорівнювати розміру вказаному в запиті на виділення. Нехай різниця між запитаним розміром та наданим блоком дорівнює Δ . Якщо числове значення Δ порівняно велике, то в системі має відбутися поєднання блоків, розмір яких сумірний з цим значенням. Ефективність роботи всієї системи може підвищитися в тому випадку, коли для запиту буде наданий блок, розмір якого відповідає запитаному розміру. Для того, щоб кількісно оцінити вплив блоків виділеної пам'яті з різним розміром невикористаного простору, необхідно розмежувати вільний простір в блоках на два типи: великий простір та малий простір. Розмежування необхідно виконати згідно параметру «граничне значення» – Δ_{\min} . В тому випадку, якщо $\Delta \geq \Delta_{\min}$, вільний простір блоку формується як новий вільний блок. В тому випадку, коли $\Delta < \Delta_{\min}$, нові динамічні блоки не утворюються.

Згідно робіт [4, 5], фрагментація визначається наступним виразом:

$$f_i = f_e + f_i, \quad (2)$$

де f_e – зовнішня фрагментація;
 f_i – внутрішня фрагментація.

В цьому виразі зовнішня фрагментація f_e визначається формулою 3.

$$f_e = \frac{NK_{cp}S_o}{2M_o}, \quad (3)$$

де N – кількість зайнятих сегментів;
 K_{cp} – коефіцієнт, що розраховується на основі правила «п'ятдесяти відсотків»;
 S_o – середній розмір зайнятого сегмента пам'яті;
 M_o – розмір розподіленої пам'яті в словах.
 Внутрішня фрагментація f_i визначається формулою:

$$f_i = \frac{N\Delta_{\min}}{2M_o}, \quad (4)$$

Розмір розподіленої пам'яті в словах M_o визначається виразом:

$$M_o = NS_o \left(1 + \frac{K_{cp}}{2}\right). \quad (5)$$

Кінцевий вираз фрагментації, яка має місце при виділенні динамічної пам'яті, визначається таким виразом:

$$f_1 = \frac{2K_{cp}S_o + S_oK_{cp}^2 + 2\Delta_{\min} + K_{cp}\Delta_{\min}}{4S_o + 4S_oK_{cp} + S_oK_{cp}^2}, \quad (6)$$

Явище фрагментації являє собою дуже серйозну проблему. Її існування має негативний характер по двом основним причинам. Перша – при виділенні/звільненні динамічної пам'яті, фрагментація сповільнює роботу всієї системи. Друга – створюються блоки пам'яті, які неможливо використати при подальшій роботі системи. З цих тверджень випливає, що кількісний показник, який її визначає, обернено пропорційний ефективності роботи всієї комп'ютерної системи.

Крок 2. Проаналізувавши вплив всіх трьох параметрів (час роботи програми, об'єм необхідної пам'яті та фрагментація), від яких залежить робота алгоритму розподілення динамічної пам'яті, визначимо вираз:

$$e = \frac{1}{\tau(N)Vf_1} = \frac{1}{\tau(N)V\left(\frac{2K_{cp}S_o + S_oK_{cp}^2 + 2\Delta_{\min} + K_{cp}\Delta_{\min}}{4S_o + 4S_oK_{cp} + S_oK_{cp}^2}\right)} = \frac{4S_o + 4S_oK_{cp} + S_oK_{cp}^2}{\tau(N)V(2K_{cp}S_o + S_oK_{cp}^2 + 2\Delta_{\min} + K_{cp}\Delta_{\min})}, \quad (7)$$

де $\tau(N)$ – час, що необхідний для роботи програми, с;
 V – об'єм пам'яті в байтах.

Висновки: В ході проведеного дослідження, згідно з поставленою задачею, був визначений показник ефективності роботи алгоритму розподілу динамічної пам'яті залежно від параметрів часу роботи програми, об'єму необхідної пам'яті та показник фрагментації. Показник ефективності був сформований шляхом аналізу впливу кожного параметру на ефективність роботи даного алгоритму. Всі три параметри (час необхідний для роботи програми, об'єм використовуваної пам'яті та показник фрагментації) обернено пропорційні показнику ефективності роботи програми розподілення динамічної пам'яті. Даний показник може бути застосований в якості визначення ефективності роботи різних алгоритмів розподілення пам'яті, які поширені в сучасних операційних системах. В якості подальшого дослідження, можливо визначити граничні значення даного показника для різних алгоритмів розподілення динамічної пам'яті.

Література

1. Yusuf Hasan, J. Morris Chang. A tunable hybrid memory allocator [Електронний ресурс]. – Режим доступу : www.home.engineering.iastate.edu/~morris%2Fpapers%2Fjss_tunable_yusuf_06.pdf
2. Maged M. Michael. Scalable Lock-Free Dynamic Memory Allocation [Електронний ресурс]. – Режим доступу : www.com%2Fdocs%2FClassics%2FLater_on%2FScalable%2520Lock-Free%2520Dynamic%2520Memory%2520Allocation.pdf
3. Stelios Joannou, Rajeev Raman. An empirical evaluation of extendible arrays [Електронний ресурс]. – Режим доступу : www.cs.le.ac.uk%2Fpeople%2Fsj148%2F092_joannou.pdf
4. Denning P. J. Origin of Virtual Machines and Other Virtualities / P. J. Denning // АНС. – 2001. – Vol. 23. – № 3. – 73 p.
5. Denning P. J. Virtual Memory / P. J. Denning // Computing Surveys. – 1970, Vol. 2. – № 3 – P. 153 – 189.
6. Mark S. Johnstone, Paul R. Wilson. The memory fragmentation problem: solved? [Електронний ресурс]. – Режим доступу : www.citeseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.97.5185%26rep%3Drep1%26type%3Dpdf
7. Синеглазов В. М. Оценивание системных издержек при динамическом сегментном распределении памяти / В. Синеглазов, А. Юрченко // ISSN 1990-5548 Электроника и системы управления. – К. : Национальный авиационный университет. – 2011. – № 2 (28). – С. 125–128

References

1. Yusuf Hasan, J. Morris Chang. "A tunable hybrid memory allocator." (02 27 2013). www.home.engineering.iastate.edu/~morris%2Fpapers%2Fjss_tunable_yusuf_06.pdf
2. Maged M. Michael. "Scalable Lock-Free Dynamic Memory Allocation." (02 27 2013). www.com%2Fdocs%2FClassics%2FLater_on%2FScalable%2520Lock-Free%2520Dynamic%2520Memory%2520Allocation.pdf
3. Stelios Joannou, Rajeev Raman. "An empirical evaluation of extendible arrays." (02 27 2013). www.cs.le.ac.uk%2Fpeople%2Fsj148%2F092_joannou.pdf
4. Denning P. J. "Origin of Virtual Machines and Other Virtualities.", Vol. 23(3), 2001, 73 p.
5. Denning P. J. "Virtual Memory", Computing Surveys, Vol. 2(3), 1970, p. 153 – 189.
6. Mark S. Johnstone, Paul R. Wilson. "The memory fragmentation problem: solved?", (02 27 2013). www.citeseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.97.5185%26rep%3Drep1%26type%3Dpdf
7. Sineglazov V.M. Evaluation of system overhead for dynamic memory allocation segment / Sineglazov V. M., Yurchenko A. S. // ISSN 1990-5548 Electronics and Control Systems – K.: National Aviation University. – 2011. – № 2 (28). – p. 125–128

Рецензія/Peer review : 18.5.2013 p.

Надрукована/Printed : 19.6.2013 p.
 Рецензент: д.т.н. В.Ю. Кучерук