ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ФИЗТЕХ-ШКОЛА РАДИОТЕХНИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Лабораторная работа по программированию.

По теме

«Ускорение работы хэш-таблицы»

Студента 1 курса группы Б01-003
Крейнина Матвея Вадимовича

Долгопрудный, 2021

Исследование эффективности работы хэш-функций

Цель работы:

Оптимизировать хэш таблицы любыми доступными средствами.

В работе используются:

Кривые руки Матвея Крейнина, классы list и line, написанные им же в 1-м семестре, Visual Studio с её прекрасным, красивым и невообразимым профилировщиком, хорошее настроение и любовь к программированию. Все тесты проводились при максимальной производительности ноутбука, подключенного к зарядному устройству и в режиме оптимизации компилятора o2.

Теоретическое введение:

Это секретная информация, которую нельзя разглашать. Возможно, вы сможете найти ответ в книге человека, которого нельзя называть.

Ход работы:

1. Для начала я решил измерить время работы своей программы, загрузив в хэш таблицу англо-русский словарь на 158 тысяч слов. Далее я ищу каждое из слов по несколько раз с целью того, чтобы найти «узкое горлышко» в своей программе.

Рис. 1: Режим компилятора – o2, время исполнения = 180 мс

Function Name	Self CPU [unit, %]
T09HASHETABLE_OPT.exe (PID: 18308)	0 (0,00%)
rot_13	516 (98,47%)
mk5::hash_table::insert	6 (1,15%)

Я нашел узкое место - это вызов функции rot_13, я перешел в неё и обнаружил, что в цикле я каждый раз считаю длину строки для сравнения.

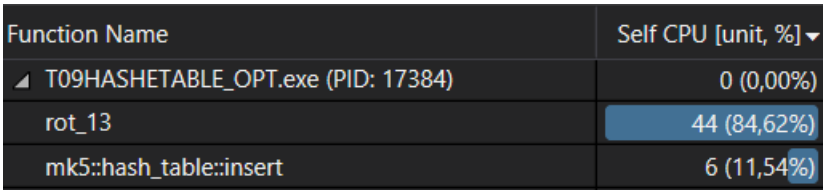
Я вынес подсчёт длины строки из цикла и получил следующие показатели:

Рис. 2: Режим компилятора – o2, время исполнения = 43 мс

Function Name	Self CPU [unit, %]
T09HASHETABLE_OPT.exe (PID: 16984)	0 (0,00%)
rot_13	131 (92,91%)
mk5::hash_table::insert	6 (4,26%)

Рис. 3: Режим компилятора – o2, время исполнения = 30 мс

Теперь доля функции rot_13 снизилась с 98.47% до 92.91%, что даже очень хорошо, но можно лучше. При этапе чтения англо-русского словаря, я теперь буду запоминать длину строки и сохраняю её для каждого элемента и теперь её считать не нужно.



Теперь время работы функции rot_13 снизилась с 92.91% до 80.43% и выросла работа функции insert.

180 мс - время работы с подсчётом строки на каждом шаге,

43 мс - время работы с подсчётом длины строки вне цикла,

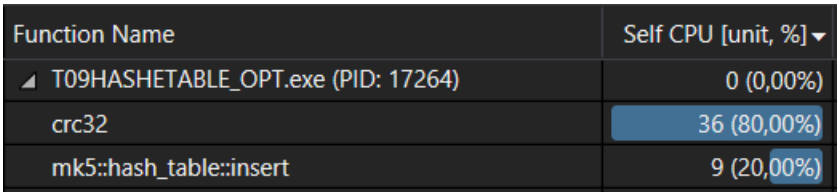
35 мс - время работы с предпосчётом длины строки на этапе загруки.

Благодаря манипуляциям с подсчётом длины строки я получил ускорение в 4.18 раза при подсчёте вне цикла, вероятно, это связано с тем, что строчки у меня короткие, поэтому не так часто их приходится пересчитывать, в 5.14 раза при подсчёте на этапе загрузки. Это уже довольно хороший результат, но нужно двигаться дальше.

Вооружившись книгой Брайнта-Холларона «Компьютерные системы архитектура и программирование», в пятой по счету главе я нашел, что цикл можно сворачивать выполняя на каждом шаге не одну операцию, а, например, 2. Применив данный метод, для свертывания цикла в два раза я получил - 33 мс, я выиграл 2 мс. Далее я поэксперементировал с 3, 4, 5 и 6, получил соотвественно: 32 мс, 29 мс и 30 мс, 32, больше я не стал продолжать этот опыт, т.к. локальный минимум был найден.

Погуглив еще немного, я нашел в икстринсиках хэш-функцию - это crc32, я внедрил её в свой проект и получил следующий результат.

Рис. 4: Режим компилятора – o2, время исполнения = 30 мс



Дальнейшее ускорение программы не представляется возможным, т.к. переписывание хэш функции на икстринсики в первый раз дало проигрыш по времени, а во вторый точно такой же результат, как и оптимизация от Visual Studio, дальнейшее оптимизация функций rot_13 и crc32 не представляется возможным, по причине того, что подсчёт длины строки оптимизирован и выполняется на этапе загрузки, цикл в подсчёте хэш-функции оптимизирован.

Вывод

Я ускорил программу в 6.2 раза в режиме компиляции O2, что является хорошим результатом, т.к. оптимизатор visual studio является одним из лучших. Теперт нужно подсчитать самое главное число

Список литературы

1. Язык программирования СИ, Брайан Керниган и Деннис Ритчи

2. Компьютерные системы архитектура и программирование, Брайнт-Холларон

3. Дединский Илья Рудольфович http://ded32.net.ru

Возможно, вы заходите заняться компутерной графикой и тогда вы сможете найти одну библиотеку, которая сможет вам помочь.

4. Google https://www.google.ru