Реквизиты

Жолобов Даниил Валерьевич
3 курс
z33434
2024

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. [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.

```
touch .gitignore
echo "**/build" > .gitignore
clang++ -O3 lab_02/src/main.cpp -o lab_02/build/a.out
lab_02/build/a.out 2 10000
-->
x = 2
n = 100000
x' = 2
1.30815 ms
```

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.

```
python3 lab_02/src/main.py 2 100000
-->
x = 2.0
n = 100000
x' = 2.0
45.00577400176553 ms
```

1. [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).

```
echo "git stash -u" > revert_to_last_commit.sh
sudo chmod +x ./revert_to_last_commit.sh
```

Заключение

Сравнил производительность C++ и python на простом алгоритме. Научился парсить аргументы в программе на C++. Вспомнил, как работать с библиотекой <chrono>.