Київський національний

університет

імені Т.Шевченка

**Звіт**

до лабораторної роботи №1

на тему:

«***Визначення швидкодії***

***обчислювальної системи»***

***Студента другого курсу***

***Групи К-23***

***Факультету комп’ютерних наук***

***та кібернетики***

***Тарасюка Олександра***

***Київ-2022***

**Постановка лабораторної роботи**

* **Мета**

Розробка програми, яка вимірює кількість виконуваних базових операцій (команд) за секунду конкретною ОбСист (комп’ютер + ОС + Система програмування).

* **Деякі відомості про виконання**

Робота виконувалася мовою програмування С++ у середовищі розробки Code::Blocks 20.03

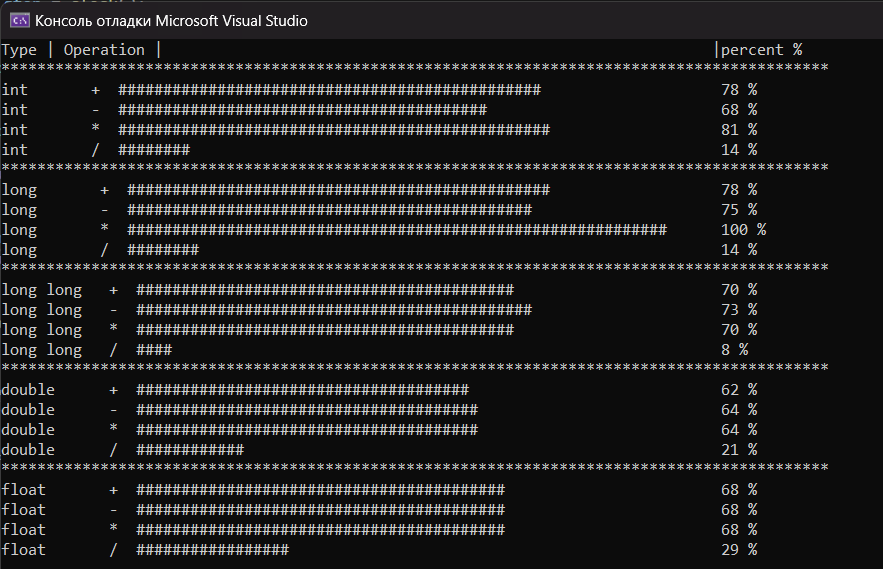
Для дослідження обчислень були вибрані наступні типи даних: Int, long, double, float, char; та наступні типи операцій: “ + “, “ – “, “ \* “, “ / “.

Вивід являє собою “таблицю”, у якій вказано операції, типи, шкала заповнення значення величини максимально швидкої операції та типу, і інших відносно цього значення, а також відсоток відносно найшвидшої операції.

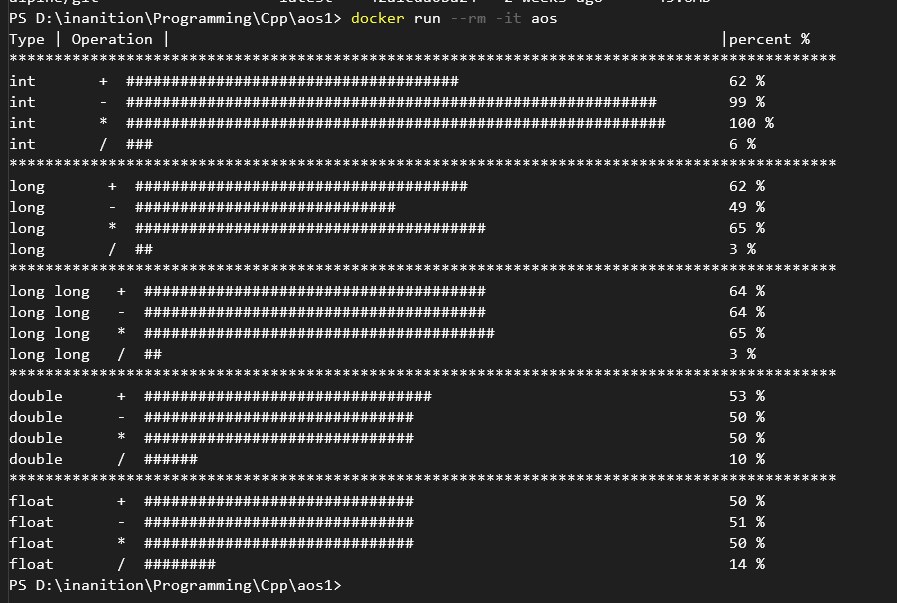
Для проведення вимірів було використано наступну ОбСист: **Процессор:** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz

**ОС:**  Windows 11 x64

**Результат з Windows 11**



**Результат з Docker**

****

**Висновки**

* Отже, у моєму випадку ділення було найшвидшим оператором, додавання та віднімання займали майже однакову кількість часу, а множення – найбільше.
* На результати ОбСист впливають такі характеристики, як об’єм оперативної пам’яті, тип материнської плати, параметри відео підсистеми і т.д.

**Код програми**

#include <iostream>

#include <time.h>

#include <vector>

#define TIMER 10000000

using namespace std;

template <typename T>

double getTime(char oper, T a, T b)

{

clock\_t start, stop;

T c;

if (oper == '+') {

start = clock();

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

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

c = a + b;

}

stop = clock();

}

else if (oper == '-') {

start = clock();

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

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

c = a - b;

}

stop = clock();

}

else if (oper == '\*') {

start = clock();

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

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

c = a \* b;

}

stop = clock();

}

else {

start = clock();

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

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

c = a / b;

}

stop = clock();

}

clock\_t loop\_start = clock();

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

}

clock\_t loop\_stop = clock();

return (double)(stop - start + loop\_start - loop\_stop) / CLOCKS\_PER\_SEC;

}

long long calc(double time) {

return (long long)((TIMER \* 10) / time);

}

void Draw(int height, long long max\_val, long long val) {

int s\_count = (int)((double)val / (double)max\_val \* height);

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

if (i < s\_count) cout << "#"; else cout << " ";

}

cout << "\t" << (int)((double)val / (double)max\_val \* 100) << " %\n";

}

void Graph(vector<string> name, vector<long long> time) {

char operation[4] = { '+','-','\*','/' };

int Draw\_height = 60;

long long mx = -1;

for (int i = 0; i < time.size(); i++) {

mx = max(mx, time[i]);

}

int t = 0;

for (int i = 0; i < name.size(); i++) {

if (i % 4 == 0) cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << name[i] << operation[i % 4];

for (int i = 0; i < (12 - name[i].size()); i++) {

cout << " ";

}

Draw(Draw\_height, mx, time[i]);

}

}

int main() {

vector<string> name;

vector<long long> time;

vector <pair <string, long long>> res;

int a1 = 345435, b1 = 346436;

long a2 = 234235346, b2 = 346436346;

long long a3 = 674574357456, b3 = 457457474357326;

double a4 = 5453.346346, b4 = 534.346436436;

float a5 = 546.654645, b5 = 647.57567;

char operation[4] = { '+','-','\*','/' };

string operations = "+-\*/";

cout << "Type | Operation | |percent %\n";

for (int i = 0; i < 4; i++) {

name.push\_back("int ");

time.push\_back(calc(getTime(operation[i], a1, b1)));

}

for (int i = 0; i < 4; i++) {

name.push\_back("long ");

time.push\_back(calc(getTime(operation[i], a2, b2)));

}

for (int i = 0; i < 4; i++) {

name.push\_back("long long ");

time.push\_back(calc(getTime(operation[i], a3, b3)));

}

for (int i = 0; i < 4; i++) {

name.push\_back("double ");

time.push\_back(calc(getTime(operation[i], a4, b4)));

}

for (int i = 0; i < 4; i++) {

name.push\_back("float ");

time.push\_back(calc(getTime(operation[i], a5, b5)));

}

Graph(name, time);

return 0;

}

**Код докерфайлу**

FROM gcc:latest

COPY . ./aos

WORKDIR /aos

RUN g++ -o app aos1.cpp

CMD ["./app"]