Изготвил:
гл.ас. д-р. Нора Ангелова

Файлове и Манипулатори

Файлове

- Файлове с последователен достъп
- Файлове с пряк достъп (бинарни)

Файлове с последователен достъп

Компонентите на тези файлове са редица от символи завършващи с '\n'.
За да бъде достигнат и прочен елементът с пореден номер n, трябва последователно да бъдат прочетени всички предшестващи го елементи.

abcd 1 da 9 '\n'
. . . . . '\n'
. . . . . '\n'
. . . . . '\n'
. . . . . '\n'

Файлове с пряк достъп

Търсеният елемент се достига направо (по адреса му), без да е необходимо да се прочетат предшестващите го елементи.
Данните, които записваме не се третират като символи.
Пример:
Използват се когато искаме да запишем обекти от даден клас.

обект1 памет
обект2 памет
обект3 памет
...
обектx памет

Файлове с пряк достъп

- За да се достъпи директно елемент на определена позиция е необходимо всички елементи да имат една и съща дължина (големина).
- Може да се използват класове и обекти.

Двоични файлове

- Неформатиран вход и изход
- Позволяват пряк достъп

Бинарни файлове

- read/write

```
oFileName.write((char*)&st1, sizeof(ClassName));

iFileName.read((char*)&st2, sizeof(ClassName));
```

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char c1;
fileName.get(c1);
cout << c1;
```

Резултат
a

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char c1;
while (fileName.get(c1)) {
  cout << c1;
}
```

Резултат
abcd
12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char c1;
while (fileName >> c1) {
  cout << c1;
}
```

Резултат
abcd12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char c1;
while (fileName.get(c1)) {
  cout << c1;
}

char c2;
while (fileName.get(c2)) {
  cout << c2;
}
```

Резултат
abcd
12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char c1;
while (fileName.get(c1)) {
  cout << c1;
}

fileName.clear();
fileName.seekg(0, ios::beg);

char c2;
while (fileName.get(c2)) {
  cout << c2;
}
```

Резултат
abcd
12345abcd
12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char str[10];

fileName.get(str, 10, '\n');
cout << str;
```

Резултат
abcd

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char str[10];
fileName.get(str, 10, '\n');
cout << str;


char str2[10];
fileName.get(str2, 10, '\n');
cout << str2;
```

Резултат
abcd

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char str[10];

fileName.get(str, 10, '\n');
cout << str;

fileName.get();

char str2[10];

fileName.get(str2, 10, '\n');
cout << str2;
```

Резултат
abcd12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

char str[10];
fileName.getline(str, 10, '\n');
cout << str;

char str2[10];
fileName.get(str2, 10, '\n');
cout << str2;
```

Резултат
abcd12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt");

fileName.write("789", 3);
```

Резултат
789d
12345

Пример

Файл
abcd
12345

```
fstream fileName;
fileName.open("C:/Users/eminor/Desktop/file.txt“, ios::app);

fileName.write("789", 3);
```

Резултат
abcd
123457 89

Пример

```
ofstream file("clients.dat", ios::out);
if (!file) {
  cerr << "File couldn’t be opened! \n";
  return 1;
}

int account;
char name[16];
float balance;
while (cin >> account >> name >> balance) {
  file << account << " " << name << " " << balance << '\n';
  cout << '?';
}



//Ctrl+Z
```

Пример

```
struct Student {
  int fn;
  double result;
};
Student st1 = { 44394, 6.0 };

ofstream oFileName;
oFileName.open("C:/Users/eminor/Desktop/file.txt", ios::out|ios::binary); //check

oFileName.seekp(0,ios::beg);
oFileName.write((char*)&st1, sizeof(Student));
oFileName.close();


ifstream iFileName;
iFileName.open("C:/Users/eminor/Desktop/file.txt", ios::in);// check

Student st2;
iFileName.read((char*)&st2, sizeof(Student));
cout << st2.fn << endl;
cout << st2.result << endl;
iFileName.close();
```

Пример

Да се напише програмент фрагмент, който записва данните за шестия студент.

Пример

```
struct Student {
  int fn;
  double result;
};
Student st1 = { 44394, 6.0 };

ofstream oFileName;
oFileName.open("C:/Users/eminor/Desktop/file.txt",
ios::out|ios::binary);

oFileName.seekp(5 * sizeof(Student));
oFileName.write((char*)&st1, sizeof(Student));
oFileName.close();
```

Динамично заделяте на памет

```
class Student {
  int fn;
  double result;
  char * name;

public:
  Student(int fnData=0, double resultData=0, char* nameData="no") {
    fn = fnData;
    result = resultData;
    name = new char[strlen(nameData)+1];
    strcpy(name, nameData);
  }
};

***
Student st1( 44394, 6.0, "nora angelova");

ofstream oFileName;
oFileName.open("C:/…", ios::out|ios::binary); // check

oFileName.seekp(0,ios::beg);
oFileName.write((char*)&st1, sizeof(Student)); // int + double + char*
oFileName.close();
```

Динамично заделеняте на памет

- При стандартни операции се записват член-данни от тип int, double, char*.
- Динамично заделената памет съществува до нейното изтриване или до приключване на изпълнението на програмата.
- Директен прочит на член-данните ще изведе цялото съдържание.
- След спиране и стартиране на програмата член-данната за име, няма да бъде достъпна – динамичната памет е изтрита.

- Как да запишем динамично заделен масив?

Динамично заделяните на памет

- Как да запишем динамично заделен масив?

```
char * name = new char[20];
strcpy(name, "nora angelova");

ofstream oFileName;
oFileName.open("C:/....txt", ios::out|ios::binary); //check

oFileName.seekp(0,ios::beg);
oFileName.write(name, strlen(name)+1);
oFileName.close();


ifstream iFileName;
iFileName.open("C:/....txt", ios::in);// check

char nameResult[100];
iFileName.read(nameResult, strlen(name) +1);
```

- Как да разберем колко е големината на name при повторно стартинаре на програмата?

Динамично заделеняте на памет

- Как да разберем колко е големината на name при повторно стартинаре на програмата?

```
char * name = new char[20];
strcpy(name, "nora angelova");

ofstream oFileName;
oFileName.open("C:/....txt", ios::out|ios::binary); //check

oFileName.seekp(0,ios::beg);
int strLength = strlen(name);

// Записваме дължината в началото
oFileName.write((char*)& strLength, sizeof(strLength));
oFileName.write(name, strLength);
oFileName.close();
```

Динамично заделяните на памет

- Как да разберем колко е големината на name при повторно стартинаре на програмата?

```
ifstream iFileName;
iFileName.open("C:/....txt", ios::in); // check

int lengthResult = 0;
char nameResult[100];

// Прочитаме размера
iFileName.read((char*)& lengthResult, sizeof(lengthResult));

// Прочитаме низа, може паметта да е диначино заделена
iFileName.read(nameResult, lengthResult);

// Добавяме детерминираща 0
nameResult[lengthResult] = '\0';

cout << nameResult << endl;


iFileName.close();
```

Функции

- istream& ignore (streamsize n = 1, int delim = EOF);

Извлича n символа от потока и ги игнорира.
Спира при достигане на n или delim.

Манипулатори

- Форматиране на вход и изход
- Изисква включване на #include <iomanip>

```
hex, oct, dec;

setbase(int)
int – 8, 10, 16

Пример:
cout << setbase(16) << 124 << endl; // 7c
cout << hex << 124;                 // 7c
```

** Ще важи до използване на друг манипулатор.*

Манипулатори

- Форматиране на вход и изход
- Изисква включване на #include <iomanip>

setprecision

Пример:

```
double f =3.14159;
cout << setprecision(5) << f << '\n'; // 3.1416
cout << setprecision(9) << f << '\n'; // 3.14159
cout << fixed;
cout << setprecision(5) << f << '\n'; // 3.14159
cout << setprecision(9) << f << '\n'; // 3.141590000
```

** Ще важи до използване на друг манипулатор.*

Манипулатори

- Форматиране на вход и изход
- Изисква включване на #include <iomanip>

setprecision

Пример:

```
double f =3.14159;
cout << setprecision(5) << f << '\n'; // 3.1416
cout << setprecision(9) << f << '\n'; // 3.14159
cout << scientific;
cout << setprecision(5) << f << '\n'; // 3.14159e+000
cout << setprecision(9) << f << '\n'; // 3.141590000e+000
```

** Ще важи до използване на друг манипулатор.*

Манипулатори

- Форматиране на вход и изход
- Изисква включване на #include <iomanip>

setf, unsetf

Пример:

```
cout.setf (ios::hex, ios::basefield );
cout << 124 << '\n'; // 7c

cout.unsetf(ios::basefield);
cout << 124 << '\n'; // 124
```

** Ще важи до използване на друг манипулатор или unsetf.*

Манипулатори

- Форматиране на вход и изход
- Изисква включване на #include <iomanip>

`setf, unsetf`

Пример:

fmtfl format flag value	*mask* field bitmask
left, right or internal	adjustfield
dec, oct or hex	basefield
scientific or fixed	floatfield

** Ще важи до използване на друг манипулатор или unsetf.*

Край