**Report on Hash Table Offline**

The task was to generate 10000 words and then store them in a Hash Table using two different hash functions. For this, I have used the following functions:

// 01

unsigned long hashFunction(char \*str)

{

    unsigned long hash = 5381;

    int c;

    while (c = \*str++)

        hash = ((hash << 5) + hash) + c; /\* hash \* 33 + c \*/

    return hash;

}

// 02

unsigned long hashFunction(char \*str)

{

    unsigned long hash = 0;

    int c;

    while (c = \*str++)

        hash = c + (hash << 6) + (hash << 16) - hash;

    return hash;

}

// aux

unsigned long auxHashFunction(char \*str)

{

    unsigned long hash = 0x55555555;

    while (\*str) {

        hash ^= \*str++;

        hash = (hash << 5) | (hash >> (32 - 5));

    }

    return hash;

}

Now, the performance of these two functions (the third one is an auxiliary hash function used in double hashing and custom probing) for 3 different kinds of collision resolution could be found in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Hash 01 | | Hash 02 | |
|  | # of collisions | average probes | # of collisions | average probes |
| Chaining Method | 1553 | 9.262 | 1627 | 9.416 |
| Double Hashing | 76481 | 8.745 | 71876 | 7.978 |
| Custom Probing | 75954 | 8.208 | 77662 | 7.499 |