­­МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №9 дисциплины

«Разработка программных систем»

Вариант 2

Выполнил студент группы ИВТб-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А. /

Киров 2023

1. Постановка задачи

Для выполнения лабораторной работы необходимо решить следующие

задачи.

 По результатам лабораторной работы No1 выбрать один из наиболее

требовательных методов.

 Реализовать данный метод на языке С/C++.

 Оформить реализацию в виде подключаемого модуля.

 Провести сравнения реализаций метода на Python и на C/C++ для

набора подготовленных данных.

2. Ход работы

Для реализации на языке С++ был выбран метод умножения двух длинных чисел «multiply\_num». Разработанный файл является динамической библиотекой.

2.1 Динамическая библиотека с методом умножения

extern "C" \_\_declspec(dllexport) const char\* multiply\_num(char\* str1, int len1, char\* str2, int len2) {

mpz\_t num1, num2, result;

mpz\_init(num1);

mpz\_init(num2);

mpz\_init(result);

string string1(str1, len1);

string string2(str2, len2);

const char\* number1 = string1.c\_str();

const char\* number2 = string2.c\_str();

mpz\_set\_str(num1, number1, 10);

mpz\_set\_str(num2, number2, 10);

auto start = std::chrono::high\_resolution\_clock::now();

// Умножение чисел

mpz\_mul(result, num1, num2);

auto end = std::chrono::high\_resolution\_clock::now();

size\_t bufferSize = mpz\_sizeinbase(result, 10) + 2;

char\* charArray = new char[bufferSize];

mpz\_get\_str(charArray, 10, result);

string arr(charArray);

std::chrono::duration<double> duration = end - start;

std::cout << "Время выполнения: " << duration.count() << " сек" << std::endl;

// Освобождение памяти

mpz\_clear(num1);

mpz\_clear(num2);

mpz\_clear(result);

const char\* cstr = arr.c\_str();

return cstr;

}

2.2 Модуль main.py (фрагмент, в котором происходит вызов функции из библиотеки)

lib = ctypes.CDLL('./Module.dll')  
lib.multiply\_num.restype = ctypes.c\_char\_p  
lib.multiply\_num.argtypes = [ctypes.c\_char\_p, ctypes.c\_int, ctypes.c\_char\_p, ctypes.c\_int]

if combo.get() == "Умножение":  
 main\_numerator\_str = str(main.numerator)  
 sec\_numerator\_str = str(sec.numerator)  
 mn\_bytes = main\_numerator\_str.encode('UTF-8')  
 sn\_bytes = sec\_numerator\_str.encode('UTF-8')  
 start = time.time()  
 num = lib.multiply\_num(mn\_bytes, len(mn\_bytes), sn\_bytes, len(sn\_bytes))  
 end = time.time() - start  
 print("Time py = ", end)  
 num\_str = num.decode('utf-8')  
 den = BigFraction.multiply\_den(main.denominator, sec.denominator)  
 res = fractions.Fraction(int(num\_str), den)

3. Сравнительные характеристики

Сравнительные характеристики на языках Python и C++ представлены в таблице 1.

Таблица 1 – Время выполнения функции

|  |  |  |
| --- | --- | --- |
| Каждый символ в числе равен «1» | Время на Python, сек | Время на С++, сек |
| Число длиною в  10000 цифр | 0.0009982 | 0.0001648 |
| Число длиною в  100000 цифр | 0.1000108 | 0.001032 |
| Число длиною в  1000000 цифр | 9.1272664 | 0.0164055 |
| Число длиною в  10000000 цифр | 203.2810427 | 0.230577 |

Вывод

В ходе выполнения лабораторной работы была написана функция, перемножающая два длинных числа. Данная функция была написана как часть динамической библиотеки, которая подключалась к проекту. Время выполнения метода, написанного на языке С++ оказался значительно быстрее.