



FAYL KATALOGLAR SHAJARASINI HOSIL QILISH FOYDASI

Umarov Bekzod Azizovich

Farg'ona davlat universiteti amaliy matematika va informatika kafedrasida katta o'qituvchisi

ubaumarov@gmail.com

Karimova Nargizaxon Abdurasul qizi

Farg'ona davlat universiteti 2-kurs talabasi

nargizaxon592@gmail.com

Annotatsiya. Ushbu maqolada fayl kataloglarining shajara (tree) ko'rinishida tashkil qilinishi va bu jarayonni dasturiy tarzda hosil qilish usullari tahlil qilinadi. Fayl tizimining mantiqiy tuzilmasi, kataloglar orasidagi ierarxik bog'liqlik va shajaraviy model asosida ularni tartiblashning afzalliklari ko'rib chiqiladi. Shuningdek, rekursiv yondashuv asosida dasturlash tillarida fayl kataloglar shajarasini yaratish amaliyoti namoyish etiladi.

Kalit so'zlar: fayl tizimi, katalog, shajara modeli, rekursiya, fayl tuzilmasi, ierarxiya, tizimli dasturlash, C#, operatsion tizim

Аннотация. В данной статье рассматривается организация файловых каталогов в виде древовидной структуры и методы их программного построения. Анализируется логическая структура файловой системы, иерархические связи между каталогами и преимущества представления в виде дерева. Также демонстрируется практика построения дерева каталогов с использованием рекурсивного подхода в языках программирования.

Ключевые слова: файловая система, каталог, древовидная структура, рекурсия, файловая иерархия, системное программирование, C#, операционная система

Annotation. This article explores the organization of file directories as a tree structure and the methods for programmatically generating them. It analyzes the logical structure of the file system, the hierarchical relationships between directories, and the advantages of tree-based representation. The article also demonstrates the practical implementation of directory trees using recursive approaches in programming languages.

Keywords: file system, directory, tree structure, recursion, file hierarchy, systems programming, C#, operating system



Kirish

Zamonaviy kompyuter tizimlarida ma'lumotlarni samarali saqlash va ularga tezkor kirish imkonini berish uchun fayllar maxsus tuzilma asosida tashkil qilinadi. Bu tuzilma, odatda, fayl kataloglar shajarasi deb ataladi. Shajara — bu ierarxik (daraxtsimon) model bo'lib, unda har bir katalog (papka) boshqa katalog ichida joylashgan bo'lishi mumkin. Natijada, ildiz katalogdan boshlab pastga qarab turli darajadagi kataloglar hosil bo'ladi. Bunday tuzilma foydalanuvchiga fayllarni tartibli saqlash, kerakli fayl yoki papkani tez topish, hamda yirik hajmdagi ma'lumotlarni tizimli boshqarish imkonini beradi. Fayl kataloglar shajarasi nafaqat foydalanuvchi interfeysi darajasida, balki operatsion tizimlar va dasturlarda ham chuqur texnik asosga ega. Ushbu maqolada fayl kataloglar shajarasining mohiyati, ierarxik tuzilma asosida qanday tashkil etilishi va amaliy dasturlashda bu modeldan qanday foydalanish mumkinligi batafsil tahlil qilinadi.

Asosiy qismi:

Fayl kataloglarining shajaraviy tuzilmasi zamonaviy kompyuter tizimlarida eng ko'p qo'llaniladigan va samarali boshqaruv modelidir. Ushbu model orqali fayllar va kataloglar o'zaro ierarxik munosabat asosida joylashtiriladi. Shajaraviy tuzilma daraxt ko'rinishiga ega bo'lib, u ildiz katalogdan boshlanadi va pastga qarab bir nechta darajali papkalar orqali rivojlanadi. Har bir katalog o'z navbatida boshqa kataloglar (ichki kataloglar) va fayllarni o'z ichiga oladi. Bu shakl tizimni tartibga solish, optimallashtirish va boshqarish jarayonlarini sezilarli darajada yengillashtiradi. Fayl kataloglar shajarasi uchta asosiy elementdan iborat bo'ladi: ildiz katalog, ichki kataloglar, va fayllar. Ildiz katalog — bu butun fayl tizimining boshlanish nuqtasi. Uning ichida birinchi darajadagi kataloglar bo'ladi, ular esa o'z navbatida yana boshqa kataloglarni va fayllarni o'z ichiga oladi. Bu ierarxik munosabat ota-bola printsipli asosida quriladi: har bir katalog o'zining ota katalogiga bo'ysunadi va unga bog'liq holda faoliyat yuritadi. Masalan, C:\Users\Admin\Documents\file.txt shaklidagi yo'l orqali faylning qayerda joylashganligini aniq bilish mumkin. Bu yo'l ketma-ketlik asosida ildizdan to faylgacha bo'lgan barcha kataloglarni aks ettiradi. Shajara modeli bir nechta asosiy afzalliklarga ega. Birinchidan, bu tuzilma orqali fayllarni mantiqiy va funktsional guruhlash imkoniyati yaratiladi. Masalan, bir xil loyihaga oid barcha fayllar alohida katalogda saqlanishi mumkin, bu esa tizimdagi tartibni ta'minlaydi. Ikkinchidan, xavfsizlik darajasi oshadi — har bir katalog va uning ichidagi fayllar uchun alohida ruxsat darajalari belgilanishi mumkin. Bu ayniqsa ko'p foydalanuvchili tizimlarda muhim hisoblanadi. Uchinchidan, masshtablash imkoniyati mavjud: yirik tashkilotlarda yoki serverlarda minglab katalog va millionlab fayllar mavjud bo'lishi.



mumkin, va shajaraviy tuzilma bu hajmni boshqarishda o'zini oqlaydi. Yana bir muhim jihat — shajara modeli nafaqat vizual ko'rinishda, balki operatsion tizimlar darajasida ham asosiy mexanizmdir. Windows, Linux, macOS kabi tizimlar fayl tizimlarini aynan shu model asosida tashkil qiladi. Bundan tashqari, dasturchilar dasturiy loyihalarda ham xuddi shu tuzilmani amalda qo'llaydilar: modullar, paketlar, konfiguratsiya fayllari alohida papkalarda tartiblanadi. Shuning uchun kataloglar shajarasi dasturchi uchun nafaqat saqlash tizimi, balki dasturiy arxitektura asosidir. Hozirgi kunda ko'plab dasturlash tillari orqali fayl kataloglar shajarasini avtomatik ravishda hosil qilish, o'qish, tahlil qilish va qayta tuzish imkoniyati mavjud. Bunda, ayniqsa, rekursiv algoritmlar keng qo'llaniladi. Chunki har bir katalog ichida yana katalog bo'lishi mumkin, bu esa tahlil jarayonini chuqurlashtiradi va rekursiyani tabiiy yechimga aylantiradi.

Amaliy qismi:

Fayl kataloglar shajarasini dasturiy tarzda hosil qilishda rekursiv yondashuv asosiy usullardan biridir. C# dasturlash tilida bu vazifani bajarish uchun System.IO kutubxonasi yordamida kataloglar va fayllar ustida amallar bajarish mumkin. Dastur rekursiv tarzda har bir katalogni tekshiradi, uning ichidagi fayllar va ichki kataloglarni ajratadi, so'ng ularning ichiga kirib, ayni operatsiyalarni takrorlaydi. Natijada butun katalog daraxti hosil bo'ladi va ekranga chiqariladi.

Quyida katalog shajarasini hosil qilishga oid C# tilidagi oddiy rekursiv dastur misoli keltirilgan:



```
using System;
using System.IO;
class Program
{ static void Main(string[] args)
    { string rootPath = @"C:\Test"; // Bu yerga siz tekshirmoqchi bo'lgan ildiz
      papkani yozing
      TraverseDirectory(rootPath, 0); }
  static void TraverseDirectory(string path, int level)
  { try
    { // Katalog nomini chiqarish
      Console.WriteLine($"{new string(' ', level * 2)}{
Path.GetFileName(path)}");

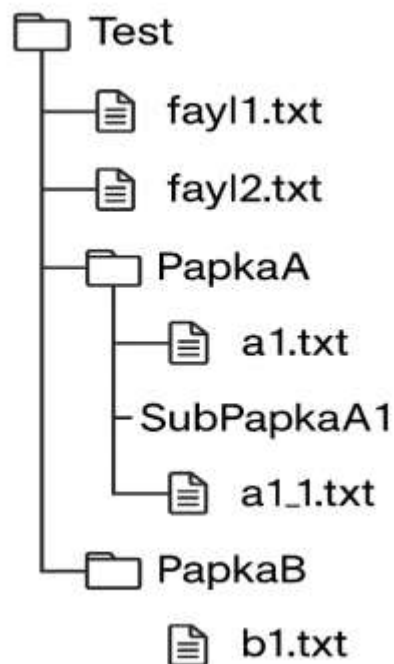
      // Ichidagi fayllarni chiqarish
      foreach (string file in Directory.GetFiles(path))
```



```
Console.WriteLine($"{new string(' ', (level + 1) * 2)} □  
{Path.GetFileName(file)}");
```

```
// Ichki kataloglar bo'yicha rekursiv chaqirish  
foreach (string dir in Directory.GetDirectories(path))  
{  
    TraverseDirectory(dir, level + 1); } }  
catch (UnauthorizedAccessException)  
{  
    Console.WriteLine($"{new string(' ', level * 2)} □ Kirish taqiqlangan:  
{path}"); }  
catch (Exception ex)  
{ Console.WriteLine($" □ Xatolik: {ex.Message}");
```

Natijasi:



Ushbu dastur quyidagi vazifalarni bajaradi:

- Berilgan ildiz katalogdan boshlab, uning barcha ichki kataloglari va fayllarini rekursiv tarzda aylanadi.
- Har bir elementni daraxt ko'rinishida ekranga chiqaradi, darajaga qarab bo'sh joy qo'shadi.



c) Xatoliklar, ayniqsa, kirish huquqlari bo'yicha cheklovlar yuzaga kelganda, ularni foydalanuvchiga bildiradi.

Bu kabi yondashuvlar dasturchilarga katta hajmdagi fayl tuzilmasini tahlil qilish, zaxira nusxalar yaratish, kataloglar sathlarini ko'rsatish yoki fayl boshqaruv interfeysi ishlab chiqish uchun xizmat qiladi.

Amaliy jihatdan bunday dasturlar quyidagi sohalarda foydali:

1. Server fayl monitoringi
2. Avtomatik hisobot yaratish
3. Boshqaruv panellari uchun interfeys yaratish
4. IDE lar (masalan, Visual Studio, Rider) tomonidan ishlatiladigan loyiha daraxtlari

Xulosa.

Xulosa qilib aytganda, fayl kataloglar shajarasi — bu zamonaviy kompyuter tizimlarida axborotlarni mantiqiy, tartibli va qulay boshqarish uchun qo'llaniladigan asosiy model hisoblanadi. Shajaraviy tuzilma orqali fayllar va kataloglar orasidagi ierarxik bog'liqlik aniq ko'rinishda ifodalanadi. Bu modelning mavjudligi operatsion tizimlar, dasturiy ta'minotlar va foydalanuvchi darajasidagi amaliyotlar uchun ajralmas vosita bo'lib xizmat qiladi. Ushbu maqolada kataloglar shajarasining nazariy mohiyati, amaliy afzalliklari hamda uni C# dasturlash tilida rekursiv algoritm yordamida hosil qilish usullari yoritib berildi. Amaliy misolda shajara modeli orqali fayl va kataloglar tuzilmasini qanday avtomatlashtirish mumkinligi ko'rsatildi. Dastur orqali kataloglarni bosqichma-bosqich tahlil qilish, ularni daraxt shaklida chiqarish va foydalanuvchi uchun tushunarli ko'rinishda taqdim etish imkoniyati yaratilgan. Fayl kataloglarining shajaraviy modeli nafaqat strukturaviy jihatdan qulay, balki xavfsizlik, masshtablash va tizimni optimallashtirish kabi funksional imkoniyatlarga ham boydir. Dasturchilar, tizim administratorlari hamda IT sohasida faoliyat yurituvchi mutaxassislar uchun bu modelni chuqur o'zlashtirishkundalik amaliyotda unumdorlikni oshirish, yirik loyihalarni samarali boshqarish va ma'lumotlar ustida mustahkam nazoratni ta'minlashda muhim ahamiyatga ega.

Foydalanilgan adabiyotlar:

1. V.Jo'rayev, B.Umarov. "Kompyuter tarmoqlari", 2024.
2. B.Umarov., M.Umarova. THE PROBLEM OF APPROXIMATING SIGNALS BASED ON MODELING OF WAVELET - HAAR TRANSFORMATION. - 2020. - C. 502-506.



3. Azizovich UB. INNOVATSION TEXNOLOGIYALAR ORQALI O'QITUVCHILAR KOMETANSIYATINI SHAKLLANTIRISH TASOSIYLARI. Finlyandiya xalqaro ta'lim ilmiy jurnali. Ijtimoiy va gumanitar fanlar. 2023;11(5):823-8.

4. Umarov, B., G'ulomjonova, S. (2024). BULUT TEXNOLOGIYASI VA ULARDAN FOYDALANISH. Zamonaviy ta'limda innovatsion tadqiqotlar, 2(7), 12-14. Qaytadan olindi

5. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th Edition). Wiley..

6. Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th Edition). Pearson.

7. Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th Edition). Pearson.

8. Nutt, G. J. (2004). *Operating Systems: A Modern Perspective* (2nd Edition). Addison Wesley.

9. Bovet, D. P., & Cesati, M. (2005). *Understanding the Linux Kernel* (3rd Edition). O'Reilly Media.

10. Silberschatz, A., & Galvin, P. (2003). *Operating System Concepts Essentials*. Wiley.

11. William Shotts (2019). *The Linux Command Line: A Complete Introduction*. No Starch Press.

12. Gagne, G., Galvin, P., & Silberschatz, A. *Operating System Concepts with Java*. Wiley.

13. Online sources such as GeeksforGeeks.org, TutorialsPoint.com, and IBM Developer Docs.

14. O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi. *Axborot texnologiyalari va operatsion tizimlar* (darsliklar va o'quv qo'llanmalari).

15. <https://www.geeksforgeeks.org>