

## VN-ქსელი

რომან სამხარაძე  
საქართველოს ტექნიკური უნივერსიტეტი

### რეზიუმე

სტატიაში შემოთავაზებულია ახალი მიდგომა პროცესების მდგომარეობათა ცვლილებისა და პროცესორების დატვირთვის დაგეგმვის ვიზუალიზებისადმი. შემუშავებულია პეტრის ახალი ტიპის – VN-ქსელი (Visual Net), რომელიც იძლევა ოპერაციული სისტემის მუშაობის დროს კომპიუტერში მიმდინარე პროცესების ვიზუალიზების შესაძლებლობას. VN-ქსელის ბაზაზე შემუშავებულია პროცესების მდგომარეობათა ცვლილებისა და პროცესორების დატვირთვის დაგეგმვის ვიზუალიზების მოდელები, ალგორითმები და სასწავლო პროგრამული ტრენაჟორები.

**საკვანძო სიტყვები:** პეტრის ქსელი. სიმრავლე. პროცესი. პროცესის მდგომარეობა. ბლოკირებული მდგომარეობა. ოპერაციული სისტემა. პროცესორის დატვირთვა. დროის კვანტი.

### 1. შესავალი

როგორც ცნობილია, ოპერაციული სისტემის ბირთვი მრავალ ფუნქციას ასრულებს. სტუდენტებისათვის მათი სათანადო დონეზე სწავლება აქტუალური საკითხია. ამ ფუნქციებიდან სტატიაში განხილულია პროცესების მდგომარეობათა ცვლილება და პროცესორების დატვირთვის დაგეგმვა. მათი ვიზუალიზება და მონიტორის ეკრანზე გამოტანა ზრდის სწავლების ხარისხს და აადვილებს შესასწავლი საკითხების კარგად გააზრებას. ამ დროს, მთავარი პრობლემაა სტუდენტის მიერ კომპიუტერში მიმდინარე პროცესების წარმოდგენისა და აღქმის სირთულე. პრობლემის გადაჭრა შესაძლებელია ოპერაციული სისტემის ბირთვის მიერ შესრულებული ფუნქციების ვიზუალიზების გზით.

ამ მიმართულებით აუცილებელია თანამედროვე კომპიუტერული ტექნოლოგიების გამოყენება, რომლებიც წარმოადგენს საგანმანათლებლო ტექნოლოგიების ერთ-ერთ მთავარ შემადგენელ ნაწილს. სასწავლო პროცესში კომპიუტერის ჩართვა დიდ და ფართო შესაძლებლობებს იძლევა ოპერაციული სისტემის მუშაობის ვიზუალიზების საქმეში. კერძოდ, შესაძლებელია პროგრამული სასწავლო ტრენაჟორების შემუშავება და სასწავლო პროცესში მათი დანერგვა.

### 2. ძირითადი ნაწილი

**განსაზღვრება.** (Visual Net) VN-ქსელი არის სამეული:

$$VN = (R, D, F)$$

სადაც R - საინფორმაციო პოზიციების სიმრავლეა,  $R = \{r_1, r_2, r_3, r_4, r_5\}$ ; D -

გადასასვლელების სიმრავლეა,  $D = \{d_1, d_2, d_3, d_4\}$ ;  $F$ - გადასასვლელებისათვის მიწერილი ფუნქციების სიმრავლე,  
 $F = \{f_1, f_2, f_3, f_4\}$ .

თითოეულ პროცესს შეესაბამება  $m$  მარკერი,  $m \in M$ . თითოეულ მარკერს თან ახლავს ატრიბუტები. თითოეული წყვილის პირველი ელემენტია პროცესის მდგომარეობა: აქტიური ან ბლოკირებული; მეორე ელემენტია დროის ერთეული, რომლის განმავლობაშიც გრძელდება პროცესის აქტიური ან ბლოკირებული მდგომარეობა. ეს ატრიბუტები წარმოადგენს გადასასვლელების ფუნქციების არგუმენტებს. თითოეულ  $f$  ფუნქციას აქვს ხუთი არგუმენტი:

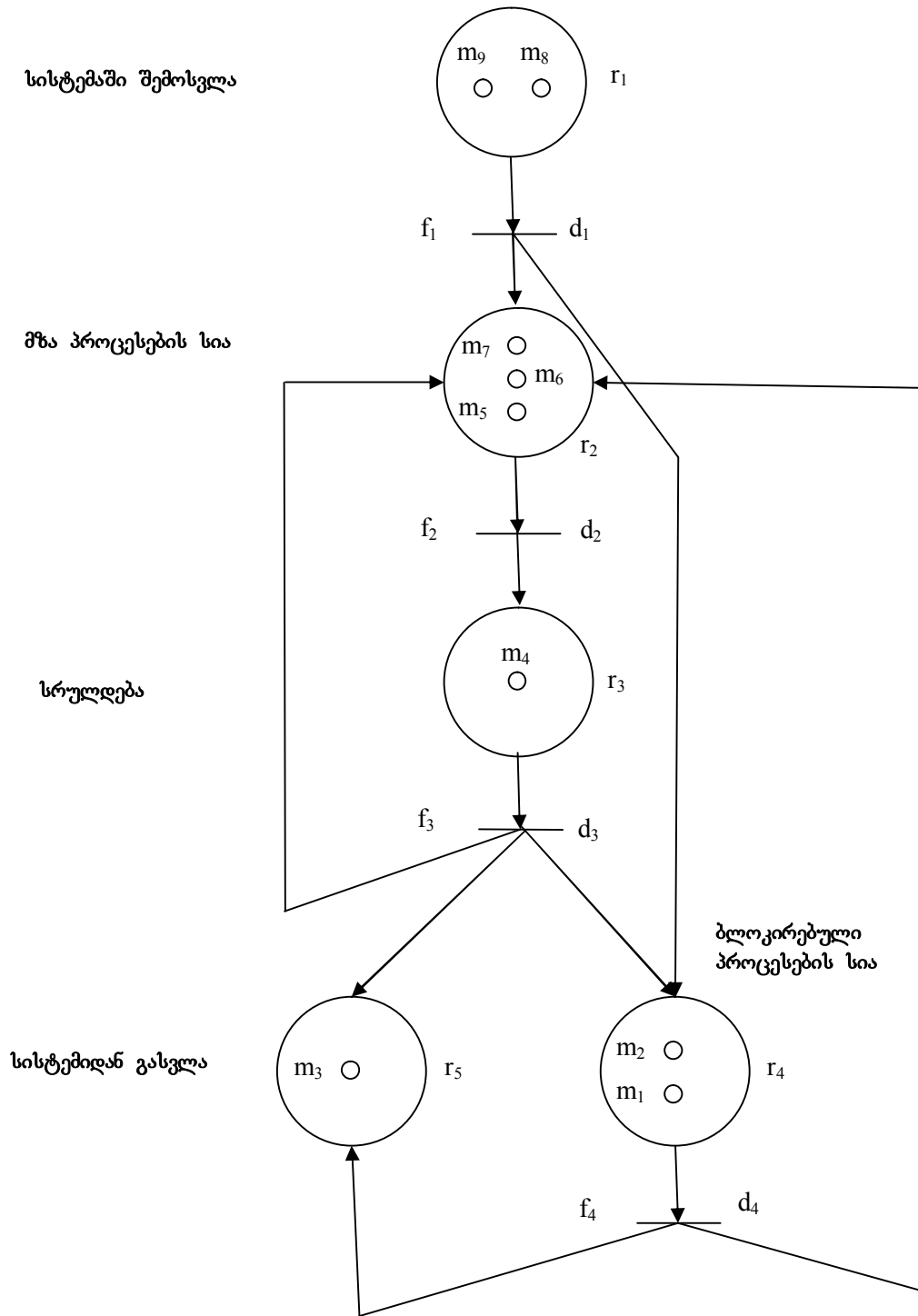
**$f$ (პროცესის იდენტიფიკატორი, მდგომარეობა, ხანგრძლივობა, პრიორიტეტი, მონიშნული ელემენტი)**

აქ პროცესის იდენტიფიკატორი არის პროცესის სახელი ან აღნიშვნა, მაგალითად,  $P_1$ ; მდგომარეობა არის პროცესის აქტიური ან ბლოკირებული მდგომარეობა, 'ა' ან 'ბ'; ხანგრძლივობა არის პროცესის აქტიურობის ან ბლოკირების დრო,  $t_a$  ან  $t_b$ ; პრიორიტეტი არის პროცესის პრიორიტეტი, მაგალითად  $\Pi_1$ ; მონიშნული ელემენტი არის მიმდევრობის მომდევნო ელემენტი.

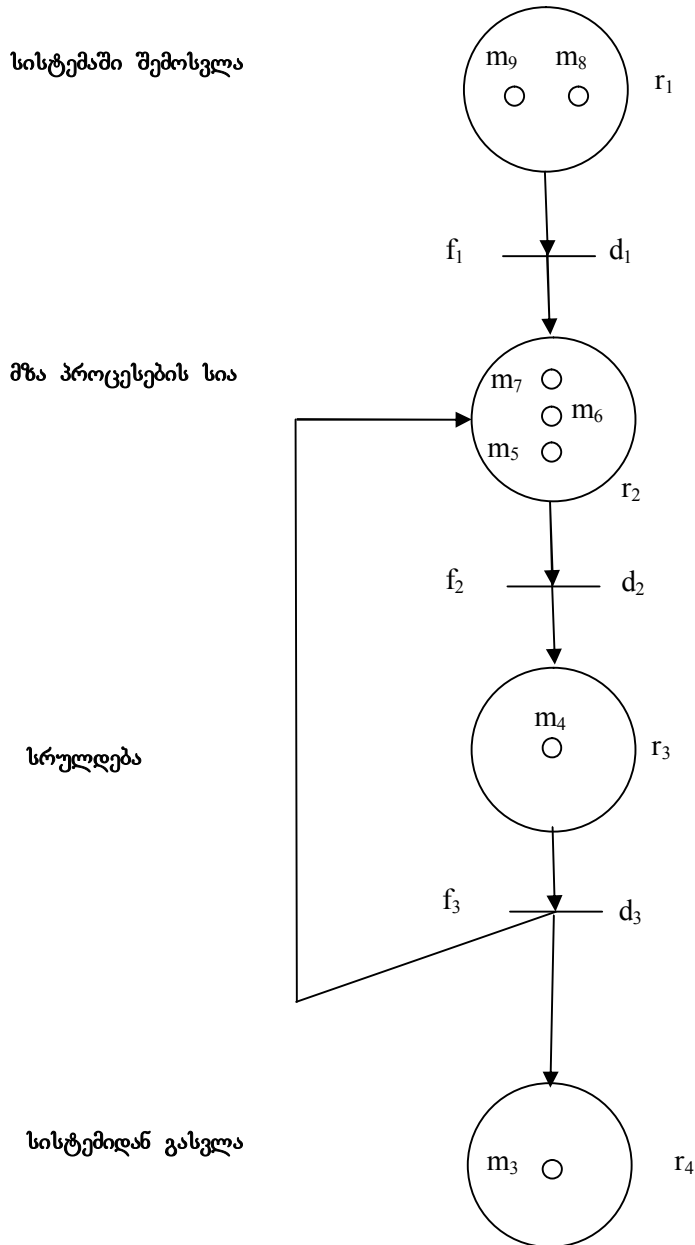
VN-ქსელს პროცესების მდგომარეობათა ცვლილების მოდელირების დროს აქვს 1-ელ ნახაზზე ნაჩვენები სახე, ხოლო პროცესორების დატვირთვის დაგეგმვის მოდელირების დროს - მე-2 ნახაზზე.

VN-ქსელი შეიცავს ხუთი ტიპის საინფორმაციო პოზიციას. პირველი ტიპის პოზიციაში ( $r_1$ ) მოთავსებულია სისტემაში შემოსული პროცესების შესაბამისი მარკერები. მეორე ტიპის პოზიციაში ( $r_2$ ) მოთავსებულია შესასრულებლად მზადყოფი პროცესების შესაბამისი მარკერები. ამ პოზიციაში მარკერების მოწესრიგება ხდება FIFO-პრინციპით, თუ პროცესებს თანაბარი პრიორიტეტები აქვს, ან პრიორიტეტების მიხედვით, თუ ისინი განსხვავებული პრიორიტეტებთაა. აქვთ. მესამე ტიპის პოზიციაში ( $r_3$ ) მოთავსებულია აქტიური ანუ შესრულების სტადიაში მყოფი პროცესის შესაბამისი მარკერი. მეოთხე ტიპის პოზიციაში ( $r_4$ ) მოთავსებულია ბლოკირებული პროცესების შესაბამისი მარკერები. ამ პოზიციაში მარკერების მოწესრიგება ყოველთვის ხდება FIFO პრინციპით. მეხუთე ტიპის პოზიციაში ( $r_5$ ) მოთავსებულია სისტემიდან გასული პროცესების შესაბამისი მარკერები.

VN-ქსელი შეიცავს ოთხი ტიპის გადასასვლელს. პირველი ტიპის გადასასვლელზე ( $d_1$ ) მუშაობს  $f_1$  ფუნქცია, რომელიც  $r_2$  პოზიციაში მარკერებს FIFO პრინციპის მიხედვით ათავსებს, თუ მარკერებს ერთნაირი პრიორიტეტები აქვს, ან ათავსებს პრიორიტეტის მიხედვით, თუ მარკერებს არათანაბარი პრიორიტეტები აქვს ამ და ყველა დანარჩენ გადასასვლელზე მომუშავე ფუნქციას ზემოთ აღნიშნული ატრიბუტების გარდა არგუმენტებად გადაეცემა, აგრეთვე, მონიშნული წყვილები და პროცესების პრიორიტეტები. ფუნქცია ამოწმებს თითოეული მარკერის პირველი წყვილის პირველ ელემენტს. მასზე დამოკიდებულებით მარკერი გადაადგილდება  $r_2$  პოზიციისაკენ, ანუ დგება მზა პროცესების სიაში, ან  $r_4$  პოზიციისაკენ, ანუ დგება ბლოკირებული პროცესების სიაში.



ნახ. 1.



ნახ. 2.

მეორე ტიპის გადასასვლელზე ( $d_2$ ) მუშაობს  $f_2$  ფუნქცია, რომელიც  $r_2$  პოზიციიდან ანუ მზა პროცესების სიიდან რიგით პირველი პროცესის შესაბამისი მარკერის ამორჩევას და  $r_3$  პოზიციაში მოთავსებას იმ შემთხვევაში, როცა  $r_3$  პოზიცია ცარიელია, ანუ პროცესორი თავისუფალია. ფუნქცია ამოწმებს  $T_{კვ}$  კვანტის მნიშვნელობას. თუ  $T_{კვ} = 0$ , მაშინ რიგით პირველი მარკერი გადაადგილდება  $r_3$  პოზიციისაკენ.

მესამე ტიპის გადასასვლელზე ( $d_3$ ) მუშაობს  $f_3$  ფუნქცია, რომელიც ახდენს დროის  $T_{კვ}$  კვანტს ერთი ერთეულით ამცირებს. კვანტის ამოწურვისთანავე ატრიბუტებზე დამოკიდებულებით მარკერი მოთავსდება ან  $r_5$  პოზიციაში ანუ გადის სისტემიდან, ან  $r_4$  პოზიციაში ანუ გადადის

ბლოკირებულ მდგომარეობაში, ან  $r_2$  პოზიციაში ანუ დგება მზა პროცესების სიაში პრიორიტეტის მიხედვით. თუ  $T_{კვ} = 0$  და  $T_{P_{1k}} \neq 0$ , მაშინ მარკერი გადაადგილდება  $r_2$  პოზიციისაკენ. თუ  $T_{კვ} = 0$  და  $T_{P_{1k}} = 0$ , მაშინ მარკერი გადაადგილდება  $r_4$  პოზიციისაკენ. თუ  $T_{კვ} \neq 0$  და  $T_{P_{1k}} = 0$ , მაშინ მარკერი გადაადგილდება  $r_4$  პოზიციისაკენ. თუ  $T_{კვ} \neq 0$  და  $T_{P_{1k}} \neq 0$ , მაშინ მარკერი რჩება  $r_3$  პოზიციაში. ფუნქცია ამოწმებს აგრეთვე, მონიშნულია თუ არა უკანასკნელი წვეილი. თუ მონიშნულია, მაშინ მარკერი  $r_3$  პოზიციიდან გადაადგილდება  $r_5$  პოზიციისაკენ.

მარკერის  $r_3$  ან  $r_4$  პოზიციაში მოთავსების შემდეგ მონიშნება მიმდევრობის მომდევნო წვეილი. თუ მონიშნა მიმდევრობის უკანასკნელი წვეილი, ანუ “\*\*”, მაშინ მარკერი გადაადგილდება  $r_5$  პოზიციისაკენ, ანუ პროცესი სისტემიდან გადის.

მეოთხე ტიპის გადასასვლელზე ( $d_4$ ) მუშაობს  $f_4$  ფუნქცია, რომელიც ამოწმებს მონიშნული წვეილის მეორე ელემენტს. თუ ის განულდა, მაშინ მარკერი გადაადგილდება  $r_2$  პოზიციისაკენ და მონიშნება მიმდევრობის მომდევნო წვეილი. თუ მონიშნული წვეილი უკანასკნელია, მაშინ მარკერი  $r_4$  პოზიციიდან გადაადგილდება  $r_5$  პოზიციისაკენ. შევნიშნოთ, რომ  $f_3$  და  $f_4$  ფუნქციები პარალელურად მუშაობს იმ შემთხვევაში, როცა  $r_3$  და  $r_4$  პოზიციები მარკერებს შეიცავს.

რაც შეეხება მარკერებს, ისინი ორი ტიპისაა. პირველი ტიპის მარკერები შეესაბამება იმ პროცესებს, რომლებიც მუშაობას იწყებს აქტიური მდგომარეობით. მეორე ტიპის მარკერები შეესაბამება იმ პროცესებს, რომლებიც მუშაობას იწყებს ბლოკირების მდგომარეობით.

პროცესორის დატვირთვის დაგეგმვის მოდელირებისათვის ხდება VN-ქსელის გადაწყობა, რაც შემდეგში მდგომარეობს (ნახ.2). ქსელში არ გვექნება ბლოკირებული პროცესების შესაბამისი წვერი ( $r_4$ ) და შესაბამისი გადასასვლელი ( $d_4$ ).  $f_1$  ფუნქცია  $r_2$  პოზიციაში მარკერებს მოაწესრიგებს პროცესორების დაგეგმვის დისციპლინის მიხედვით, პრიორიტეტების გათვალისწინებით. მარკერები ერთი ტიპისაა. თითოეულ მათგანს თან ახლავს ატრიბუტები: პროცესის ლოდინის დრო; დროის ერთეული, რომლის განმავლობაშიც გრძელდება პროცესის აქტიური დრო; პროცესის პრიორიტეტი.

VN-ქსელი მიეკუთვნება როგორც პეტრის ფერად ქსელებს, ისე FIFO-ქსელებს. რადგან სისტემაში სრულდება განსხვავებული პროცესები, ამიტომ მათი შესაბამისი მარკერებიც იქნება განსხვავებული, ანუ სხვადასხვა ფერის. FIFO-ქსელებს მიეკუთვნება იმიტომ, რომ  $d_2$  გადასასვლელის  $f_2$  ფუნქცია  $r_2$  პოზიციაში მარკერების რიგს აწესრიგებს FIFO-წყისის მიხედვით.

### **3. დასკვნითი ნაწილი**

კომპიუტერული პედაგოგიკა თანამედროვე პედაგოგიკის ერთ-ერთი ძირითადი მიმართულებაა. თანამედროვე საინფორმაციო ტექნოლოგიების გამოყენებას პედაგოგიკაში ახალ, თანამედროვე დონეზე აჰყავს სასწავლო პროცესი. სტატიაში შემოთავაზებული მიდგომა ამ მიმართულებით გადადგმული ნაბიჯია. შემუშავებული პეტრის ახალი ტიპის ქსელი – VN-ქსელი, იძლევა ოპერაციული სისტემის მიერ შესრულებული რიგი ფუნქციების ვიზუალიზების

შესაძლებლობას. VN-ქსელის ბაზაზე შემუშავებულია პროცესების მდგომარეობათა ცვლილებისა და პროცესორების დატვირთვის დაგეგმვის ვიზუალიზების მოდელები, ალგორითმები და სასწავლო პროგრამული ტრენაჟორები. შედეგად, სტუდენტი ეკრანზე ხედავს კომპიუტერში მიმდინარე პროცესებს, ცვლის პარამეტრებს, აკვირდება მიღებულ შედეგებს და აქტიურად მონაწილეობს აღნიშნული საკითხების შესწავლის პროცესში.

#### **ლიტერატურა**

1. Дейтел. Г. Введение в операционные системы: В 2-х т. Т. 1. Пер. с англ. – М.: Мир, 1987.
2. Дейтел. Г. Введение в операционные системы: В 2-х т. Т. 2. Пер. с англ. – М.: Мир, 1987.
3. ნანობაშვილი ქ., ზივზივაძე მ. მათემატიკის ზოგიერთი საკითხის კომპიუტერული სწავლების მეთოდთა ელემენტარულ სკოლაში. ქუთაისი. 2005.
4. Питерсон Дж. теория сетей Петри и моделирование систем. – М.: Мир, 1984.
5. Каплан Б.С. и др. Методы обучения математике. – Мн.: Нар. асвета, 1981.

#### **VN-СЕТЬ**

Самхарадзе Р.

Грузинский Технический университет

#### **Резюме**

В статье предложен новый подход к визуализации изменения состояний процессов и планирования нагрузки процессоров. Разработан новый тип сети Петри – VN-сеть (Visual Net), которая позволяет визуализировать процессы, протекающие в компьютере при работе операционной системы. На основе VN-сети разработаны модели и алгоритмы визуализации изменения состояний процессов и планирования нагрузки процессоров, а также соответствующие учебные программные тренажеры.

#### **VN-NET**

Roman Samkharadze

Georgian Technical University

#### **Summary**

In clause the new approach to visualization of change of conditions of processes and planning of loading of processors is offered. The new type of network Petri - a VN-network which allows visualization the processes proceeding in a computer at work of operational system is developed. On the basis of a VN-network models and algorithms of visualization of change of conditions of processes and planning of loading of processors, and also corresponding educational program simulators are developed.