

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

Цель

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.

Задача

- 1. [C++ EXPRESSION] Создать и скомпилировать программу на C++**
Результат сборки (компиляции) сохранять в папку build. Папку build сделать игнорируемой для GIT. Программа должна получать на вход число – это количество итераций для выполнения расчета. В рамках итерации выполнять следующее вычисление: $x^2 - x^2 + x^4 - x^5 + x + x$. Вычисление выполнять в виде отдельной от main функции, которая будет вызвана циклически из main. Фиксировать время выполнения программы, затрачиваемое на расчет выражения n раз (n задается в консоли перед вычислением). Предусмотреть дополнительный цикл на повторную итерацию запуска программы вычислений. Если было введено не число, то завершить выполнение программы.
- 2. [PYTHON EXPRESSION] Создать и скомпилировать программу на Python 3**
Результат сборки (компиляции) сохранять в папку build. Папку build сделать игнорируемой для GIT. Программа должна получать на вход число – это количество итераций для выполнения расчета. В рамках итерации выполнять следующее вычисление: $x^2 - x^2 + x^4 - x^5 + x + x$. Вычисление выполнять в виде отдельной от main функции, которая будет вызвана циклически из main. Фиксировать время выполнения программы, затрачиваемое на расчет выражения n раз (n задается в консоли перед вычислением). Предусмотреть дополнительный цикл на повторную итерацию запуска программы вычислений. Если было введено не число, то завершить выполнение программы.
- 3. [SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку doc)**
Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

Решение

1.

```
1  #include <iostream>
2  #include <cmath>
3  #include <ctime>
4
5  using namespace std;
6
7
8  void f(int x)
9  {
10     pow(x, 2) - pow(x, 2) + x * 4 - x * 5 + x + x;
11 }
12
13
14 int main()
15 {
16     int n;
17     unsigned int start_time, end_time, d_time;
18
19     while (cin >> n)
20     {
21         cout << n << endl;
22         start_time = clock();
23
24         for (int i = 1; i <= n; i++)
25         {
26             f(n);
27         }
28
29         end_time = clock();
30         d_time = end_time - start_time;
31         cout << d_time << endl;
32     }
33 }
```

2.

```
1  import time
2
3
4  def f(x):
5      return x ** 2 - x ** 2 + x * 4 - x * 5 + x + x
6
7
8  def main():
9      n = input()
10     while (not n.isalpha()):
11         n = int(float(n))
12         start_time = time.process_time()
13         for i in range(n):
14             f(n)
15         end_time = time.process_time()
16         t = end_time - start_time
17         print(int(t*1000))
18         n = input()
19
20
21  main()
```

Заключение

В результате лабораторной работы было произведено знакомство с принципами компиляции исходного кода, ознакомление с типами данных, составлена программа с использованием циклов, условий и функций, сравнено быстродействие между C++ и Python.