ASSIGNMENT 6 DESIGN DOCUMENT

Pehara Vidangamachchi

November 11th 2022

Ethan L. Miller

About:

This assignment aims to create a bloom filter and a hash table. By implementing such, my program should be able to filter through newspeak and bad speak and filter out any phrases using bad speak. Essentially the bloom filter acts as a space-efficient probabilistic data structure. As described in the assignment document, a bloom filter can be represented as an array of bits "m", it will utilize k different hash functions. Which then allows the filter to set an element to the filter, generating a uniform pseudo-random distribution. And in order to complete the above, they will be executed through the use of several functions such as ht.c, ll.c, node.c, bf.c, bv.c, parser.c and finally Makefile

Design Process for ht.c

The purpose of this program is to create a hash tab;e. Sharing similarities with the bloom filter, a hash table contains a salt that gets passed down whenever a new old speak element is added. Using chained hash tables allows us to avoid collisions. Following the pseudo-code given in the assignment, I should be able to execute the code in a functional manner.

#include any header files

#declare any variables

Create a function to create the hash table:

Set the hash table equal to its variable counterparts pointer

Which receives the size of the hash table

Create an if statement that sets up all the parameters of several variables

Set the mtf, salt, num hits, num keys, size and lists equal to their respective counterpart.

Create an if statement in case the hash table is not equal to the lists

Free the allocated memory and set it equal to null

Create a function to delete the hash table:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the hash table

Clear the memory if the statement runs true

Return with the empty memory

Create a function to create the hash tables size:

Initialise any variables that will be utilized within the code

Get the size of the hash table using a while loop:

Followed by an if statement when the conditions are true:

Getting the math portion of its functionality

Create a function that looks through the hash tables contents:

Initialise any variables that will be utilized within the code

Create a while loop that searches for the nodes and old speak:

Using the number of stats and linked lists

Update the old information with the new information

Return the difference

Create a function that keeps count of the hash tables:

Initialise any variables that will be utilized within the code

Get the count of bits within the hash table

Create a function that prints the hash tables contents:

Initialise any variables that will be utilized within the code

Create a print statement to print out the hash table

Create a function that keeps track of the hash tables stats:

Initialise any variables that will be utilized within the code

Using pointers set the value equal to the original

Go through each variable till they are all converted using a while loop

Design Process for ll.c

The purpose of this program is to create a linked list. They are defined as having two sentinel nodes and the field mtf. The assignment document goes in depth about their functionality and purpose. So following the directions given there I should be able to implement my own program regarding this segment.

#include any header files

#define any variables

Create a function that creates a linked list:

Initialise any variables that will be utilized within the code

Create a while loop to run given the correct parameters:

Create an if statement when mtf is true and a bool:

Filter through the nodes

Create a function that deletes the linked list:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the linked list

Clear the memory if the statement runs true

Return with the empty memory

Create a function that gets the length of the linked list:

Initialise any variables that will be utilized within the code

Get the size of the linked list using a while loop:

Followed by an if statement when the conditions are true:

Getting the math portion of its functionality

Create a function that looks up the contents within the linked list:

Initialise any variables that will be utilized within the code

Create a while loop that searches for the nodes and old speak:

Using the number of stats and linked lists move front

Update the old information with the new information

Return the difference

Create a function that inserts a character within the linked list:

Initialise any variables that will be utilized within the code

Create an if statement if a node contains oldspeak:

Replacing it with new speak

And checking that the node does nto already contain a match

Else the node is added at the head of the list

Create a function that prints the contents of the linked list:

Initialise any variables that will be utilized within the code

Create a print statement to print out the linked list

Create a function that compiles the stats of the linked list:

Initialise any variables that will be utilized within the code

Using pointers set the value equal to the original

Go through each variable till they are all converted using a while loop

Check how the numbers of links traversed between all the other programs

Design Process for node.c

The purpose of this program is to create a doubly linked list that will be used to solve hash collisions, with each passing node containing a segment of old speak and its respective newspeak translation. Each node contains some pointer towards a previous node due to the implementation of linked lists. This will be done by following the assignment documents directions with its pseudo code.

#include any header files

Create a function to create the node:

Initialise any variables that will be utilized within the code

Create a copy of old speak and new speak by allocating it memory

Create a function to delete the node:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the nodes n

Clear the memory if the statement runs true

Return with the empty memory

create a function to print the node:

Initialise any variables that will be utilized within the code

Create a print statement for each condition within an if statement

Design Process for bf.c

The purpose of this program is to create a bloom filter following the directions given to us in the assignment document. This can be done by following the given pseudo code and executing the code in such a way that creates a functioning filter. Information within the filter will be stored within an array and collect certain statistics.

#include any header files

#define any variables

Create a function that creates the bloom filter:

Initialise the variables used within the code

Create an if statement to run the code when the conditionals are true:

Set the bloom filter equal to the number of keys

Set the number of misses equal to the number of bits examined

Create a for a loop when an integer is within a certain range:

Set the salts equal to the default salts

Set the filter equal to the create function

Create an if statement in the case that filter is equal to null:

Free the memory for the bloom filter

Set it equal to null

Create a function for deleting the bloom filter:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the bloom filter

Clear the memory if the statement runs true

Return with the empty memory

Create a function for getting the size of the bloom filter:

Initialise any variables that will be utilized within the code

Get the size of the bloom filter using a while loop:

Followed by an if statement when the conditions are true:

Getting the math portion of it

Create a function for inserting the bloom filter:

Initialise any variables that will be utilized within the code

Filter through old speak and add it to the filter

This is done by hashing old speak with the five salts

Create a function for probing the bloom filter:

Initialise any variables that will be utilized within the code

Create an if statement so when all the bits in the previous code are true

Return true to say that old speak was added to the filter

Else return false

Create a function for keeping count with the bloom filter:

Initialise any variables that will be utilized within the code

Get the count of bits within the bloom filter

Create a function that prints out the bloom filters contents:

Initialise any variables that will be utilized within the code

Create a print statement to print out the bloom filter

Create a function for the bloom filters stats:

Initialise any variables that will be utilized within the code

Using pointers set the value equal to the original

Go through each variable till they are all converted using a while loop

Design Process for bv.c

The purpose of this program is to create a Bit Vector, the length defines the size of the bit vector. This will be used to help keep track of the number of bits. The code requires the usage of bitwise operations. following the assignment pdf, I plan to execute my code by using the given directions and the accompanying pseudo-coed.

#include any header files

#define any variables

Create a function for deleting the bit vector:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the bit vector

Clear the memory if the statement runs true

Return with the empty memory

Create a function for getting the length of the bit vector:

Initialise any variables that will be utilized within the code

Get the size of the bit vectors using a while loop:

Followed by an if statement when the conditions are true:

Getting the math portion of its functionality

Create a function for setting the bit vectors size;

Initialise any variables that will be utilized within the code

Get the size of the bit vectors using a while loop:

Followed by an if statement when the conditions are true:

Getting the math portion of it

Create a function to clear the bit vector:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the bit vector

Clear the memory if the statement runs true

Return with the empty memory

Create a function to get the bit vectors contents;

Initialise any variables that will be utilized within the code

Create an if statement taht returns:

Returning the ith bit vector

And checking that the node does nto already contain a match

Create a function to print the bit vectors contents:

Initialise any variables that will be utilized within the code

Create a print statement to print out the bloom filter

Design Process for parser.c

The purpose of this program is to act as a regulator for the words that are spoken. It will filter through the valid words and compare them. The program will have to be compatible with almost everything and analyze them properly. I plan to implement this following the directions given within the assignment document.

#include any header files

Create a function to create the parser:

Initialise any variables that will be utilized within the code

Construct the main code for the parser using a while loop:

When the conditions are true execute the code

Create a function to delete a parser:

Initialise any variables that will be utilized within the code

Create an if statement condition that checks for the parser

Clear the memory if the statement runs true

Create a function to transition to the next word:

Initialise any variables that will be utilized within the code

Create a while loop that goes to the next word till it reaches the end:

Using pointers keep transitioning towards the next word

When there are no more words left:

Return a false

Design Process for Makefile

The purpose of this program

#include any header files

Initiazilize all the important variables

Use clang and set up the error messages

Create a function to open the files and compile them

Create a function to close the files and delete the.o files created

Design Process for banhammer.c

The purpose of this program

#include any header files

Create the main function

Combine all the previous code segments into a working function

Create a switch case option to parse through the options that users can input

Using fgets to jump between the different lists