Сравнение скорости работы std::cin и scanf.

Санду Р.А.

В качестве данных для тестирования была выбрана последовательность, i-ый член которой является i-ым числом Фибоначчи по модулю 10^9+7, делённым на 10^6. Для генерации был использован следующий код (`generate.cpp`):

```
#include <cstdio>
#include <cstdlib>

int main(int argc, const char* argw[])
{
  if (argc != 3) return 1;

  int count = 0;
  count = atoi(argw[2]);


  FILE* out = fopen(argw[1], "w");

  unsigned long long a = 0;
  unsigned long long b = 1;
  for (int i = 0; i < count; i++)
  {
    fprintf(out, "%lf ", b / 1000000.0);
    unsigned long long n = (a + b) % 1000000007;
    a = b;
    b = n;
  }

  fclose(out);

  return 0;
}
```

Для проведения тестов была использованна следующая программа (`clock.cpp`):

```
#include <cstdio>
#include <iostream>
#include <fstream>
#include <ctime>
#include <functional>


void time_function(std::function<void(void)> f)
{
  int start = 0;
  int end = 0;

  start = clock();
  f();
  end = clock();

  std::cout << "Function took ";
  std::cout << (end - start);
  std::cout << " clocks (";
  std::cout << ((end - start) / (CLOCKS_PER_SEC / 1000.0));
  std::cout << "ms) to finish." << std::endl;
}

int main(int argc, const char* argw[])
{
  if (argc != 3) return 1;

  int count = 0;
  count = atoi(argw[2]);
  const char* file = argw[1];

  //No more copy-pasta :)

  std::function<void(std::function<void(double&)>)> reader =
    [&count, &file](std::function<void(double&)> read_func)
    {
      freopen(file, "r", stdin);
      double input = 0;
      for (int i = 0; i < count; i++)
      {
        read_func(input);
      }
```

```
    };

  std::cout << "Testing std::cin: ";
  time_function(std::bind(reader,
    [](double& input)
    {
      std::cin >> input;
    }));

  std::cout << "Testing scanf: ";
  time_function(std::bind(reader,
    [](double& input)
    {
      scanf("%lf", &input);
    }));



  return 0;
}
```

А также для удобства был написан следующий скрипт, делающий всё и сразу `test.sh`:

```
#!/bin/bash
g++ -Wall -std=c++14 -O0 "./clock.cpp" -o "./clock"
g++ -Wall -std=c++14 -O0 "./generate.cpp" -o "./generate"

for i in `seq 4 7`;
do
  echo "Testing for 10^"$i
  ./generate "output"$1 $((10**$i))
  ./clock "output"$1 $((10**$i))
  echo
done
```

С помощью программы-генератора были сгенерированны 4 файла с соответственно 10^4, 10^5, 10^6 и 10^7 числами, каждый из которых был по-отдельности считан программой-тестом, в результате чего были получены следующие данные:

Кол-во чисел	std::cin	scanf
10^4	46.875	31.25
10^5	437.5	125.0
10^6	3968.75	1421.88
10^7	41812.5	14703.1
Время приведено в миллисекундах.		

Из этих данных можно сделать очевидный вывод: std::cin работает намного медленнее, нежели scanf. Но на самом деле, если использовать std:ifstream и fscanf, внезапно окажется, что потоки C++ работают быстрее (этот факт тоже был проверен). Вывод номер два: не использовать freopen для считывания файлов.