Национальный исследовательский университет «Высшая школа экономики»

Домашнее задание №2

Архитектура вычислительных систем

Пояснительная записка

Исполнитель

студент группы БПИ202

Галюта Ева Валерьевна

2021 г.

Задание

Вариант: 46

Номер задачи: 4

Номер функции: 6

Составление программы по условию:

* Обобщенный артефакт, используемый в задании: объемная (трехмерная) геометрическая фигура.
* Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив):

1. Шар (целочисленный радиус)
2. Параллелепипед (три целочисленных ребра)
3. Правильный тетраэдр (длина ребра – целое)

* Общие для всех альтернатив переменные:

1. Плотность материала фигуры (действительное число)

* Общие для всех альтернатив функции

1. Вычисление площади поверхности (действительное число)

* Расположение всех альтернатив в едином контейнере.
* Упорядочивание элементов контейнера по возрастанию используя Shaker Sort. В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.
* Формат ввода:

1. Готовые тестовые файлы (./figures\_hw1 -f <input file with data> <output file> <output sorted file>) input file:
2. Первый аргумент тип фигуры число от 1 до 3 (1 – сфера, 2 – параллелепипед, 3 – правильный тетраэдр).
3. Второй аргумент уникальный (сфера – радиус (целочисленное значение), параллелепипед – 3 стороны (целочисленные значения), правильный тетраэдр – сторона (целочисленное значение)).
4. Третий аргумент общий для всех – плотность материала фигуры – действительное число.
5. Генерация случайных значений (./figures\_hw1 -n <number of figures> <output file> <output sorted file>)

Структурная схема изучаемой архитектуры ВС с размещенной на ней разработанной программы

Таблица типов

|  |  |
| --- | --- |
| int  double | 4 байта [0]  8 байт [4] |
| struct Sphere | 12 байт |
| int radius  double density | 4 байта [0]  8 байт [4] |
| struct Parallelepiped | 20 байт |
| int a, b, c  double density | 3 \* 4 байт (12 байт) [0]  8 байт [12] |
| struct Tetrahedron | 12 байт |
| int edge  double density | 4 байта [0]  8 байт [4] |
| Struct Shape | 52 байта |
| enum key  key k  Sphere s  Parallelepiped p  Tetrahedron t | 4 байта [0]  4 байта [4]  12 байт [8]  20 байт [20]  12 байт [40] |
| Struct Container | 1560008 байт |
| enum max\_len  int len  Shape cont | 4 байта [0]  4 байта [4]  52 \* 30000 байт (1560000 байт) [8] |

Память программы

|  |  |
| --- | --- |
| main(int argc, char \*argv[])  int argc  char \*argv  Container c  int size | 4 байта [0]  8 байт [4]  1560008 байт [12]  4 байта [1560020] |
| void ErrMessage1 |  |
| void ErrMessage2 |  |
| void Init(Container &c) |  |
| void In(Container &c, FILE \*file) |  |
| void InRnd(Container &c, int size) |  |
| void ShakerSort(Container &c, int size) |  |
| bool swapped  int start, end, i | 1 байт[0]  3 \* 4 байта (12 байт)[1] |
| void Swap(Shape &a, Shape &b) |  |
| Shape temp | 52 байта[0] |
| void Out(Container &c, FILE \*fileOutout) |  |
| Int i | 4 байта[0] |
| void Clear(Container &cont) |  |
| Int i | 4 байта[0] |

Стек вызовов (| - вызов одной из функций, ? – поведение программы при вызове функции)

|  |
| --- |
| main  ErrMessage1? – ~~main~~  Init  ~~Init~~  In (error – ~~main~~) | InRnd | ErrMessage2? – ~~main~~  ~~In~~ | ~~InRnd~~  Out  ~~Out~~  ShakeSort  ~~ShakeSort~~  Out  ~~Out~~  Clear  ~~Clear~~  ~~main~~ |

Характеристики программы

Интерфейсные модули: 6

Модули реализации: 7

Общий размер исходных тестов: 827 Кб

Общий размер результатов тестов: 6,1Мб

Результаты тестов (в секундах)

|  |  |  |
| --- | --- | --- |
| Количество элементов | Ввод из файла | Генерация элементов |
| 20 (test\_1) | 0.668352 | 0.000799 |
| 100 (test\_2) | 0.667335 | 0.001639 |
| 1000 (test\_3) | 2.643455 | 0.107607 |
| 10000 (test\_4) | 2.692486 | 7.374314 |
| 30000 (test\_5) | 10.250734 | 56.974075 |

Различия во времени не зависят от типа ввода, минимальные различия может выдавать сортировка, так как генерируемые данные могут быть частично упорядочены.

Заключение

Было разработано приложение по заданному ТЗ, а также проведен полный анализ структуры, памяти и времени работы.