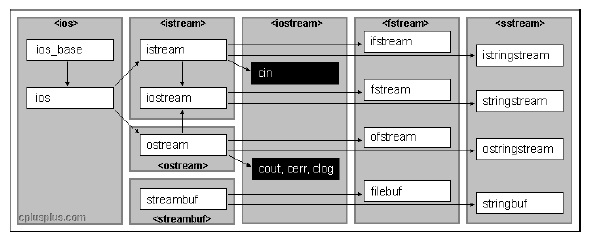
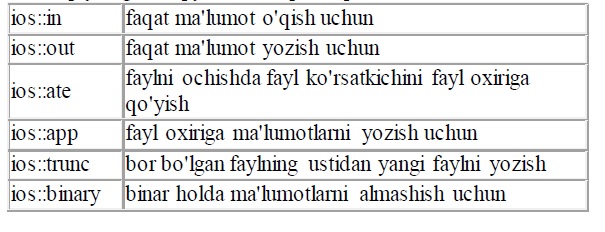
**Fayllar bilan ishlash. Matnli fayllar**

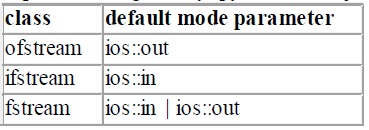
Matnli fayllar bilan ishlash binar fayllar bilan ishlashdan bir oz farq qiladi. Matnli fayllarda ma`lumotlar satrlarda saqlanadi. Matnli fayl elementilari har xil uzunlikdagi satrlardir. Bu satrlar bir biridan satr oxiri belgisi bilan ajratiladi. Matnli fayl elementlari indekslanmagan bo`lganligi uchun, faylning istalgan elementiga bevosita murojaat qilib bo`lmaydi. C++ da matnli yoki binar fayllar bilan ishlash uchun keng imkoniyatlar berilgan. Matnli fayllar bilan ishlashda oddiy C ning funksiyalaridan ham foydalanish mumkin. Masalan, formatli o'qish va yozish funksiyalari yoki oldingi mavzudagi funksiyalardan foydalanishimiz mumkin. Matnli fayllar bilan ishlashning bunday usuli kitoblarda keng yoritilgan. Ularni mustaqil o'qib - o'rganishingiz mumkin. Bu mavzu fayllar bilan ishlovchi oqimlarni qisqacha o'rganamiz va buni matnli fayl misolida ko'ramiz. Standart kiritish / chiqarish kutubxonasi sinflari quyidagicha shajaraga ega:



**Fayllar bilan ishlash uchun quyidagi sifnlar ob'ektlari hosil qilinadi:**  
• ofstream - faylga ma'lumot yozish uchun  
• ifstream - fayldan ma'lumot o'qish uchun  
• fstream - fayldan ma'lumot o'qish uchun va yozish uchun  
Bu sinflarni dasturda ishlatish uchun <fstream> sarlavha faylini qo'shish kerak bo'ladi. Bundan keyin programmada aniq fayllar oqimini aniqlash mumkin. Masala: ofstream yozish; // faylga yozish oqimini e'lon qilish ifstream oqish; // fayldan o'qish oqimini e'lon qilish fstream yoz\_oqi; // faylga yozish va o'qish oqimini e'lon qilish Keyin faylni ochish kerak bo'ladi. Faylni ochish deganda, uning ustida nima amal qilinishi haqida amaliyot tizimiga xabar berish tushuniladi. void open (const char \* filename, ios\_base::openmode mode = ios\_base::out ); mode parametri quyidagicha qiymatlarni qabul qilishi mumkin:



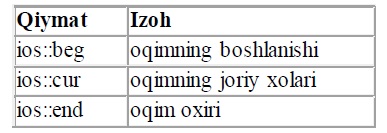
Har bir sinf uchun mode parametrining odatiy qiymatlari mavjud:



Fayl ustida o'qish yoki yozish amalini bajarib bo'lgandan song, faylni yopish kerak bo'ladi. Faylni yopish uchun close funksiyadi ishlatiladi. Masalan:  
yozish.close(); oqish.close();  
**Matnli faylga ma'lumot yozish**  
#include <iostream>

#include <fstream>  
using namespace std;  
int main ()  
{  
ofstream yozish; // faylga yozish oqimini hosil qilish  
yozish.open("namuna.txt");  
// yangi namuna.txt nomli fayl hosil qilinadi.  
// agar namuna.txt fayli oldindan bo'lsa,  
// uning eski qiymatlari o'chiriladi  
// va yangi fayl hosil qilinadi  
yozish << "Matnli faylga ma'lumot yozish" << endl;  
yozish << "Juda oson!" << endl;  
yozish.close(); // faylni yopish  
return 0;  
} **Matnli fayldan o'qish**  
#include <iostream>  
#include <fstream>  
#include <string>  
using namespace std;  
int main ()  
{  
ifstream oqish; // fayldan o'qish oqimini hosil qilish  
string satr;  
oqish.open("namuna.txt");  
// faylni ochishda xatolik sodir bo'lsa  
if (!oqish.is\_open())  
{  
cout << "Faylni ochishda xatolik sodir bo'ldi." << endl;  
exit(1); // dasturni tugatish  
}  
while (!oqish.eof())  
{  
// fayldan o'qish  
getline(oqish, satr);  
// ekranga chiqarish  
cout << satr << endl;  
}  
// namuna.txt fayli bilan oqish oqimi aloqasini uzish

oqish.close();  
return 0;  
}  
istream sinfi funksiyalari  
istream& seekg ( streampos pos ); istream& seekg ( streamoff off, ios\_base::seekdir dir );;  
oqish oqimi ko'rsatkichini o'rnatish (siljitish).  
pos - oqim buferining yangi pozitsiyasi. dir parametri quyidagilardan birini qabul qilishi mumkin:



long tellg();  
o'qish oqimining joriy xolatini (pozitsiyasi) aniqlash. ostream sinfi funksiyalari ostream& seekp ( streampos pos ); ostream& seekp ( streamoff off, ios\_base::seekdir dir ); yozish oqimi o'rnini (pozitsiyasini) o'rnatish. pos - oqim buferining yangi pozitsiyasi dir parametri beg, cur, end qiymatlaridan birini qabul qilishi mumkin. long tellp() - yozish oqimining kelgan joyini aniqlash. **Fayldan nusxa olish**  
//ushubu dastur orqali ixtiyoriy fayldan nusxa olish mumkin  
#include <iostream>  
#include <fstream>  
using namespace std;  
int main ()  
{  
int length;  
char \* buffer, fayl[] = "matn.txt", yangi[]="yangi\_fayl.txt";

// fayl - nusxalanadigan fayl nomi  
// yangi - yangi nusxalangan fayl nomi  
// o'qish oqimi  
ifstream fromfile(fayl, ios::binary );  
if (!fromfile.is\_open())  
{  
cout << "faylni o'qishda xatolik sodir bo'ldi\n";  
exit(1);  
}  
// yozish oqimi  
ofstream tofile(yangi, ios::binary );  
// fayl xajmini aniqlash:  
fromfile.seekg (0, ios::end); // fayl oxiriga o'tish  
length = fromfile.tellg();  
fromfile.seekg (0, ios::beg); // fayl boshiga o'tish  
// xotira ajratish:  
buffer = new char [length];  
// blokka ma'lumotlarni o'qish:  
fromfile.read (buffer, length);  
fromfile.close();  
// nusxalanishi kerak bo'lgan faylga yozish  
tofile.write (buffer, length);  
// xotirani bo'shatish  
delete[] buffer;  
cout << "fayl nusxalandi\n";  
return 0;  
}  
dic.txt nomli fayl berilgan . Faylning har bir satrida inglizcha va o`zbekcha so`zlar "-" belgisi bilan ajratilgan. Inglizcha so`zlarni english.txt fayliga, o`zbekcha so`zlarni uzbek.txt fayliga o`tkazuvchi programma tuzilsin.  
dic.txt fayli quyidagicha bo'ladi: hello - salom bread - non car - mashina  
#include <iostream>  
#include <fstream>  
#include <string>

using namespace std;  
int main ()  
{  
ifstream dic("dic.txt");  
ofstream uzbek("uzbek.txt");  
ofstream english("english.txt");  
if (!dic.is\_open())  
{  
cout << "dic.txt - fayli topilmadi\n";  
exit(1);  
}  
string s, uzb, eng;  
int p;  
cout << "dic.txt fayli ma'lumotlari\n";  
while (!dic.eof())  
{  
getline(dic, s);  
p = s.find("-");  
eng.assign(s, 0, p - 1);  
uzb.assign(s, p + 1, s.length() - (p + 1));  
uzbek << uzb << endl;  
english << eng << endl;  
cout << s << endl;  
}  
dic.close();  
uzbek.close();  
english.close();  
return 0;  
}