

# ИДЗ-2 Вариант 6

Лобанов Кирилл Сергеевич БПИ-213

Ноябрь 2022

Условие задачи - Разработать программу, заменяющую все согласные буквы в заданной ASCII-строке их ASCII кодами в десятичной системе счисления. Строки лежат в заранее выделенных массивах buf и ans. Файл avs.c - программа на Си, avs\_asm.S - изначальная программа на Ассемблере, avs\_reg.S - версия с регистрами, avs\_full.S и avs\_compute.S - версия с регистрами, разбитая на 2 единицы компиляции. 4 балла) Была написана программа на языке C avs.c, затем она была собрана с помощью флагов gcc -masm=intel -fno-asynchronous-unwind-tables -fno-jump-tables -fno-stack-protector -fno-exceptions ./avs.c -S -o ./avs\_asm.S После чего был получен ассемблерный код avs\_asm.S, к нему были добавлены необходимые комментарии, затем была проверена корректность работы программ с помощью тестов из папки tests.

5 баллов) Программа avs.c была изначально разбита на функции main, compute и write, которые считывали исходную строку, получали новую строку и выводили её, все переменные были локальные и параметры передавались в функции, поэтому код на C и на ассемблере не поменялся.

6 баллов) В функции main был использован регистр r12 для хранения переменной i, то есть длины исходного текста. В функции compute, которая самая большая по времени работы, потому что мы сами задаем число её повторений, были использованы регистры r13, r14 и r15. Это не сделало код меньше по объему, но ускорило, потому что обращение к памяти дольше, чем к регистрам. Более точную оценку ускорения можно увидеть ниже.

7-8 баллов) Код на ассемблере был разбит на 2 части - в одной функция compute, в другой все остальное: получилось 2 файла avs\_compute.S и avs\_full.S, затем они оба были преобразованы в машинный код - получили файлы avs\_compute.o и avs\_full.o, затем они слинковывались в исполняемый файл a.out командой gcc avs\_compute.o avs\_full.o Был сделан файловый ввод/вывод и возможность генерации случайных тестов, выбор пользователю для генерации/ввода предоставляется с помощью командной строки - для генерации надо первым параметром ввести -random, для ввода - название файла или его путь, для вывода в файл всегда надо третьим параметром передавать его название или путь. Второй параметр отвечает за число повторений в цикле, чтобы было удобнее замерять время работы программы. В итоге время работы программы на строке из 10 символов "abcdefghij" без регистров:

100000 повторений - 0.007 секунд  
1000000 повторений - 0.075 секунд  
10000000 повторений - 0.749 секунд  
100000000 повторений - 7.557 секунд

После добавления регистров вместо обращения к памяти:

100000 повторений - 0.007 секунд  
1000000 повторений - 0.071 секунд  
10000000 повторений - 0.711 секунд  
100000000 повторений - 7.098 секунд

Из этих экспериментов видно, что разница во времени составляет от 5 до 6 процентов, что, конечно, не очень много, но любое ускорение - вещь приятная.