Lab01 - mnożenie przez 2

Mateusz Krawczuk, nr indeksu 209147

Wygenerowano przez Doxygen 1.8.6

N, 15 mar 2015 20:19:16

Część I

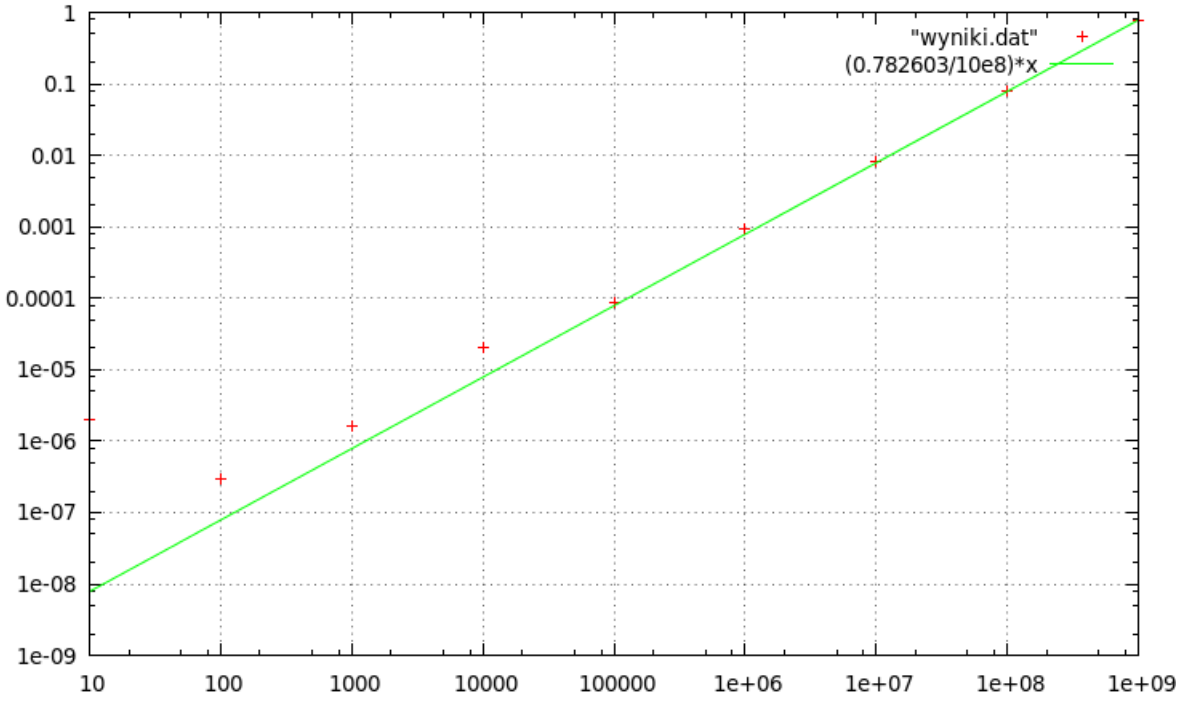
Streszczenie

Niniejszy dokument zawiera wyniki pomiaru czasu, którego potrzebował mój komputer na wykonanie operacji mnożenia przez 2 na zestawach danych o długościach od 1 do 10e8 elementów. Zawiera także dokumentację kodu programu użytego do wykonania tego badania.

Część II

Sprawozdanie

Obliczenia wykonano na 64-bitowym procesorze AMD Athlon X2. Wykres przedstawia zależnośćczasu wykonywania operacji od długości ciągu danych. Został wygenerowany za pomocą Gnuplota. Podziałki na obydwu osiach są w skali logarytmicznej. Na wykresie umieszczono także prostą y = (0.7803/10e8)x. Dzięki temu lepiej widać, że jest to zależność mocno zbliżona do liniowej - można na tej podstawie domniemywać, że złożoność obliczeniowa tej operacji jest O(n). Warto zwrócić uwagę na długość wykonywania operacji na jednym elemencie danych - przewyższa on o rząd wielkości czas wykonywania na dziesięciu elementach.



Część III

Dokumentacja kodu

1 Dokumentacja pliku benchmark.h

Deklaracje funkcji związanych z analizą prędkości operacji.

```
#include "operacja.h"
#include "ustawienia.h"
#include <fstream>
#include <iostream>
#include <vector>
#include <ctime>
#include <cstdlib>
#include <cmath>
```

Funkcje

- void Benchmark ()

  Główna funkcja programu.
- int licz_dekady (int dlugosc_ciagu)

  Funkcja pomocnicza do określania ilości dekad.

1.1 Opis szczegółowy

Plik zawiera deklaracje funkcji Benchmark() oraz licz_dekady(). Funkcja Benchmark() jest główną funkcją programu benchmarkującego, natomiast funkcja licz_dekady() jest funkcją pomocniczą wywoływaną w funkcji Benchmark().

Definicja w pliku benchmark.h.

1.2 Dokumentacja funkcji

1.2.1 void Benchmark ( )

Funkcja wywoływana w funkcji main(). Nie przyjmuje argumentów ani nie zwraca wartości. Funkcja otwiera plik, który powinien zawierać ciąg danych, na których przeprowadzona zostanie operacja. Otwiera także plik, do którego zapisane mają zostać wyniki czasowe operacji. Nazwy plików określone są w nagłówku "ustawienia.h". Z pliku wejściowego wczytuje do obiektu klasy std::vector ciąg danych. Do zmiennej typu clock_t zostaje zapisany czas procesora od rozpoczęcia procesu. Do funkcji operacja() zostaje przekazana referencja do obiektu zawierającego ciąg danych. Po wyjściu z funkcji operacja() do zmiennej typu clock_t() zostaje tym razem zapisana różnica czasów procesora pomiędzy rozpoczęciem procesu a zakończeniem pracy funkcji operacja(). Czas ten przeliczany jest na sekundy i zapisany do pliku wyjściowego. W "ustawienia.h" określona jest ilość powtórzeń pomiaru. Pomiary zaczynają się od operacji na jednej wartości i kończą na ciąu danych o długości największej potęgi dziesiątki. Pomiary powtarzane są co dekadę. Informację o największej potędze dziesiątki dostarcza funkcja licz_dekady().

Definicja w linii 4 pliku benchmark.cpp.

1.2.2 int licz_dekady (int *dlugosc_ciagu*)

Przyjmuje jako argument liczbę całkowitą reprezentującą długość ciągu danych i zaokrągla ją w dół do najbliższej potęgi liczby 10.

Parametry

in	*dlugosc_ciagu*	Długość ciągu danych.

Zwraca

Zwraca zaokrąglenie w dół do najbliższej potęgi dziesiątki wartości dlugosc_ciagu.

Definicja w linii 61 pliku benchmark.cpp.

2 Dokumentacja pliku operacja.h

Plik zawiera deklarację funkcji operacja().

```
#include <vector>
#include <cassert>
```

Funkcje

- void operacja (std::vector< int > &wektor, unsigned int ilosc_operacji)
  Wykonuje operację na wczytanym ciągu danych.

2.1 Dokumentacja funkcji

2.1.1 void operacja (std::vector< int > & *wektor,* unsigned int *ilosc_operacji*)

Funkcja odpowiada za wykonanie porządanej operacji na ciągu danych.

Parametry

in	*wektor*	Referencja do wczytanego ciąg danych.
in	*ilosc_operacji*	Określa na ilu pierwszych elementach wektora wektor ma zostać wykonana operacja.

Definicja w linii 10 pliku operacja.cpp.

3 Dokumentacja pliku ustawienia.h

Plik zawiera stałe preprocesora związane z generowaniem, obróbką i pomiarem właściwości danych.

Definicje

- #define ILOSC 10e7
- #define KRES_GORNY INT_MAX/2 - 1
- #define KRES_DOLNY 0
- #define NAZWA_PLIKU_WE "dane.dat"
- #define NAZWA_PLIKU_WY "wyniki.dat"
- #define ILOSC_POWTORZEN 10

3.1 Opis szczegółowy

Makra zawarte w tym pliku służą do sterowania pracą programu generującego dane oraz programu, który te dane przetwarza. Służy też synchronizacji pomiędzy tymi dwoma programami poprzez ujednolicenie nazwy pliku zawierającego dane.

Definicja w pliku ustawienia.h.

3.2 Dokumentacja definicji

3.2.1 #define ILOSC 10e7

Określa dłuość ciągu danych stworzonych przez program generuj.

Definicja w linii 14 pliku ustawienia.h.

3.2.2 #define ILOSC_POWTORZEN 10

Tyle razy zostanie powtórzoy pomiar dla jednego zestawu danych.

Definicja w linii 24 pliku ustawienia.h.

3.2.3 #define KRES_DOLNY 0

Określa najmniejszą możliwą liczbę do wygenerowania.

Definicja w linii 18 pliku ustawienia.h.

3.2.4 #define KRES_GORNY INT_MAX/2 - 1

Określa największą możliwą liczbę do wygenerowania.

Definicja w linii 16 pliku ustawienia.h.

3.2.5 #define NAZWA_PLIKU_WE "dane.dat"

Określa jak nazywa się plik wygenerowany przez program 'generuj', jednocześnie pliku o tej nazwie poszukuje program 'program' w katalogu, w którym został uruchomiony.

Definicja w linii 20 pliku ustawienia.h.

3.2.6 #define NAZWA_PLIKU_WY "wyniki.dat"

Tak nazywać się będzie wygenerowany przez program 'program' plik z wynikami przeprowadzonych pomiarów.

Definicja w linii 22 pliku ustawienia.h.