Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Сравнение скорости работы форматированного и потокового ввода и вывода»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Проверил:

Белодед Николай Иванович

2023, Минск

**Содержание**

1. Предисловие … 3
2. Сравнение функции scanf\_s() и потока cin… 4

2.1 Сравнение скорости считывания целых чисел … 4

2.2 Сравнение скорости считывания вещественных чисел … 5

2.3 Сравнение скорости считывания символов … 6

1. Сравнение функции printf() и потока cout … 8

3.1 Сравнение скорости вывода целых чисел … 8

3.2 Сравнение скорости вывода вещественных чисел … 9

3.3 Сравнение скорости вывода символов … 10

1. Выводы … 12

**Предисловие**

В данном реферате будут рассмотрены функции printf() и scanf\_s(), которые используются при форматированном вводе/выводе, а также потоки cin и cout, которые используются при потоковом вводе/выводе.

**2.Сравнение функции scanf\_s() и потока cin**

**2.1 Сравнение скорости считывания целых чисел**

Для данной проверки будем использовать файл, содержащий 200000 целых чисел. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cin:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

int a;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

cin >> a;

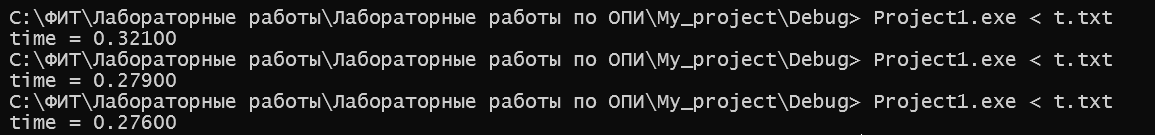
}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:

**Среднее время выполнения:** 0.292 секунды

**Функция scanf\_s():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

int a;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

scanf\_s(“%d”, &a);

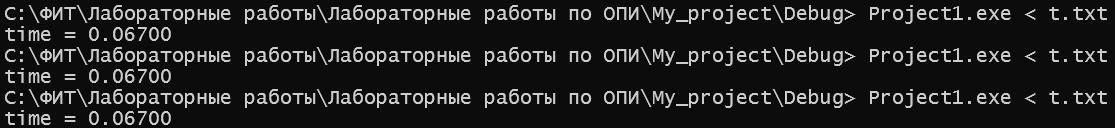
}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:

**Среднее время выполнения:** 0.067 секунды

В данном сравнении очевидно, что функция scanf\_s() быстрее.

**2.2 Сравнение скорости считывания вещественных чисел**

Для данной проверки будем использовать файл, содержащий 20000 вещественных чисел. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cin:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

double a;

clock\_t start = clock();

for (int i = 0; i < 20000; i++) {

cin >> a;

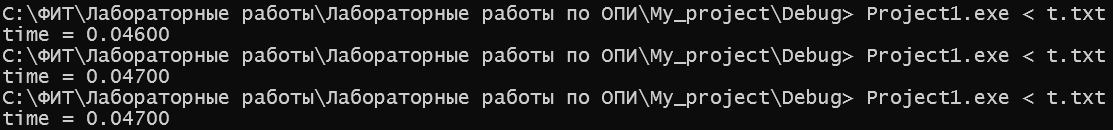
}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:

**Среднее время выполнения:** 0.04667 секунды

**Функция scanf\_s():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

double a;

clock\_t start = clock();

for (int i = 0; i < 20000; i++) {

scanf\_s(“%lf”, &a);

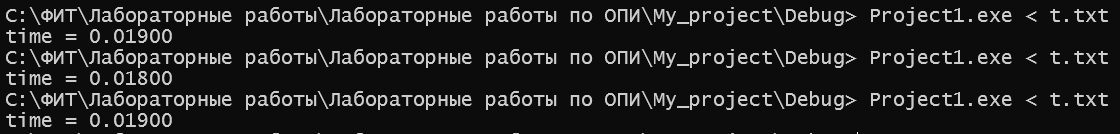
}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:

**Среднее время выполнения:** 0.01867 секунды

В данном сравнении функция scanf\_s() снова оказалась быстрее.

**2.3 Сравнение скорости считывания символов**

Для данной проверки будем использовать файл, содержащий 20000 символов. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cin:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

char a;

clock\_t start = clock();

for (int i = 0; i < 20000; i++) {

cin >> a;

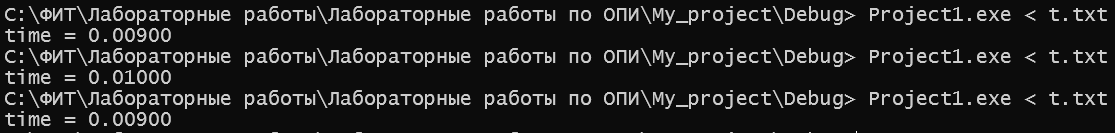
}

clock\_t end = clock();

double duration = (double)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результат выполнения:

**Среднее время выполнения:** 0.00933 секунды

**Функция scanf\_s():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

char a;

clock\_t start = clock();

for (int i = 0; i < 20000; i++) {

scanf\_s(“%c”, &a);

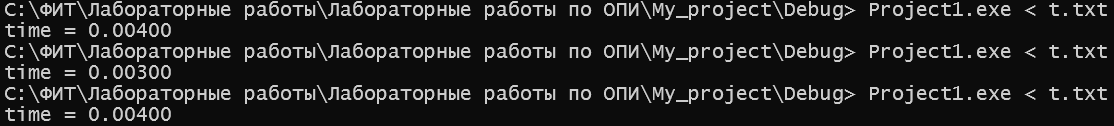
}

clock\_t end = clock();

double duration = (double)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

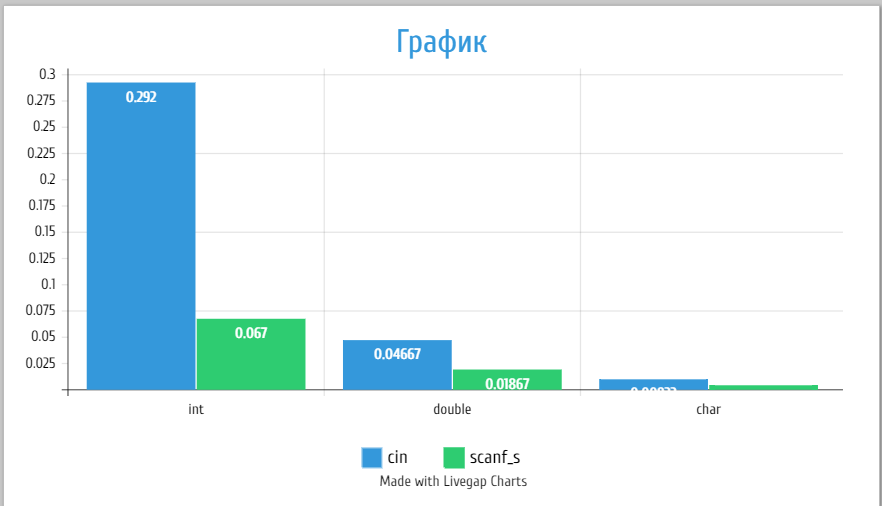
}

Результаты выполнения:

**Среднее время выполнения:** 0.00367 секунды

И снова функция scanf\_s() оказалась быстрее.

На основании проведённых исследований построим график зависимости времени работы от типа данных и выбранного ввода.



По результатам проведённых исследований можно сделать вывод, что функция scanf\_s() работает быстрее, чем поток cin.

**3.Сравнение функции printf() и потока cout**

**3.1 Сравнение скорости вывода целых чисел**

Для данной проверки будем выводить 1 целое число 200000 раз. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cout:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a = 1;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

cout << a << “ “;

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:



**Среднее время выполнения:** 0.27033 секунды

**Функция printf():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a = 1;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

printf(“%d “, a);

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(“time = %.5f”, duration);

}

Результаты выполнения:



**Среднее время выполнения:** 0.071 секунды

В данном сравнении видно, что функция printf() быстрее.

**3.2 Сравнение скорости вывода вещественных чисел**

Для данной проверки будем выводить 1 вещественное число 200000 раз. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cout:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 13.13;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

cout << a << “ “;

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:



**Среднее время выполнения:** 0.607 секунды

**Функция printf():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a = 13.13;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

printf(“%.2lf “, a);

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:



**Среднее время выполнения:** 0.24167 секунды

В данном сравнении снова видно, что функция printf() быстрее.

**3.3 Сравнение скорости вывода символов**

Для данной проверки будем выводить 1 символ 200000 раз. Для объективности результата будем делать 3 попытки и находить среднее время выполнения.

**Поток cout:**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char a = ‘Б’;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

cout << a << “ “;

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(«time = %.5f», duration);

}

Результаты выполнения:



**Среднее время выполнения:** 0.06633 секунды

**Функция printf():**

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <fstream>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

char a = ‘Б’;

clock\_t start = clock();

for (int i = 0; i < 200000; i++) {

printf(“%c “, a);

}

clock\_t end = clock();

double duration = (float)(end – start) / CLOCKS\_PER\_SEC;

printf(“time = %.5f”, duration);

}

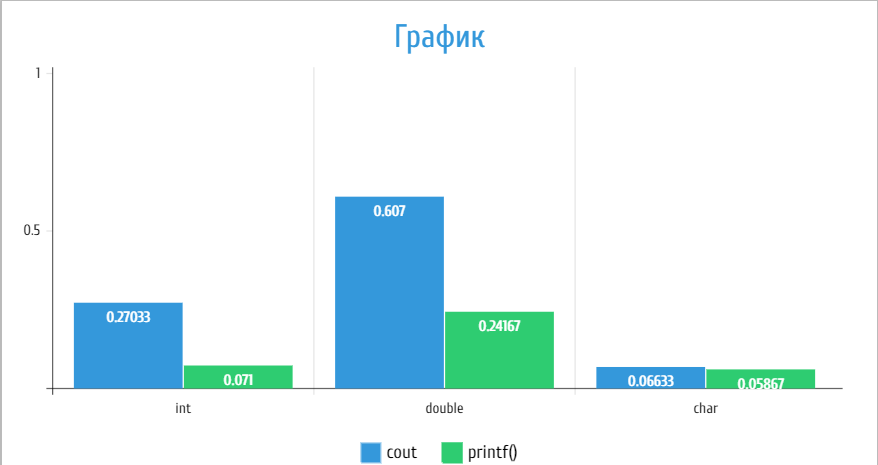
Результаты выполнения:



**Среднее время выполнения:** 0.05867 секунды

В данном сравнении функция printf() оказалась немного быстрее.

На основании проведённых исследований построим график зависимости времени работы от типа данных и выбранного вывода.



По результатам проведённых исследований можно сделать вывод, что функция printf() работает быстрее, чем поток cout.

**Выводы**

Проведённые исследования показали, что функции printf() и scanf\_s() форматированного вывода работают быстрее, чем потоки cout и cin.