Assignment 6 First Design

James Contini

2022 October 26

General information

Citations

None.

Description

Program takes a stdin file, a list of badspeak words, a text file of oldspeak and newspeak pairs. The badspeak and oldspeak should be added to the bloom filter. Oldspeak and newspeak should be added to the hash table.

From stdin read a stream of words to test. For each word check if its in the bloom filter. If there is a negative do nothing. If *bf_probe*() determines it is in the bloom filter then search the hash table for the word. If the word doesn't have a newspeak translation, then the citizen is guilty of *thoughtcrime*. Insert this word in the list of evidence of badspeak words for this citizen. If the word turns out to just be oldspeak not bad speak add it to a list of evidence on evidence for being sent to a joy camp for rightspeak. If the hash list does not the old/badspeak then the bloom filter issues a false positive, do nothing. We can use any type of data strcture to hold this evidence.

Depending on these combos, only badspeak, only oldspeak, a mix, they get different messages.

1 Bit Vector

The bitvector module I implemented works very simply. A bit vector is comprised of a pointer to blocks of uint64t type. To set a bit in the vector *or* we find the block by floor dividing the bit number by sixty four. Then to find which bit inside we take the remainder of the bit number modulo sixty four. Then we create a mask of a one at that number and zeros everywhere else and then *or* that mask with the original bit vector. To delete the mask becomes ones and a zero and the operator *and* and to inspect we *or* the vector with a of zeroes except for the set bit. If the result of that operation is equivalent to the original mask we can conclude that there was already a one there.

2 Bloom Filter

The bloom filter uses the bitvector to store its data. To insert a long string of words split by newlines is inputted. For every word denoted by a newline it is hashed 5 times with different salts and the hash result is the bit the program sets in the bitvector. To probe the vector we hash a string and see if it has at least one bit not set, if a bit is not set then it is definetly not in the hash table. But if all bits are set, there is a good chance it is.

3 Node/Hash table

The linked list will be linking a list of arrays that will house oldspeak and its newspeak translations in that order. To access an oldword it will be in a node (which will just be an array) each node will correspond to one oldword or multiple oldword's hash. Then the program will iterate through the array until it finds the oldword in the list, then the next element is the newspeak.

4 Linked list

A doubley linked list has twice the overhead as a singularly linked list. In my implementation of a linked list there will be a previous field and forward field holding the addresses in memory of different nodes. This way I can go up and down the list.

5 Banhammer

Using *fgets()*, the program will take words from an input stream. Since a words has no specifice length, but a limit to the line length if there are large words, words are parsed by the thousand character mark. They can be put together and then put under lexical scrutiny.