Национальный исследовательский университет «Высшая школа экономики»

Домашнее задание №4

Архитектура вычислительных систем

Пояснительная записка

Исполнитель

студент группы БПИ205

Верзаков Ефим Андреевич

2021 г.

Задание

Вариант: 166

Номер задачи: 12

Номер функции: 12

Составление программы по условию:

* Обобщенный артефакт, используемый в задании: Животные.
* Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив):

1. Рыбы (место проживания – перечислимый тип: река, море, озеро…)
2. Птицы (отношение к перелету: перелетные, остающиеся на зимовку – булевская величина)
3. Звери (хищники, травоядные, насекомоядные – перечислимый тип)

* Общие для всех альтернатив переменные:

1. Название – строка символов
2. Вес в граммах (целое)

* Общие для всех альтернатив функции

1. Частное от деления суммы кодов названия животного на вес (действительное число)

* Расположение всех альтернатив в едином контейнере.
* Упорядочить элементы контейнера по убыванию используя сортировку с помощью прямого обмена или пузырька (Bubble Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.
* Формат ввода:

1. Готовые тестовые файлы (bin/task\_4\_c -f <input file with data> <output file> <output sorted file>) – C + void\*

(task -f <input file with data> <output file> <output sorted file>) – C + NASM

input file:

Количество элементов в контейнере.

Элементы описываются так:

1. Первый аргумент тип животного (1 – рыба, 2 – зверь, 3 - птица).
2. Второй аргумент общий для всех – Название – строка 14 символов (для C) или 15 символов (для C + NASM)
3. Третий аргумент общий для всех – Вес в граммах – целое число
4. Четвёртый аргумент уникальный (рыбы – место проживания (перечислимый тип): 1 – река, 2 -море, 3 – озеро; птицы – отношения к перелету (булевская величина): 0 – остающиеся на зимовку, 1 – перелетные; звери – вид (перечислимый тип): 1 – хищники, 2 – травоядные, 3 - насекомоядные)
5. Генерация случайных значений (bin/task\_4\_c -n <number of animals> <output file> <output sorted file>) для C

(task -n < number of animals> <output file> <output sorted file>) – C + NASM

Характеристики программы

Интерфейсные модули: 2

Модули реализации: 7 (C + NASM), 4 (C + void\*)

Результаты тестов (в секундах)

|  |  |  |
| --- | --- | --- |
| Количество элементов | С + void\* | C + NASM |
| Random (10000 элементов) | 5.499s | 2.92s |
| 300 элементов (test01) | 0.014s | 0.01s |
| 1000 элементов (test02) | 0.093s | 0.04s |
| 2000 элементов (test03) | 0.294s | 0.14s |
| 5000 элементов (test04) | 1.594s | 0.80s |
| 10000 элементов (test05) | 5.282s | 2.90s |

Заключение

Заметно, что C + NASM работает примерно в два раза быстрее, чем C + void\*. Это связано с работой с регистрами -> быстрее работает сортировка.