

Четвёртое Домашнее задание "Архитектура вычислительных систем"

Задача: Языки программирования, Вариант 148 (8, 11)

Автор: Туракулов Исломбек Улугбекович

Условие задачи

Общие для всех альтернатив переменные * Популярность в процентах (TIOBI) — действительное * Год создания — целое

Общие для всех альтернатив функции

Частное от деления года создания на количество символов в названии (действительное число)

Базовые альтернативы

- Процедурные
- наличие, отсутствие абстрактных типов данных – булевская величина
- Объектно-ориентированные
- наследование: одинарное, множественное, интерфейса – перечислимый тип
- Функциональные
- типизация – перечислимый тип = строгая, динамическая
 - поддержка «ленивых» вычислений – булевский тип

Функционал

После размещения данных в контейнер необходимо осуществить их обработку в соответствии с вариантом задания. Обработанные данные после этого заносятся в отдельные файлы результатов.

Упорядочить элементы контейнера по возрастанию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Тестирование

Исходные данные для тестирования содержатся в каталоге `outs`.

Файл с результатами прогонов тестов `outs/reports.txt`

```
$ sh compilefile.sh
```

Процедурная (C++)

Время работы программы на разных размерах входных данных:

Количество языков программирования	Время работы, seconds	Потребляемая память, KB
7	< 0.002	~2750
100	< 0.01	~2892
1000	0.01	~3674
5000	0.15	~4622

Количество языков программирования	Время работы, seconds	Потребляемая память, KB
10000	0.96	~4877

Объектно-ориентированная (C++)

Время работы программы на разных размерах входных данных:

Количество языков программирования	Время работы, seconds	Потребляемая память, KB
7	< 0.001	~2600
100	< 0.01	~2800
1000	0.01	~3500
5000	0.14	~4150
10000	0.90	~4625

Динамическая типизация (Python)

Время работы программы на разных размерах входных данных:

Количество языков программирования	Время работы, seconds	Потребляемая память, KB
10	Source: 0.002 Sort: 0.002	~4420
100	Source: 0.005 Sort: 0.023	~5650
1000	Source: 0.038 Sort: 1.911	~8652
5000	Source: 0.069 Sort: 2.233	~10841
10000	Source: 0.423 Sort: 169.316	~20532

Разница процедурной, объектно-ориентированной, динамической типизацией с программой на ассемблере

Конечно, программу на Ассемблере писать труднее, чем программу на языке высокого уровня. Однако Ассемблер имеет и преимущества.

Во-первых, программа, написанная на языке высокого уровня, все равно транслируется в ассемблерную программу, причем весьма неоптимальным образом. То есть программа на Ассемблере практически всегда будет работать быстрее и занимать значительно меньше памяти.

Во-вторых, доступ ко многим аппаратным ресурсам можно получить только с помощью Ассемблера. Прошу подметить, ассемблерную программу можно писать в любом редакторе!

Метрики, определяющие характеристики программы:

Метрика	Значение
Общий размер исходных текстов программы	17.211 KB
Размер исполняемого файла релизной сборки (GCC, Linux)*	32.12 KB

* Версии подробнее:

```
islam@islam-VirtualBox:~$ g++ --version
g++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

islam@islam-VirtualBox:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.3 LTS
Release:        20.04
Codename:       focal

islam@islam-VirtualBox:~$ uname -a
Linux islam-VirtualBox 5.11.0-40-generic #44~20.04.2-Ubuntu SMP Tue Oct 26 18:07:44 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```