

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №2

Вариант №10

Выполнили:

Шишминцев Д.В., Язев Г.А., Абоимов А.А.

Проверил:

Мусаев А.А.

Санкт-Петербург

2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ПОСТАНОВКА ЗАДАЧИ.....	4
Ход работы.....	5
ЗАКЛЮЧЕНИЕ	8
СПИСОК ЛИТЕРАТУРЫ	9
ПРИЛОЖЕНИЯ	10

ВВЕДЕНИЕ

В данной работе будут представлены алгоритм хэширования умножением и алгоритм хэширования MD5, а также реализована программа, использующая эти алгоритмы для хэширования введённого пользователем текста.

ПОСТАНОВКА ЗАДАЧИ

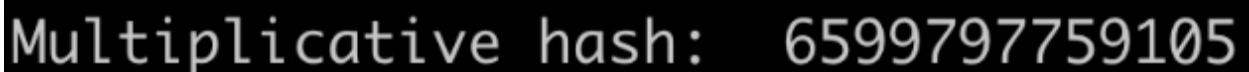
Необходимо изучить и реализовать 2 алгоритма хэширования, а именно алгоритм хэширования методом умножения и алгоритм хэширования MD5. В результате должна получиться программа, осуществляющая хэширование введённого текста с помощью этих двух алгоритмов.

Ход работы

Задание 1. Программа для вычисления хэша для введенного текста.

1.1. Метод умножения

Был реализован алгоритм хэширования методом умножения, который для вычисления хэша использует выражение $M * ((K * C) \bmod 1)$, где M – размер массива, K – ключ, $C = (5^{*(0.5)} - 1) / 2$ – оптимальная константа, которую вывел Дональд Кнут. Строка из букв переводится в строку из цифр (ASCII кодов), после чего вычисляется выражение. После вычисления данного выражения мы получаем строку из цифр – нужный нам хэш. Пример результата работы данного алгоритма показан на рисунке 1.



```
Multiplicative hash: 6599797759105
```

Рисунок 1. Результат выполнения алгоритма умножения

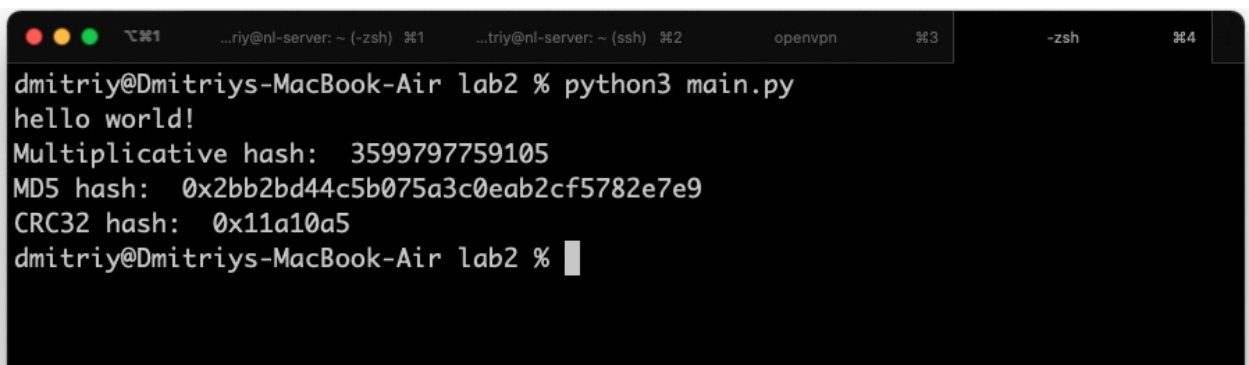
Вывод: Метод умножения – простой алгоритм хэширования, который является основой для более продвинутых алгоритмов. Однако, при использовании данного алгоритма наблюдается большое число коллизий (ситуаций, когда хэши слов равны, но сами слова различаются), из-за чего приходится проводить множество дополнительных проверок, замедляющих общую работу.

1.2. Алгоритм MD5

Был реализован алгоритм хэширования MD5. Данный алгоритм основан на 5 этапах: выравнивание потока, добавление длины, инициализация буфера, вычисление, представление результата. На первом этапе происходит перевод изначальной строки, в строку = $(512 * L + 448)$ бит, где L – натуральное число. Это является

подготовкой ко второму этапу, на котором к полученной строке дописывается 64-битное представление длины исходной строки. По итогу выходит строка кратная 512, которая разбивается на 32 блока по 16 битов в каждом. На 3-ем этапе инициализируются буффер, состоящий из 4х 32-битных переменных. На 4-ём этапе происходит само вычисление хэша. Определяется 4 вспомогательные логические функции и определяется константа $T[i] = 2^{32} \cdot \sin(i)$, где $i = 1 \dots 64$ (Она нужна для усиления алгоритма) и константы сдвига S . Далее каждый 16-битный блок X копируется в отдельные массивы и происходит замена: $AA = A$, $BB = B$, $CC = C$, $DD = D$. Затем происходит 64 преобразования по формуле $A = B + ((A + F(B, C, D) + X[k] + T[i]) \gg S)$, где $X[k]$ – k -ый элемент 16-битного блока. На каждом преобразовании происходит суммирование изначального значения буффера с новым, после чего следует правый сдвиг буфферов ($ABCD \ggg DABC$). После 64 итераций, результат (изменённые переменные буффера (A, B, C, D)) суммируется с изначальным значением переменных буффера. На 5-ом этапе происходит побайтовый вывод буффера $ABCD$ (Начиная с A , заканчивая D). Выведенная строка и будет искомым хэшем.

Результат работы данного алгоритма показан на рисунке 2.



```

dmitriy@Dmitriys-MacBook-Air lab2 % python3 main.py
hello world!
Multiplicative hash: 3599797759105
MD5 hash: 0x2bb2bd44c5b075a3c0eab2cf5782e7e9
CRC32 hash: 0x11a10a5
dmitriy@Dmitriys-MacBook-Air lab2 %

```

Рисунок 2. Результат выполнения алгоритма MD5

Вывод: MD5 – сложный, но точный алгоритм и крайне надёжный алгоритм, применяющийся для защиты данных и обнаружения ошибок в потоке информации. При выполнении данного алгоритма всё ещё проявляются

коллизии, но их уже значительно меньше, чем при использовании метода умножения.

ЗАКЛЮЧЕНИЕ

В ходе данной работы были изучены различные методы хэширования, а также написана программа на языке Python, реализующая работу метода умножения и алгоритма MD5.

СПИСОК ЛИТЕРАТУРЫ

1. Хабр. Хэш-функция MD5. [Электронный ресурс]. [Сайт]URL:
<https://goo.su/upgi>

ПРИЛОЖЕНИЯ

A. <https://goo.su/0mJULz> - ссылка на github.