

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 C++ 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

Консоль отладки Microsoft Visual Studio

Type	Operation	percent %

int	+ #####	78 %
int	- #####	68 %
int	* #####	81 %
int	/ #####	14 %

long	+ #####	78 %
long	- #####	75 %
long	* #####	100 %
long	/ #####	14 %

long long	+ #####	70 %
long long	- #####	73 %
long long	* #####	70 %
long long	/ ###	8 %

double	+ #####	62 %
double	- #####	64 %
double	* #####	64 %
double	/ #####	21 %

float	+ #####	68 %
float	- #####	68 %
float	* #####	68 %
float	/ #####	29 %

Результат з Docker

PS D:\inanition\Programming\Cpp\aos1> docker run --rm -it aos

Type	Operation	percent %

int	+ #####	62 %
int	- #####	99 %
int	* #####	100 %
int	/ ###	6 %

long	+ #####	62 %
long	- #####	49 %
long	* #####	65 %
long	/ ##	3 %

long long	+ #####	64 %
long long	- #####	64 %
long long	* #####	65 %
long long	/ ##	3 %

double	+ #####	53 %
double	- #####	50 %
double	* #####	50 %
double	/ #####	10 %

float	+ #####	50 %
float	- #####	51 %
float	* #####	50 %
float	/ #####	14 %

PS D:\inanition\Programming\Cpp\aos1>

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

[illegible]

```

    }
    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"]