Пояснительная записка по программе

1. Общие положения

1.1 Исполнитель

ФИО: Петрова Ксения Эдуардовна   
Группа: БПИ202  
Почта: [kepetrova@edu.hse.ru](mailto:kepetrova@edu.hse.ru)

1.2 Язык разработки

Разработка осуществляется на языке программирования Nasm расширение файлов .asm.

1.3 Основные характеристики программы

1.3.1 Число заголовочных файлов: 5

1.3.2 Число файлов .mac: 1

1.3.3 Объём памяти, занимаемый решением: 37,1 KБ

1.3.4 Размер Unix Executable: 27 КБ

1.4 Описание полученного задания

Вариант 56

1.4.1 Начальное условие задачи

|  |  |  |  |
| --- | --- | --- | --- |
| Обобщенный артефакт, ис- пользуемый в задании | Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив) | Общие для всех альтернатив переменные | Общие для всех альтерна- тив функции |
| 14. Автомо- бильный транс- порт | 1. Грузовик (грузоподъем- ность кг – целое) 2. Автобус (пассажировместимость – короткое целое)  3. Легковой автомобиль (максимальная скорость – короткое целое) | 1. Емкость топливного бака в литрах (целое)  2. Расход топлива на 100 км в литрах (действи- тельное) | Максимальное расстояние, которое может пройти авто- мобиль в км (действитель- ное число) |

1.4.2 Обработка данных в контейнере

4. Упорядочить элементы контейнера по возрастанию используя шей- керную сортировку (Shaker Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

1. Метрики, определяющие характеристики программы, для различных тестовых прогонов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Количество | Nasm | C | C++ | Python |
| 10 | 0 sec 2396 nsec | 0.000833 sec | 0.000503 sec | 0.001081 sec |
| 100 | 0 sec 99010 nsec | 0.001418 sec | 0.001610 sec | 0.005176 sec |
| 1000 | 0 sec 47187384 nsec | 0.050799 sec | 0.023431 sec | 0.286122 sec |
| 10000 | 1 sec 230001229 nsec | 1.439316 sec | 1.057313 sec | 29.346563 sec |

3. Данные, демонстрирующие сравнение с характеристиками ранее разработанных программ.

В сравнении с программами, разработанными на С, С++ и Python:

* Имеет меньшее время работы, но приближенное к времени программы на С++
* Имеет больше исходного кода (37,1 КБ против 16 KБ на C, 14 КБ на Python, 10 КБ на С++)
* Прямая работа с памятью
* Написание кода и дебаг занимает намного больше времени

Вывод:

ООП код выглядит структурирование, код удобное для написания и дальнейшего расширение.

Динамически типизированный язык требует от разработчиков большей внимательности при разработке, так как ошибки не всегда получается отловить на этапе компиляции. Однако время исполнения программы превышает время исполнения на статически типизируемом языке.

Программирование на низкоуровневом языке намного усложняет разработку, работа с памятью очень тонкое дело, требующее максимальной концентрации и досконального понимания происходящего. Код занимает намного больше места из-за этого его читабельность ухудшается. Непосредственно есть выигрыш во времени, но не особо значительные по сравнению с языком С. Впрочем низкоуровневый язык в основном применяется тогда, когда другие языки не могут подойти.