

Київський національний університет імені
Т.Шевченка

Звіт

Про виконання лабораторної роботи з дисципліни:
архітектура обчислювальних систем.

На тему: “Визначення швидкодії обчислювальної
системи”

Мета: розробити програму, яка вимірює кількість виконуваних базових операцій (команд) за секунду конкретною обчислювальною системою (комп'ютер + ОС + Система програмування).

Нижче подані основні принципи написання, не вдаючись до детального розбору коду.

- Для роботи було обрано компільовану мову програмування C++, повний код програми міститься в єдиному файлі. Після запуску програми користувач виводиться таблиця із певною статистикою по часу виконання певних арифметичних операцій.
- Перевірки робляться для 6 типів даних: int, long double, long long, char, float, double; та для чотирьох операцій «+», «-», «*», «/».
- Дублювання коду зведено до мінімуму при, якому зберігається його читабельність.
- Для підтвердження достовірності результатів були проведені повторні тестування у Docker.

Результати на ноутбуці (ос. Вiндовс, компiлятор gmw)

+	int	5.125614e+009	XXXXXXXXXXXXXXXXXXXXXXXXX	99%
-	int	5.147615e+009	XXXXXXXXXXXXXXXXXXXXXXXXX	100%
*	int	3.848633e+009	XXXXXXXXXXXXXXXXXXXXX	74%
/	int	1.604847e+009	XXXXXXX	31%
+	float	1.861395e+009	XXXXXXXXXX	36%
-	float	1.765000e+009	XXXXXXXXXX	34%
*	float	5.302508e+008	XX	10%
/	float	3.983672e+008	X	7%
+	double	1.003054e+009	XXXXX	19%
-	double	1.051978e+009	XXXXXX	20%
*	double	2.342284e+008	X	4%
/	double	1.553300e+007		0%
+	long double	4.525629e+008	XX	8%
-	long double	4.304456e+008	XX	8%
*	long double	7.660930e+007		1%
/	long double	1.660930e+007		0%
+	long long	1.591023e+009	XXXXXXXXX	30%
-	long long	1.541165e+009	XXXXXXXXX	29%
*	long long	8.810920e+007		1%
/	long long	6.318728e+008	XXX	12%
+	char	4.925635e+009	XXXXXXXXXXXXXXXXXXXXXXXXX	95%
-	char	4.695615e+009	XXXXXXXXXXXXXXXXXXXXXXXXX	91%
*	char	3.642633e+009	XXXXXXXXXXXXXXXXXXXXX	70%
/	char	1.384947e+009	XXXXXXX	26%

Результати на Docker за образ обраний образ gcc

+	int	5.319614e+09	XXXXXXXXXXXXXXXXXXXXXXXXX	99%
-	int	5.340115e+09	XXXXXXXXXXXXXXXXXXXXXXXXX	100%
*	int	4.041133e+09	XXXXXXXXXXXXXXXXXXXXX	75%
/	int	1.797347e+09	XXXXXXX	33%
+	float	2.053895e+09	XXXXXXXXXX	38%
-	float	1.957500e+09	XXXXXXXXXX	36%
*	float	7.227508e+08	XXX	13%
/	float	5.908672e+08	XX	11%
+	double	1.195554e+09	XXXXX	22%
-	double	1.244478e+09	XXXXXX	23%
*	double	4.267284e+08	X	7%
/	double	1.769670e+08		3%
+	long double	6.450629e+08	XXX	12%
-	long double	6.229456e+08	XX	11%
*	long double	1.158907e+08		2%
/	long double	1.758907e+08		3%
+	long long	1.783523e+09	XXXXXXXXX	33%
-	long long	1.733665e+09	XXXXXXXXX	32%
*	long long	2.806092e+08	X	5%
/	long long	8.243728e+08	XXX	15%
+	char	5.118135e+09	XXXXXXXXXXXXXXXXXXXXXXXXX	95%
-	char	4.888115e+09	XXXXXXXXXXXXXXXXXXXXXXXXX	91%
*	char	3.835133e+09	XXXXXXXXXXXXXXXXXXXXX	71%
/	char	1.577447e+09	XXXXXXX	29%

Висновок: будь-яку обчислювальну систему можна досить точно оцінити за допомогою подібних тестів, проте на результати тестування буде впливати компілятор та операційна система обчислювальної системи, тож за допомогою цих тестів слід порівнювати лише обчислювальні системи у однакових умовах.

Додаток

Код програми

```
#include <iostream>
#include <chrono>
#include <vector>
#include <iomanip>
#include <cmath>
#include <string>
#pragma GCC optimize("O0")
```

```
using namespace std;
```

```
double maximum(vector<double> a){
    double tmp = a[0];
    for (int i = 1; i < a.size(); i++){
        if (tmp < a[i]){tmp = a[i];}
    }
    return tmp;
}
```

```
template<typename T>
T plusic(T a, T b){return a + b;}
```

```
template<typename T>
T minusic(T a, T b){return a - b;}
```

```
template<typename T>
T mult(T a, T b){return a * b;}
```

```
template<typename T>
T divid(T a, T b){return a / (!b + b);}
```

```
template<typename T>
T nothing(T a, T b){return a;}
```

```
template<typename T>
T divid_helper(T a, T b){return (!b + b);}
```

```
long long QUON_OF_ITER = pow(10, 6);
```

```
template<typename T>
double iter(T (*foo)(T, T), T a, T b, T k){
    auto c = foo(a, b);
    auto begin = std::chrono::steady_clock::now();
```

```

    for (long long i = 0; i < QUON_OF_ITER; i++){
        b = foo(a, b);
        a = foo(a, b);
        b = foo(a, b);
        a = foo(a, b);
        b = foo(a, b);
        a = foo(a, b);
        b = foo(a, b);
        a = foo(a, b);
        b = foo(a, b);
        a = foo(a, b);
    }

    auto end = std::chrono::steady_clock::now();

    auto dur = std::chrono::duration_cast<std::chrono::milliseconds>(end -
begin);

    return (double)dur.count() / 100000;
}

template<typename T>
double result(char c, T a, T b, T k){
    T (*i)(T, T);
    if (c == '+'){i = plusic;}
    else if (c == '-'){i = minusic;}
    else if (c == '*'){i = mult;}
    else {i = divid;}
    return iter(i, a, b, k);
}

void output(vector<double>a){

    cout << "\n\n";

    cout << scientific;

    vector<string> types = {"int", "float", "double", "long double", "long
long", "char"};
    vector<string>probel = {"", " ", " ", " ", " ", " ", " ", " ", " ", "
"};
    vector<char> opers = {'+', '-', '*', '/'};
    double max = maximum(a);

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

        cout << setw(5) << "|" << opers[i % 4] << "|"
            << setw(9) << "|" << types[i / 4] << "|"
            << probel[i / 4]

```

```

        << setw(1) << "|" << a[i] << "|"
        << "  |" << string(int(25 * abs(a[i]) / max), 'X')
        << string(25 - int(25 * a[i] / max), ' ') << "|" << setw(10)
<< " | " << int(100 * a[i] / max) << "% |";

        cout << "\n";
        if (i % 4 == 3){cout << "\n";}
    }
}

```

```

int main(){

```

```

    vector<double> a, l, d;

    d.push_back(iter(divid_helper, (int)214, (int)2112, (int)10000));
    d.push_back(iter(divid_helper, (float)214.2413, (float)2112.2634,
(float)1000.42121));
    d.push_back(iter(divid_helper, (double)214.2413, (double)2112.2634,
(double)1000.42121));
    d.push_back(iter(divid_helper, (long double)21454326.32659, (long
double)21123423.2352135, (long double)10000000.3245214126));
    d.push_back(iter(divid_helper, (long long)2145432623, (long
long)2112342323, (long long)1000000000));
    d.push_back(iter(divid_helper, (char)'L', (char)'o', (char)'-'));

    l.push_back(iter(nothing, (int)214, (int)2112, (int)10000));
    l.push_back(iter(nothing, (float)214.2413, (float)2112.2634,
(float)1000.42121));
    l.push_back(iter(nothing, (double)214.2413, (double)2112.2634,
(double)1000.42121));
    l.push_back(iter(nothing, (long double)21454326.32659, (long
double)21123423.2352135, (long double)10000000.3245214126));
    l.push_back(iter(nothing, (long long)2145432623, (long
long)2112342323, (long long)1000000000));
    l.push_back(iter(nothing, (char)'L', (char)'o', (char)'-'));

    //int
    a. push_back( QUON_OF_ITER / (result('+', (int)214, (int)2112,
(int)10000) - l[0]));
    a. push_back( QUON_OF_ITER / (result('-', (int)214, (int)2112,
(int)10000) - l[0]));
    a. push_back( QUON_OF_ITER / (result('*', (int)214, (int)2112,
(int)10000) - l[0]));
    a. push_back( QUON_OF_ITER / (result('/', (int)214, (int)2112,
(int)10000) - d[0]));

    //float
    a. push_back( QUON_OF_ITER / (result('+', (float)214.2413,
(float)2112.2634, (float)1000.42121) - l[1]));

```



```

a. push_back( QUON_OF_ITER / (result('-', (float)214.2413,
(float)2112.2634, (float)1000.42121) - l[1]));
a. push_back( QUON_OF_ITER / (result('*', (float)214.2413,
(float)2112.2634, (float)1000.42121) - l[1]));
a. push_back( QUON_OF_ITER / (result('/', (float)214.2413,
(float)2112.2634, (float)1000.42121) - d[1]));

//double
a. push_back( QUON_OF_ITER / (result('+', (double)214.2413,
(double)2112.2634, (double)1000.42121) - l[2]));
a. push_back( QUON_OF_ITER / (result('-', (double)214.2413,
(double)2112.2634, (double)1000.42121) - l[2]));
a. push_back( QUON_OF_ITER / (result('*', (double)214.2413,
(double)2112.2634, (double)1000.42121) - l[2]));
a. push_back( QUON_OF_ITER / (result('/', (double)214.2413,
(double)2112.2634, (double)1000.42121) - d[2]));

//long double
a. push_back( QUON_OF_ITER / (result('+', (long
double)21454326.32659, (long double)21123423.2352135, (long
double)10000000.3245214126) - l[3]));
a. push_back( QUON_OF_ITER / (result('-', (long
double)21454326.32659, (long double)21123423.2352135, (long
double)10000000.3245214126) - l[3]));
a. push_back( QUON_OF_ITER / (result('*', (long
double)21454326.32659, (long double)21123423.2352135, (long
double)10000000.3245214126) - l[3]));
a. push_back( QUON_OF_ITER / (result('/', (long
double)21454326.32659, (long double)21123423.2352135, (long
double)10000000.3245214126) - d[3]));

//long long
a. push_back( QUON_OF_ITER / (result('+', (long long)2145432623,
(long long)2112342323, (long long)1000000000) - l[4]));
a. push_back( QUON_OF_ITER / (result('-', (long long)2145432623,
(long long)2112342323, (long long)1000000000) - l[4]));
a. push_back( QUON_OF_ITER / (result('*', (long long)2145432623,
(long long)2112342323, (long long)1000000000) - l[4]));
a. push_back( QUON_OF_ITER / (result('/', (long long)2145432623,
(long long)2112342323, (long long)1000000000) - d[4]));

//char
a. push_back( QUON_OF_ITER / (result('+', (char)'L', (char)'o',
(char)'-') - l[5]));
a. push_back( QUON_OF_ITER / (result('-', (char)'L', (char)'o',
(char)'-') - l[5]));
a. push_back( QUON_OF_ITER / (result('*', (char)'L', (char)'o',
(char)'-') - l[5]));
a. push_back( QUON_OF_ITER / (result('/', (char)'L', (char)'o',
(char)'-') - d[5]));

```

output (a) ;

```
    return 0;  
}
```