

ЛР 2: [C++ UNIX]: C++ BUILD / IF / LOOP, PYTHON

Готов Константин
Z33431

Цель работы

Познакомить студента с принципами компиляции исходного кода. Составить программу с использованием циклов, условий и функций. Сравнить быстродействие между C++ и Python. Ознакомление с типами данных.

1. [C++ EXPRESSION] Создать и скомпилировать программу на C++

1.1 Создание программы

```
1 #include <iostream>
2 #include <cmath>
3 #include <chrono>
4 using namespace std;
5 using namespace std::chrono;
6
7
8 double f (double x) {
9     return pow(x, 2) - pow(x, 2) + 4 * x - 5 * x + x + x;
10 }
11
12
13 int main (void) {
14     int n;
15     double res = 1;
16
17     while (true) {
18         cin >> n;
19
20         while (cin.fail()) {
21             cout << "Error with input's type" << endl;
22             return 30;
23         }
24
25         auto start = high_resolution_clock::now();
26
27         for (int i = 0; i < n; i++)
28             res += f(res);
29         cout << res << endl;
30
31         auto stop = high_resolution_clock::now();
32         auto duration = duration_cast<microseconds>(stop - start);
33
34         cout << "Time taken by function: "
35              << duration.count() << " microseconds" << endl;
36
37     }
```

```

38     return 30;
39 }

```

Листинг 1: main.cpp

Также были дополнительно созданы два bash скрипта для удобной компиляции и запуска программы:

```

1 g++ -o build/main.out src/main.cpp

```

Листинг 2: build.sh

```

1 ./build/main.out

```

Листинг 3: run.sh

1.2 Компиляция и запуск

Запуск производился из корневой директории лабораторной работы.

```

1 sudo bash compile.sh
2 sudo bash run.sh

```

2 Создать и скомпилировать программу на Python 3

```

1 import time
2
3
4 def f(x):
5     return x**2 - x**2 + 4 * x - 5 * x + x + x
6
7
8 if __name__ == "__main__":
9     while True:
10         n = input()
11
12         try:
13             n = int(n)
14         except (TypeError, ValueError) as e:
15             print("Error in input's type")
16             exit(30)
17
18         x = 1
19         start_time = time.time()
20
21         for _ in range(n):
22             x += f(x)
23
24         print(x)
25         print("--- %s seconds ---" % (time.time() - start_time))

```

Листинг 4: main.py

3 Бенчмарки

При запуске на $n = 10000$. Код на C++ выполнялся за 594 микросекунды, а код на Python за 0.29 секунды! Таким образом, код на Python выполнялся на порядки дольше.

4 Выводы

В ходе лабораторной работы были созданы простые программы на C++ и Python и проверена скорость их работы. Как и ожидалось, C++ работает быстрее из-за того, что код предварительно компилировался.