

ვირტუალური მეხსიერების მართვის მოდელი

ლია გაჩეჩილაძე

სტუ-ს ასისტენტ-პროფესორი

რეზიუმე

სტატიაში შემოთავაზებულია ახალი მიდგომა ოპერაციული სისტემის ბირთვის მიერ შესრულებული ფუნქციების, კერძოდ კი ვირტუალური მეხსიერების მართვის პროცესების ვიზუალიზებისადმი. ვირტუალური მეხსიერების მართვის პროცესი ადამიანის თვალისთვის უხილავია, ამიტომ მისი ეფექტური სწავლების მიზნით საჭიროა შესაბამისი მიდგომებისა და მოდელების შემუშავება. შემუშავებული მოდელი ითვალისწინებს რამდენიმე დაშვების არსებობას: წინასწარ არის ცნობილი სისტემაში თითოეული გვერდის შემოსვლის მიმდევრობა და გვერდის მიერ მოთხოვნილი მეხსიერების ზომა. მოთხოვნის თითოეული ელემენტი შეიცავს გვერდის ნომერს, მისი გამოყენების რიგითობას და მის მიერ მოთხოვნილი ოპერაციული მეხსიერების ზომას. ვირტუალური მეხსიერების მართვის პროცესების ასეთი სახით წარმოდგენას საფუძვლად უდევს რიგების თეორია, რადგან მოთხოვნის შემოსვლა შემთხვევითი პროცესია და ემორჩილება მარკოვის პროცესს. ასეთი მიდგომა საშუალებას გვაძლევს შევიშუშოთ ვირტუალური მეხსიერების მართვის ვიზუალიზების ალგორითმები, გრაფიკული ინტერფეისი და შესაბამისი პროგრამული საწვრთნელი.

საკვანძო სიტყვები: მართვის სტრატეგიები, ვირტუალური მეხსიერების მართვა, ვიზუალიზების მოდელი.

VIRTUAL MEMORY MANAGEMENT MODEL

Lia Gachechiladze

SUMMARY

The article proposes a new approach for visualizing the functions performed by the operating system kernel, in particular, virtual memory management processes. The process of managing virtual memory is invisible for the human eye, so it is necessary to develop appropriate approaches and models for its effective learning. The developed model provides for the following assumptions: the sequence of entering the page in system and the amount of memory required by the page are known in advance. Each request element contains the page number, the sequence of its use and the requested amount of memory. Virtual memory management processes are based on queuing theory, since the request is a random process and is subject to the Markov process.

This approach allows us to create virtual memory management visualization algorithms, a graphical user interface and corresponding software simulator.

Keywords: management strategies, virtual memory management, visualization model.

* * * *

როგორც ცნობილია, ოპერაციული სისტემის მუშაობის დროს მისი ბირთვი მრავალ ფუნქციას ასრულებს. ერთ-ერთი მათგანია ვირტუალური მეხსიერების მართვა. პროგრამები შესრულების დროს ოპერაციულ სისტემას უგზავნიან მათი გვერდების ოპერაციულ მეხსიერებაში მოთავსების მოთხოვნებს. ეს მოთხოვნები იგზავნება გაცილებით სწრაფად, ვიდრე ხდება მათი მომსახურება, რის შედეგად ფორმირდება გვერდების მოთხოვნების რიგი. ამ რიგის მომსახურების ეფექტურობაზე მნიშვნელოვნად არის დამოკიდებული ოპერაციული სისტემის მუშაობის სისწრაფე.

ვირტუალური მეხსიერების მართვისთვის შემდეგი სტრატეგიები გამოიყენება [1-2]:

- შემოტანის სტრატეგიები. ისინი განსაზღვრავს, თუ როდის უნდა მოხდეს გვერდის ჩატვირთვა დისკიდან ოპერაციულ მეხსიერებაში. მოთხოვნის მიხედვით შემოტანის სტრატეგია ითვალისწინებს გვერდის ჩატვირთვას ოპერაციულ მეხსიერებაში მომუშავე პროცესის მხრიდან მოთხოვნის არსებობის შემთხვევაში. გვერდის შემოტანა დასწრებით გულისხმობს სისტემის მიერ ასეთი გვერდების წინასწარ განსაზღვრას და ოპერაციულ მეხსიერებაში ჩატვირთვას. თუ გვერდთან მიმართვის ალბათობა დიდია, მაშინ ის ოპერაციულ მეხსიერებაში ჩაიტვირთება მანამ, სანამ მასთან მოხდება მიმართვა.

- განლაგების სტრატეგიები. ისინი განსაზღვრავს, თუ ოპერაციული მეხსიერების რომელ უბანში უნდა ჩაიტვირთოს გვერდი. აქ გამოიყენება ცნობილი სტრატეგიები: ყველაზე შესაფერისი, პირველი შესაფერისი და ნაკლებად შესაფერისი.

- გამოტანის სტრატეგიები. ისინი განსაზღვრავს, თუ რომელი გვერდი უნდა იყოს გამოტანილი ოპერაციული მეხსიერებიდან იმისთვის, რომ გაათავისუფლოთ ადგილი შემოსული გვერდისთვის.

არსებობს გვერდების გამოგდების შემდეგი სტრატეგიები [1-2]:

- შემთხვევითი გვერდის გამოგდება. ამ დროს ოპერაციულ მეხსიერებაში მოთავსებული ყველა გვერ-

დი შეიძლება ამორჩეული იყოს თანაბარი ალბათობით.

- პირველი გამოიგდება პირველი შემოსული გვერდი (FIFO). ამ პრინციპის გამოყენების შემთხვევაში თითოეულ გვერდს ოპერატიულ მეხსიერებაში შემოსვლის მომენტში ენიჭება დროითი ჭდე. ოპერატიული მეხსიერებიდან შესრულება იმ გვერდის გამოტანა, რომელიც მეხსიერებაში სხვებზე დიდხანს იმყოფებოდა.

- პირველი გამოიგდება ყველაზე ხანგრძლივად გამოუყენებელი გვერდი (LRU). ამ სტრატეგიის თანახმად, ოპერატიული მეხსიერებიდან შესრულება იმ გვერდის გამოტანა, რომელიც არ გამოიყენებოდა ყველაზე დიდხანს.

- პირველი გამოიგდება ყველაზე იშვიათად გამოყენებული გვერდი (LFU). ამ სტრატეგიის გამოყენებისას ფასდება თითოეული გვერდის გამოყენების ინტენსივობა. გამოიგდება ის გვერდი, რომელიც ყველაზე ნაკლები ინტენსივობით იქნება გამოყენებული ან რომელთან მიმართებაში ყველაზე იშვიათია.

- პირველი გამოიგდება ბოლო პერიოდში გამოუყენებელი გვერდი (NUR). ამ სტრატეგიის არსი იმაში მდგომარეობს, რომ გვერდებთან, რომლებიც უკანასკნელ პერიოდში არ გამოიყენებოდა, არ იქნება მიმართვა უახლოეს მომავალშიც და შესაბამისად, შეიძლება მათი გამოტანა ოპერატიული მეხსიერებიდან.

იმისთვის, რომ მოვახდინოთ ვირტუალური მეხსიერების მართვის მოდელირება უნდა გავაკეთოთ შემდეგი დაშვებები [3-6]:

- ვიცით სისტემაში თითოეული გვერდის შემოსვლის რიგითობა, ანუ მისი გამოყენების რიგითობა.

- ვიცით ოპერაციული სისტემის მიერ გვერდისთვის გამოყოფილი მეხსიერების ზომა.

ამ დაშვებების გათვალისწინებით მომლოდინე გვერდები შეგვიძლია წარმოვადგინოთ შემდეგი მიმდევრობის სახით:

$$G_1(n_1, z_1); G_2(n_2, z_2); G_3(n_3, z_3) \dots G_i(n_i, z_i) \quad *$$

როგორც ვხედავთ, მიმდევრობის პირველი ელემენტი შეიცავს ინფორმაციას პირველი გვერდის შესახებ, მეორე ელემენტი - მეორე გვერდის შესახებ და ა.შ. მიმდევრობის ელემენტს აქვს სახე - $G_i(n_i, z_i)$. აქ G_i არის i -ური გვერდის ნომერი, n_i კი - G_i გვერდის გამოყენების რიგითობა. G_i გვერდი იკავებს ოპერატიული მეხსიერების z_i ზომის უბანს. განვიხილოთ მაგალითი. დავუშვათ გვაქვს გვერდების შემდეგი მიმდევრობა:

$$G_1(1, 8); G_2(2, 10); G_3(3, 6); G_4(4, 18); G_5(5, 20) \quad *$$

მიმდევრობის პირველ ელემენტში მოთავსებულია "G1(1,8)". ეს იმას ნიშნავს, რომ პირველად მიმართ-

ვა ხდება G1 გვერდთან. მისი ზომაა 8 მეგაბაიტი. G1 გვერდი ოპერატიულ მეხსიერებაში დაუყოვნებლივ მოთავსდება, თუ მეხსიერებაში არის თავისუფალი ადგილი. თუ მეხსიერებაში არ არის თავისუფალი ადგილი, მაშინ რომელიმე სტრატეგიის მიხედვით შესრულება ერთ-ერთი გვერდი გამოტანა მეხსიერებიდან და მის მაგივრად პირველი გვერდის ჩატვირთვა. რიგის მეორე ელემენტი შეიცავს G_2 გვერდს, რომელთანაც მეორე რიგში ხდება მიმართვა. G_2 გვერდი მეხსიერებაში იკავებს 10 მეგაბაიტი ზომის უბანს. თუ მეხსიერებაში არ არის თავისუფალი ადგილი, მაშინ რომელიმე სტრატეგიის მიხედვით შესრულება ერთ-ერთი გვერდი გამოტანა მეხსიერებიდან და მის ნაცვლად მეორე გვერდის ჩატვირთვა. როგორც რიგიდან ჩანს, ამავე გვერდთან მიმართვა შესრულება მესამე გვერდთან მიმართვის შემდეგ. ბოლოს მიმართვა მოხდება მეხუთე გვერდთან.

გვერდებთან მიმართვის ასეთი სახით წარმოდგენა შეგვიძლია განვიხილოთ როგორც გვერდის გამოტანისა და მისთვის მეხსიერების საჭირო ზომის უბნის გათავისუფლების გამარტივებული მარკოვის პროცესი. ის საშუალებას გვაძლევს შევიმუშავოთ მეხსიერებიდან გვერდის გამოტანის ვიზუალიზების ალგორითმები.

შემოვიტანოთ აღნიშვნები. i -ური გვერდი G_i -ით, გვერდების მაქსიმალური რაოდენობა $N_{\max}$ -ით, გვერდის მიერ დაკავებული მეხსიერების უბნის სანქისი მისამართი A_s -ით, ხოლო გვერდის მიერ დაკავებული მეხსიერების უბნის ბოლო მისამართი კი - A_b -ით აღვნიშნოთ. შესაბამისად, G_i გვერდის მიერ დაკავებული მეხსიერების უბნის სანქისი მისამართი იქნება A_{si} , ბოლო მისამართი კი - A_{bi} . G_2 გვერდის მიერ დაკავებული მეხსიერების უბნის სანქისი მისამართი იქნება A_{s2} , ბოლო მისამართი კი - A_{b2} და ა.შ. G_i გვერდის მიერ დაკავებული მეხსიერების უბნის საბოლოო მისამართი განისაზღვრება შემდეგნაირად: $A_{bi} = A_{si} + z_i$.

მეხსიერებიდან გვერდების გამოტანის სტრატეგიების მოდელირება შესაძლებელია შემდეგნაირად.

სტრატეგია „შემთხვევითი გვერდის გამოტანა“. G_i გვერდისთვის გამოიყოფა მეხსიერების თავისუფალი უბანი, რომელშიც ის მოთავსდება. თუ თავისუფალი უბანი არ არის მეხსიერებაში, მაშინ შესრულება შემთხვევითი გზით არჩეული G_j გვერდის გამოტანა და მის ნაცვლად მოთხოვნილი G_i გვერდის ჩატვირთვა.

სტრატეგია „პირველი შემოსული გვერდის გამოტანა“. G_i გვერდისთვის გამოიყოფა მეხსიერების თავისუფალი უბანი, რომელშიც ის მოთავსდება. თუ თავისუფალი უბანი არ არის მეხსიერებაში, მაშინ შესრულება არსებულ გვერდებს შორის პირველი შემოსული G_j გვერდის გამოტანა და მის ნაცვლად მოთხოვნილი G_i გვერდის ჩატვირთვა.

სტრატეგია „ყველაზე ხანგრძლივად გამოუყენებელი გვერდის გამოტანა“. ამ შემთხვევაში თითოეულ

გვერდს აქვს დროის მთვლელები t_i , რომელიც აფიქსირებს გვერდის გამოყენებიდან გასულ დროს. G_i გვერდისთვის გამოიყოფა მესხიერების თავისუფალი უბანი, რომელშიც ის მოთავსდება. თუ თავისუფალი უბანი არ არის მესხიერებაში, მაშინ შესრულება ყველაზე ხანგრძლივად გამოუყენებელი G_j გვერდის გამოტანა და მის ნაცვლად მოთხოვნილი G_i გვერდის ჩატვირთვა. ყველაზე ხანგრძლივად გამოუყენებელია ის გვერდი, რომლისთვისაც დროის მთვლელს აქვს უდიდესი მნიშვნელობა:

$$t_{\max} = \max \{ t_1, t_2, t_3, \dots, t_j \}.$$

სტრატეგია „იშვიათად გამოყენებული გვერდის გამოტანა“. ამ შემთხვევაში, თითოეულ გვერდს აქვს მისი გამოყენების მთვლელები M_i , რომელიც აფიქსირებს გვერდის გამოყენების რაოდენობას. G_i გვერდისთვის გამოიყოფა მესხიერების თავისუფალი უბანი, რომელშიც ის მოთავსდება. თუ თავისუფალი უბანი არ არის მესხიერებაში, მაშინ შესრულება იშვიათად გამოყენებული G_j გვერდის და მის ნაცვლად მოთხოვნილი G_i გვერდის ჩატვირთვა. იშვიათად გამოყენებულია ის გვერდი, რომლისთვისაც გამოყენების მთვლელს აქვს უმცირესი მნიშვნელობა:

$$M_{\min} = \min \{ M_1, M_2, M_3, \dots, M_j \}.$$

სტრატეგია „ბოლო ხანებში გამოუყენებელი გვერდის გამოტანა“. ამ შემთხვევაში თითოეულ გვერდს აქვს დროის მთვლელები t_i , რომელიც აფიქსირებს უკანასკნელ T პერიოდში გვერდის გამოყენების რაოდენობას. G_i გვერდისთვის გამოიყოფა მესხიერების თავისუფალი უბანი, რომელშიც ის მოთავსდება. თუ თავისუფალი უბანი არ არის მესხიერებაში, მაშინ შესრულება ბოლო ხანებში გამოუყენებელი G_j გვერდის გამოტანა და მის ნაცვლად

მოთხოვნილი G_i გვერდის ჩატვირთვა. ბოლო ხანებში გამოუყენებელია ის გვერდი, რომლისთვისაც უკანასკნელ T პერიოდში გვერდის გამოყენების რაოდენობას აქვს უმცირესი მნიშვნელობა:

$$B_{\min} = \min \{ B_1, B_2, B_3, \dots, B_j \}.$$

ასეთი მოდგომა საშუალებას გვაძლევს შევიშუაოთ ვირტუალური მესხიერების მართვის ვიზუალიზების ალგორითმები, გრაფიკული ინტერფეისი და პროგრამული სანვრთნელი.

გამოყენებული ლიტერატურა

1. Дейтел. Г. Введение в операционные системы: В 2-х т. Т. 1. Пер. с англ. - М.: Мир, 1987. - 359 с., ил.
2. Таненбаум Э. Современные операционные системы. - СПб.: Питер, 2002. - 1040 с., ил. 1985. - 176 с.
3. Самхарадзе Р., Кевхიшвили М., Гачечиладзе Л. О вопросе построения интеллектуальной оболочки дистанционного обучения и тестирования. XIV Международный Симпозиум Управление Большими Системами. Труды Симпозиума - Информатизация, Автоматизация, Управление. 2000.
4. რ. სამხარაძე. პეტრის ქსელები კომპიუტერულ სწავლებაში (მონოგრაფია). თბილისი, „ტექნიკური უნივერსიტეტი“, 2007 ISBN 978-99940-57-93-1. 156 გვ.
5. რ. სამხარაძე, ლ. გაჩეჩილაძე. იმიტაციური მოდელები კომპიუტერულ სწავლებაში (მონოგრაფია). თბილისი, „ტექნიკური უნივერსიტეტი“. 2008. ISBN 978-9941-14-057-0. 124 გვ.
6. რ. სამხარაძე, ლ. გაჩეჩილაძე, მ. ქურდაძე. რიგების თეორიის გამოყენება კომპიუტერულ სწავლებაში (მონოგრაფია). თბილისი, სტუ-ს „IT-კონსალტინგის სამეცნიერო ცენტრი“, 2017. ISBN 978-9941-0-9643-3. 93 გვ.