

UDK: 004.42:004.651

*Shermatova Xilola Mirzayevna**FarDU Axborot texnologiyalari kafedrası dotsenti**shermatovahilola1978@gmail.com <https://orcid.org/0000-0001-5014-9549>*

998904099095

*Tursunaliyeva Sarvinoz G'ulomjon qizi**Farg'ona davlat universiteti Axborot tizimlari va**texnologiyalari yo'nalishi I kurs talabasi**sarvinot280@gmail.com*

C++DA FAYL TIZIMI BILAN ISHLASH

Annotatsiya: Ushbu maqolada C++ dasturlash tilida fayl tizimi bilan ishlashning asosiy tamoyillari yoritiladi. Maqolada fayl tizimi tushunchasi, uning asosiy elementlari va fayl tizimi turlari batafsil ko'rib chiqiladi. C++ tilida fayllarni yaratish, o'qish va yozish algoritmlari tushuntiriladi va misollar bilan ko'rsatib beriladi. Shuningdek, fayl tizimining samaradorligi, uni oshirish usullari va fayl tizimi xavfsizligi masalalari ham yoritiladi. Maqolada keltirilgan bilim va misollar dasturlash jarayonida fayllar bilan samarali ishlash ko'nikmalarini shakllantirishga xizmat qiladi.

Аннотация: В данной статье рассматриваются основные принципы работы с файловой системой на языке программирования C++. В статье подробно рассматривается понятие файловой системы, ее основные элементы и типы файловых систем. Алгоритмы создания, чтения и записи файлов на языке C++ объясняются и иллюстрируются примерами. В нем также рассматриваются эффективность файловой системы, методы ее повышения и вопросы безопасности файловой системы. Представленные в статье знания и примеры помогут вам развить навыки эффективной работы с файлами в процессе программирования.

Abstract: This article covers the basic principles of working with a file system in the C++ programming language. The article examines in detail the concept of a file system, its main elements, and types of file systems. The algorithms for creating, reading, and writing files in C++ are explained and illustrated with examples. The efficiency of the file system, methods for improving it, and the security of the file system are also covered. The knowledge and examples presented in the article serve to develop skills in working effectively with files in the programming process.

Kalit so'zlar: Fayl tizimi, C++, Fayl yaratish, Fayldan o'qish, Fayldan yozish, Samaradorlik, Kesh tizimi, Fragmentatsiya, Shifrlash, Jurnalizatsiya, Indeksflash, NTFS, FAT32, exFAT.

Ключевые слова:Файловая система, C++, Создание файлов, Чтение из файла, Запись из файла, Эффективность, Система кэширования, Фрагментация, Шифрование, Журналирование, Индексирование, NTFS, FAT32, exFAT.

Keywords:File system, C++, File creation, Reading from file, Writing from file, Efficiency, Cache system, Fragmentation, Encryption, Journaling, Indexing, NTFS, FAT32, exFAT.

KIRISH

Hozirgi kunda dasturlash jarayonida ma'lumotlarni saqlash va qayta ishlash katta ahamiyatga ega. Dasturlash tillari orasida C++ tili o'zining yuqori samaradorligi va qulay imkoniyatlari bilan ajralib turadi. Fayllar bilan ishlash dasturlarda ma'lumotlarni saqlash, o'qish va qayta ishlash jarayonining muhim qismini tashkil etadi. Fayllar bilan samarali ishlash orqali dastur ish unumdorligi oshadi va ma'lumotlarning xavfsizligi ta'minlanadi.

Mazkur maqolada C++ da fayl tizimi bilan ishlash jarayonlari, fayllarni yaratish, o'qish va yozish algoritmlari, fayl tizimi samaradorligini oshirish usullari, shuningdek, fayl tizimi xavfsizligi masalalari yoritiladi.

FAYL TIZIMI TUSHUNCHASI VA UNING ASOSIY ELEMENTLARI

Fayl tizimi – bu kompyuterda ma'lumotlarni saqlash, boshqarish va ularga murojaat qilish uchun ishlatiladigan dasturiy ta'minot qoidalari majmuasidir. Fayl tizimi quyidagi asosiy elementlardan iborat:

Fayllar – ma'lumotlarni saqlovchi asosiy obyekt.

Kataloglar (papkalar) – fayllarni guruhlash va boshqarish vositasi.

Fayl atributlari – fayl turi, hajmi, yaratish vaqti kabi ma'lumotlar.

Klasterlar – diskdagi ma'lumotlarni saqlash bloklari.

Indekslar – fayl manzillarini boshqarish mexanizmi.

Fayl tizimlari turli operatsion tizimlarda turlicha tashkil qilinadi:

FAT (File Allocation Table) – kichik hajmdagi fayllar uchun mos.

FAT32 – kattaroq fayllar uchun mos, lekin 4 GB dan oshmaydi.

exFAT – 4 GB dan katta fayllarni qo'llab-quvvatlaydi.

NTFS – xavfsizlik, shifrlash va hajmni boshqarishda ustunlikka ega.

C++DA FAYL YARATISH, O'QISH VA YOZISH ALGORITMLARI

C++ dasturlash tilida fayllar bilan ishlash uchun fstream kutubxonasidan foydalaniladi. Ushbu kutubxonada quyidagi asosiy sinflar mavjud:

ofstream – faylga yozish uchun ishlatiladi.

ifstream – fayldan o'qish uchun ishlatiladi.

fstream – faylga yozish va o'qish uchun ishlatiladi.

Fayl Yaratish Algoritmi

Fayl yaratish uchun quyidagi bosqichlar bajariladi:

1. **#include <fstream>** kutubxonasini ulash.

2. **ofstream** obyektini yaratish.

3. **open()** funksiyasi orqali fayl yaratish yoki ochish.

4. << operatori orqali faylga yozish.

5. **close()** bilan faylni yopish.

Misol:

```
#include <fstream>
```

```
using namespace std;
```

```
int main() {
```

```
    ofstream fayl("mashq.txt");
```

```
    fayl << "Assalomu alaykum!";
```

```
    f.close();
```

```
    return 0; }
```

Fayldan o'qish algoritmi

Fayldan o'qish uchun quyidagi bosqichlar bajariladi: 1. ifstream obyektini yaratish.

2. Faylni open() orqali ochish. 3. >> yoki getline() bilan o'qish. 4. O'qilgan ma'lumotni ekranga chiqarish yoki dastur ichida qayta ishlash. 5. close() orqali faylni yopish.

Misol:

```
#include <fstream>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    ifstream fayl("example.txt");
```

```
    string matn;
```

```
    while (getline(fayl, matn)) {
```

```
        cout << matn << endl;
```

```
    }
```

```
    fayl.close();
```

```
    return 0; }
```

Faylga yozish algoritmi

Faylga yozish uchun quyidagi bosqichlar bajariladi:

1. Fayl obyektini yaratish.
2. Faylni open() orqali ochish.
3. << operatori bilan faylga yozish.
4. close() orqali faylni yopish.

Misol:

```
#include <fstream>
```

```
using namespace std;
```

```
int main() {
```

```
    ofstream fayl("example.txt", ios::app);
```

```
    fayl<< "Yangi satr\n";
```

```
    fayl.close();
```

```
    return 0;
```

```
}
```

FAYL TIZIMIDA MA'LUMOTLARNI SAQLASH VA QAYTA ISHLASH SAMARADORLIGI

Fayl tizimining samaradorligi quyidagi omillarga bog'liq: Tezlik - ma'lumotlarni qayta ishlash vaqti. Disk maydonidan foydalanish – fayllarni tartibli joylashtirish. Kesh tizimi – tez-tez foydalaniladigan fayllarni vaqtinchalik saqlash. Fragmentatsiya – fayllarning diskdagi joylashuvi. Samaradorlikni oshirish usullari

1. Fragmentatsiyani kamaytirish – defragmentatsiya qilish.
2. To'g'ri fayl tizimini tanlash – operatsion tizimga mos keladigan fayl tizimini ishlatish.
3. Kesh mexanizmidan foydalanish – ko'p ishlatiladigan ma'lumotlarni vaqtinchalik xotirada saqlash.
4. Parallel ishlash – bir nechta jarayon orqali fayllarni boshqarish.
5. Fayl indeksatsiyasi – qidiruv jarayonini tezlashtirish.

Fayl Tizimi Xavfsizligi

1. Zaxira nusxa yaratish – muhim ma'lumotlarni saqlash.
2. Shifrlash – ma'lumotlarni maxfiy saqlash.
3. Fayl ruxsatnomalari – fayllarga ruxsat asosida murojaat qilish.

4. Jurnalizatsiya – tizim o‘zgarishlarini qayd qilish.

XULOSA

Ushbu maqolada C++ dasturlash tilida fayl tizimi bilan ishlashning asosiy jihatlari yoritildi. Fayllar bilan ishlash dasturlash jarayonining ajralmas qismi hisoblanadi, chunki dasturlar ko‘pincha ma’lumotlarni saqlash, qayta ishlash va uzatish bilan bog‘liq vazifalarni bajaradi. C++ dasturlash tili fayllar bilan ishlash uchun keng imkoniyatlarni taqdim etadi.

Maqolada fayl tizimining asosiy tushunchalari, fayl tizimining turlari va ularning xususiyatlari ko‘rib chiqildi. FAT, FAT32, exFAT va NTFS kabi fayl tizimlari o‘ziga xos jihatlari bilan bir-biridan farqlanadi. Masalan, FAT tizimi kichik hajmdagi fayllar bilan ishlash uchun mos bo‘lsa, NTFS tizimi kattaroq hajmdagi fayllarni boshqarish, ma’lumotlarni shifrlash va xavfsizlikni ta’minlash imkoniyatiga ega.

C++ da fayllarni yaratish, o‘qish va yozish algoritmlari batafsil tushuntirildi. Fayllarni yaratish uchun ofstream, fayldan o‘qish uchun ifstream, faylga yozish va o‘qish uchun esa fstream sinflari ishlatiladi. Fayl bilan ishlashda ios::app, ios::in, ios::out, ios::trunc va ios::binary kabi rejimlardan foydalanish mumkin. ios::app rejimida fayl mavjud bo‘lsa, yangi ma’lumot fayl oxiriga qo‘shiladi. ios::out rejimi esa mavjud fayldagi ma’lumotni o‘chiradi va yangidan yozadi. Shuningdek, maqolada fayl tizimining samaradorligini oshirish usullari ham yoritildi. Fayl tizimlari orasidagi farqlar ham ko‘rib chiqildi.

C++ tilida fayllar bilan ishlashni mukammal o‘zlashtirish dasturlash samaradorligini oshiradi. Dasturchilar turli fayl tizimlarining xususiyatlarini bilgan holda ma’lumotlarni saqlash va boshqarishni optimallashtirishi mumkin. Fayllarni yaratish, o‘qish va yozish jarayonini to‘g‘ri tashkil qilish dastur ish unumdorligini oshiradi, ma’lumotlarni xavfsiz boshqarishga imkon beradi.

Yakun qilib aytganda, fayl tizimlari bilan ishlash dasturlash jarayonida muhim ahamiyatga ega. To‘g‘ri fayl tizimini tanlash, samaradorlikni oshirish, xavfsizlikni ta’minlash va faylni boshqarish jarayonlarini tushunish – bu dasturchilar uchun zarur ko‘nikmalardir. Fayllar bilan ishlashni samarali tashkil qilish natijasida dasturlar barqaror ishlaydi, ma’lumotlar xavfsizligi ta’minlanadi va ishlash tezligi oshadi.

FOYDALANILGAN ADABIYOTLAR:

1. Stroustrup B. The C++ Programming Language – Addison-Wesley, 2013.
2. Lippman S.B., Lajoie J., Moo B.E. C++ Primer – Addison-Wesley, 2012.
3. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 – O'Reilly Media, 2014.
4. Josuttis N.M. The C++ Standard Library: A Tutorial and Reference – Addison-Wesley, 2012.
5. Kerrisk M. The Linux Programming Interface – No Starch Press, 2010.
6. Overland B. C++ Without Fear: A Beginner's Guide That Makes You Feel Smart – Addison-Wesley, 2015.
7. Koenig A., Moo B.E. Accelerated C++: Practical Programming by Example – Addison-Wesley, 2000.
8. ISO/IEC 14882:2017. Programming Languages — C++ – International Organization for Standardization, 2017.
9. Eckel B. Thinking in C++ – Prentice Hall, 2000.
10. Balagurusamy E. Object-Oriented Programming with C++ – McGraw-Hill Education, 2013.
11. Sutter H. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions – Addison-Wesley, 2000.

12. Horton I. Beginning C++: The Complete Guide to C++ Programming – Wrox, 2014.
13. Schildt H. C++: The Complete Reference – McGraw-Hill, 2012.
14. Savitch W.J. Problem Solving with C++ – Pearson, 2018.
15. Malik D.S. C++ Programming: From Problem Analysis to Program Design – Cengage Learning, 2017.