

Многие знают про техническую избыточность кода написанного на ЯП высокого уровня. Некоторые знают и про алгоритмическую избыточность. Но руками мало кто трогал. Если про техническую избыточность рассуждать просто — достаточно реализовать заданный алгоритм на разных ЯП и сравнить результаты, то для выявления алгоритмической избыточности нужно иметь задачу и ее идеальную реализацию. Задачу поставить просто, но как получить идеальный алгоритм? Тут дело творческое... Нужно выдумать задачу и заставить талантливых разработчиков ее решать и решать. А кого из талантливых заставишь этим заниматься? Спешу обрадовать! Такие люди бывают!

По материалам <http://www.hugi.scene.org> (в начале 2000х я даже был победителем подобного конкурса — не рекомендую, голову сносит...) был проведен анализ технической и алгоритмической избыточности. Я приведу табличку по одному из проектов.

Анализ результатов конкурса hugi size coding competition №1

Описание и постановка задачи

Язык	Англ.	Нем.
Объем файла (байт)	2920	3174

Реализация постановщика задачи

Язык реализации	Turbo C 16 bit
Объем исходных текстов (байт)	5021
Объем программы (байт)	14510
Количество функций в исходных текстах	8
Количество функций в программе	95
Количество команд ветвления в исходных текстах	10
Количество команд ветвления в программе (jmp, call и т.п.)	468

Реализация конкурсантов (степень избыточности загрузочного кода очень мала)

Разработчик	«Altair / ODDS»
Язык реализации	tasm
Объем исходных текстов (байт)	2998
Объем программы (байт)	48
Количество функций в исходных текстах	3
Количество функций в программе	3
Количество команд ветвления в исходных текстах	2
Количество команд ветвления в программе (jmp, call и т.п.)	2
Выявленная таким образом избыточность устранена: (в размах)	
По объему исходных текстов	1,7
По объему программы	302,3

По количеству функций в исходных текстах	2,7
По количеству функций в программе	31,7
По количеству команд ветвления в исходных текстах	5,0
По количеству команд ветвления в программе	234,0
Разработчик	«Махх»
Язык реализации	tasm
Объем исходных текстов (байт)	956
Объем программы (байт)	51
Количество функций в исходных текстах	3
Количество функций в программе	3
Количество команд ветвления в исходных текстах	3
Количество команд ветвления в программе (jmp, call и т.п.)	3
Выявленная таким образом избыточность устранена: (в размах)	
По объему исходных текстов	5,3
По объему программы	284,5
По количеству функций в исходных текстах	2,7
По количеству функций в программе	31,7
По количеству команд ветвления в исходных текстах	3,3
По количеству команд ветвления в программе	156,0
Разработчик	«Mode 19»
Язык реализации	tasm
Объем исходных текстов (байт)	1125
Объем программы (байт)	51
Количество функций в исходных текстах	3
Количество функций в программе	3
Количество команд ветвления в исходных текстах	2
Количество команд ветвления в программе (jmp, call и т.п.)	2
Выявленная таким образом избыточность устранена: (в размах)	
По объему исходных текстов	4,5
По объему программы	284,5

По количеству функций в исходных текстах	2,7
По количеству функций в программе	31,7
По количеству команд ветвления в исходных текстах	5,0
По количеству команд ветвления в программе	234,0
Разработчик	«Rib»
Язык реализации	tasm
Объем исходных текстов (байт)	1398
Объем программы (байт)	52
Количество функций в исходных текстах	3
Количество функций в программе	3
Количество команд ветвления в исходных текстах	3
Количество команд ветвления в программе (jmp, call и т.п.)	3
Выявленная таким образом избыточность устранена: (в размах)	
По объему исходных текстов	3,6
По объему программы	279,0
По количеству функций в исходных текстах	2,7
По количеству функций в программе	31,7
По количеству команд ветвления в исходных текстах	3,3
По количеству команд ветвления в программе	156,0