

Отчет

Отчет по домашнему заданию №4.

Архитектура вычислительных систем.

Номер варианта: 161

Номер задания: 7 (Фильмы)

Номер функции: 12 (Bubble-sort)

Выполнил Ахундов Алексей Назимович, БПИ205.

Задание:

Основываясь на данных таблице по номеру задания и номеру функции, разработать программу в произвольном стиле, при условии динамической типизации, поддерживающую ввод, вывод, складирование и сортировку внутри контейнера объектов (фильмов), имеющих альтернативы (игровое кино, мультфильм, документальное кино)

Сортировка осуществляется по ключу, определяемому функцией на основе общих для всех альтернатив полей (год выхода, деленный на количество символов в названии фильма)

Основные характеристики программы

Число интерфейсных модулей: 0

Число модулей реализации: 8

Общий размер исходных текстов: 73,6 КБ

Полученный размер исполняемого кода: 20,4 КБ

Время выполнения программы:
(Intel Core i7-8550U @ 1.8GHz – 8CPUs)

Детерминированные тесты:

Время исполнения				
Тест	Стиль			Процедурный
	Процедурный	Объектно-ориентированный	Объектно-ориентированный	
	статическая типизация	статическая типизация	динамическая типизация	Assembly
1	0s	0s	0s	0s
2	0s	0s	0s	0s
incorrect_0	0s	0s	0s	0s
incorrect_1	0s	0s	0s	0s
incorrect_2	0s	0s	0s	0s

Тесты стохастическими данными:

Время исполнения				
Тест	Стиль			Процедурный
	Процедурный	Объектно-ориентированный	Объектно-ориентированный	
	статическая типизация	статическая типизация	динамическая типизация	Assembly
1 (1 000)	0.0156s	0.0156s	0.625s	0.09s
2 (5 000)	0.4375s	0.3437s	17.7812s	2.17s
3 (10 000)	1.8125s	1.4218s	101.9687s	8.42s

Сравнительный анализ

Программа, написанная на ассемблере (nasm), уступает в производительности решениям на C++ в обоих стилях, обосновано это в первую очередь отсутствием буферизированного ввода/вывода, во всех остальных аспектах программа как минимум ничем не уступает своим предшественникам, а где-то, за счет большей гибкости (с точки зрения возможных инструкций, более низкого уровня), обгоняет их.