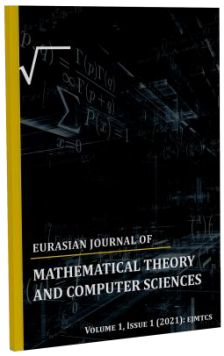
EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
UIF = 8.3 | SJIF = 5.916 www.in-academy.uz



OQIMLI PARALLEL DASTURLASHNING AFZALLIKLARI

Narziyev U.Z. [1] Turayev I.B. [2]
[1] BuxMTI AKT kafedrasi katta o‘qituvchisi,
[2] BuxMTI magistranti.

https://doi.org/10.5281/zenodo.6622512

ARTICLE INFO

Received: 28th May 2022
Accepted: 02nd June 2022
Online: 05th June 2022

KEY WORDS

dasturlash, oqimli parallel dasturlash, jarayonlar, ichki jarayonlar, ko‘p oqimli parallel dasturlash, resurslar, ma’lumotlar to‘plami, dastur ko‘rsatmalari, protsessor kvantlari, xotira katagi

ABSTRACT

Murakkab va protsessorning vaqtini ko‘p talab qiluvchi jarayonlarni dasturlash, jarayonlarni boshqarish, protsessor oqimlari va yadrolari imkoniyatlaridan to‘liq foydalanish jarayoni va oqimli parallel dasturlarni tuzish usullari talqin qilinadi.

Dasturlar protseduralardan tashkil topgan bo‘lib, jarayonlar deb ham ataladi. Oqimli dasturlash yordamida bir necha jarayonlarni ishlashini parallel amalga oshirish mumkin. Buning uchun kompyuterning mikropropsessori ko‘p yadroli, ko‘p oqimli bo‘lishi talab etiladi. Mabodo mikroprotssessor bitta yagona yadroga va oqimga ega bo‘lsa, bajariladigan amallarning ketma-ketligi tartiblanadi, ular navbatma-navbat amalga oshiriladi. Bu esa protsessorning imkoniyatlaridan parallel foydalanishga imkonyat bermaydi.

Bugungi kunda zamonaviy kompyuterlar asosida parallel oqimli dasturlardan audio, video, foto montaj ishlarini amalga oshirishda keng qo‘llanilmoqda. Misol tariqasida, Adobe Audition 2020 dasturida audio qayta ishlanganda, bir ishchi maydondagi audio ustida qandaydir tozalash, audio effekt qo‘shish, xotiraga saqlash, audioning formatini o‘zgartirish kabi amallarni bajarish vazifasi dasturda yuklansa, ikkinchi audio ma’lumotga o‘tib, ya’ni ishchi maydonda boshqa audio ma’lumotni ochib, uning ustida istalgan boshqa amallardan birini bajarishni dasturga yuklash mumkin. Dasturiy ta’minot parallel shaklda ikkala audio ma’lumot ustida boshqa-boshqa vazifalarlarni qotib qolmasdan, protsessorning imkoniyatlaridan to‘laqonli va samarali foydalangan holdan yuklangan vazifalarni amalga oshirib beradi. Hattoki uchinchi adio faylni ochib tegishli amalni bajarish ham mumkin.

Oqimli parallel dasturlash orqali protsessorning istalgan bo‘sh turgan yadrosiga, yoki oqimi(поток)ga amal bajarishni yuklash mumkin. “Operatsion

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
UIF = 8.3 | SJIF = 5.916 www.in-academy.uz

tizim ushbu vazifani o‘zi taqsimlash kerak emasmi?” - degan tabiiy savol paydo bo‘ladi. Yana “Operatsion tizim qaysi yadro yoki oqim(поток) joriy(ayni) vaqtda bo‘sh turgan bo‘lsa, amal bajarishni unga yuklasa, belgilangan vazifaning bajarilishi tezroq bo‘lmaydimi?” kabi savolga ham duch kelish mumkin. Shu vazifani operatsion tizimga yuklab, mikroprotsessorning aniq bo‘sh turgan yadrosiga yoki oqimiga yuklanganini taxmin qilib turmasdan, dasturning aynan murakkabroq qismini, parallel amallar bajarilishi lozim bo‘lgan qismini protssessorga oqimli parallel dasturlash orqali jo‘natish kompyuterning samaradorligini oshiradi. Oqimli parallel dasturlashda protsessosrning joriy vaqtda bo‘sh turgan oqimiga murojaat etiladi. Shu sababli, dasturning osilib qolishi (зависание) mumkin bo‘lgan qismi alohida, bo‘sh turgan protsessorning oqimi orqali parallel bajariladi va amallarni bajarish samaradorligi oshiriladi.

Oqimli parallel dasturlash qo‘llanilgan va bunday murakkab hisoblashlar ega bo‘lgan masala oqimli bo‘lmagan usul bilan hal etilishiga misol ko‘rib chiqsak. Delphi dasturlash muhitida ikkita loyiha hosil qilamiz, biri oqimli parallel dasturga misol bo‘lsin, ikkinchisi esa odatiy loyiha bo‘lsin. Murakkab hisoblashlarni talab etadigan va kompyuter protsessori ish vaqtini ko‘p talab etiluvchi misollar ko‘p, shu sababli biz misol tariqasida alohida protsedura(jarayon)ga ikkita o‘zgaruvchilarga taxminiy sonlarni generatsiya qilib oluvchi, cheksiz davom etuvchi jarayonni tashkil etsak. Cheksiz davom etish vazifasi taxminiy tanlanayotgan qiymatlarning ichidan taxminan bittasini ma’lum bir vaqt oralig‘ida olishni maqsad qilib belgiladik. Protseduraning kod ko‘rinishi quyidagicha:

```
procedure TMyThread.Execute;
begin
      Randomize;
       while True do
       begin
         a:=Random(99)-Random(99);
         b:=Random*100-Random(99);
       end;
end;
```

Ushbu oqimli parallel loyihamiz uchun quyidagicha obyektlar ishlatildi:

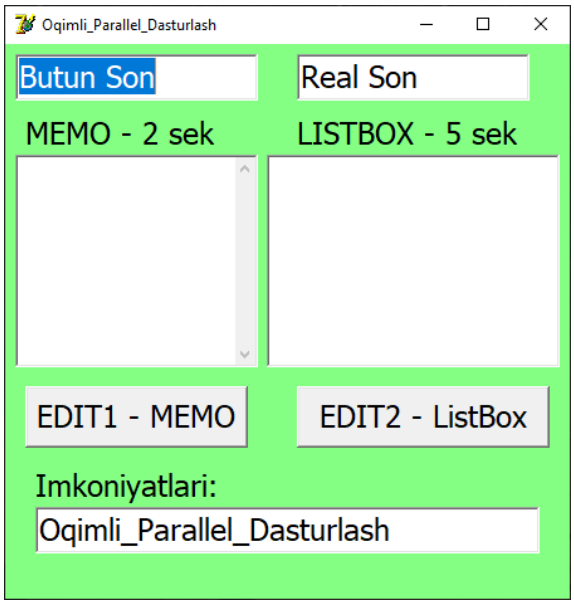
```
    type
      TForm1 = class(TForm)
        Edit1: TEdit;
        Edit2: TEdit;
        Memo1: TMemo;
        ListBox1: TListBox;
        Button1: TButton;
        Button2: TButton;
        Timer1:  TTimer;
        Timer2:  TTimer;
```

EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
UIF = 8.3 | SJIF = 5.916 www.in-academy.uz

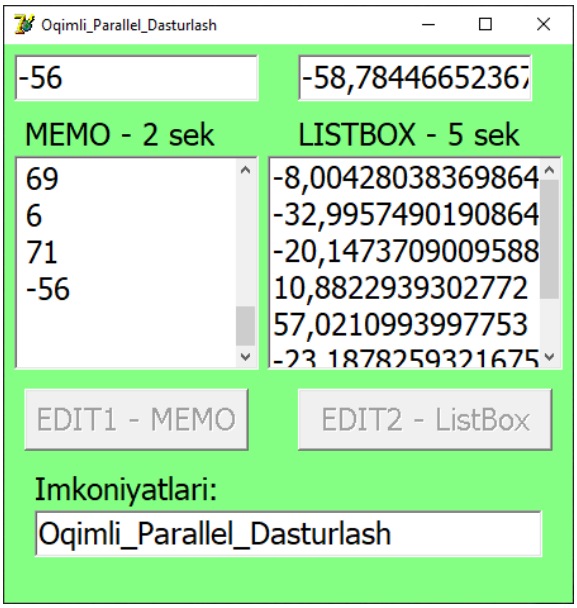
```
    Label1:  TLabel;
    Label2:  TLabel;

  TMyThread = class(TThread)
    private
      {Private}
    protected
      procedure Execute; override;
  end;
```

Loyihaning oynada(Form)gi ko‘rinishi va jarayon ishga tushirilgandan keying ko‘rinishi quyidagi rasmlarda keltirildi.



a) Dasturning boshlang‘ich ko‘rinishi



b) Dasturning ishlayotgan ko‘rinishi

Loyihaning Button1 va Button2 obyektlarida quyidagicha vazifalar yozilgan:

Button1	Button2
procedure TForm1.Button1Click (Sender: TObject); begin Timer1.Enabled:=True; MyThead:=TMyThread.Create(False); Button1.Enabled:=False; end;	**procedure TForm1.Button2Click (Sender: TObject); begin Timer2.Enabled:=True; Button2.Enabled:=False; end;**

Kodda keltirilgan buyruqlarda ikkala Button obyektlari ham tegishli Timer1 va Timer2 obyektlarini ishga tushirmoqda. Timet1 ning **Interval = 2000**, Timet2 ning **Interval = 5000**, ya’ni birinchisi 2 sekend, ikkinchisi 5 sekund vaqt intervalida amal bajarishni kutib turadi. Button1 obyektni ishlatilishi **TMyThread.Create** buyrug‘i bilan yangi oqim tashkil etib, yuqoridagi taxminiy sonlarni tanlash protsedurasi(jarayoni)ni faollashtirib beradi.

Loyihani ishlatish jarayonida Button1 obyekti ishlatilganda Edit1 obyektiga taxminiy butun son olib yozilmoqda. Edit1 ning qiymati o‘z navbatida Memo1 obyektiga borib tushirilmoqda. Har safar turli qiymatlar olinayotganligi va dastur o‘z vazifasi belgilangan shaklda bajarilayotganlini kuzatish mumkin. Button2 obyekti ishlatilsa, taxminiy generatsiyadan haqiqiy son olinib, Edit2 obyektiga, Edit2ning qiymati esa ListBox1 obyektiga jo‘natilmoqda. Dastur taxminiy butun va haqiqiy sonlarni tegishli obyektlarga yozib bordi. Dastur bajarilish jarayonida osilib qolish kuzatilmadi.

Yuqoridagi amallarni parallel oqimli bo‘lmagan, oddiy loyiha bilan bajarilishi, jarayon taxminiy sonlarni generatsiya qilish cheksiz siklining ishga tushirilishi bilan kompyuterning osilib qolishi kuzatiladi.

Xulosa sifatida shuni aytish mumkinki, ko‘p vaqt talab etiladigan, murakkab hisoblash amallari bajariladigan loyihalarda, aynan oqimli parallel dasturlash amallari talab etiladigan jarayonlarda parallel oqimli dasturlashdan foydalanish tavsiya qilinadi.

References:

1. E.D.Karepova "Основы многопоточного и параллельного программирования". Krasnoyarsk. SFU.-2016 y.
2. UZ Narziyev, IB Turayev “Oqimli dasturlash asoslari”. Involta Scientific Journal, 6 (2022): 150-152.
3. Makarevich A.G. Эффективная Многопоточность. Minsk 2013 y.
4. Narziev, Umidjon Zaripovich. "Mantiqiy qurilmalarni dasturlashda vhdl dasturlash tilidan foydalanish usullari." Academic research in educational sciences 4 (2020): 492-497.
5. Раззаков, Шавкат Инсанович, Умиджон Зарипович Нарзиев, and Расул Бакдурдиевич Рахимов. "Контроль знаний в системе дистанционного обучения." Молодой ученый 7 (2014): 70-73.
6. Ёдгоров, Ориф Очилович, Умиджон Зарипович Нарзиев, and Нигора Амановна Шарапова. "Классификация различных форм тестирования программного обеспечения." Современные инструментальные системы, информационные технологии и инновации. 2014.
7. Нарзиев, У. З., and ET Сафаров. "Регулярные алгоритмы адаптивного оценивания вектора состояния управляемых объектов." Теоретические знания–в практические дела [Текст]: Сборник научных статей (2014): 378.
8. Ёдгоров, Ориф Очилович, Шавкат Инсанович Раззаков, and Умиджон Зарипович Нарзиев. "Применение адаптивной фильтрации при распознавании изображений." Современные материалы, техника и технология. 2013.
9. Murodova, Zarina Rashidovna, et al. "Creating an Electronic Textbook in a Programming Environment." European Multidisciplinary Journal of Modern Science 4 (2022): 536-544.