Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Отчёт по лабораторной работе №5

по дисциплине «Лингвистические средства вычислительных систем»

Вариант 18

Выполнил студент гр.

Проверила:.

Тамбов 20

**Задание:** *Составить программу для работы с таблицей символов, организованной методом хеш-адресации. Для решения задачи коллизии использовать метод цепочек. В качестве исходного файла использовать любую программу из личной библиотеки.*

**Описание алгоритма.**

Все данные об идентификаторах в программе заносятся в массив структур tabl. Каждый элемент этого массива содержит имя идентификатора, его тип, значение хеш-функции, а также поле chain.

Хеш-функция организована следующим образом: вычисляется сумма кодов символов, составляющих имя идентификатора; затем полученное число делится на число элементов таблицы и остаток от деления возвращается в качестве значения хеш-функции.

Хеш-функция дает индекс элемента в хеш-таблице. Если элемент с таким хеш-кодом не встречался, то соответствующий элемент в хеш-таблице равен 0. В этом случае в таблицу символов заносится информация о данном идентификаторе, а в хеш-таблицу – номер данного идентификатора в таблице символов.

Если элемент с данным значением хеш-функции уже встречался, то осуществляется прохождение по цепочке идентификаторов с данным значением хеш-функции, при этом используется поле chain. Если обнаруживается совпадение элемента, который нужно занести в таблицу, и элемента таблицы символов, имеющих тот же хеш-код, то выдается сообщение об ошибке. В противном случае идентификатор заносится в таблицу, а в поле chain предыдущего идентификатора, имеющего тот же хеш-код, заносится его номер в этой таблице.

**Текст программы:**

#include<stdio.h>

#include<string.h>

#include<conio.h>

int hash (char \*str,float razm)

{int sum=0,j,hash;

float y;

for(j=0;str[j]!='\0';j++) sum+=str[j];

y=(float)sum/razm;

hash=(y-(int)y)\*10;

return hash;

}

struct desk

{char name[20];

int chain;

char tip[20];

int hash;

};

int main()

{char dt[]=" i:integer;\n file:text;\n sim:simbol;\n qwer:file of byte;\n";

int i=0,j=0,sch=0,l=0,z=1,zhash;

struct desk tabl[31];

struct desk table;

int hashh[10];

printf("+-------------------+\n");

printf(" Otryvok programmy:\n%s",dt);

printf("+-------------------+\n");

for(i=0;i<31;i++) tabl[i].chain=0;

for(i=0;i<10;i++) hashh[i]=0; i=0;

do

{ j=0; l=0;

while(dt[i]!=':') {table.name[j]=dt[i]; i++; j++;} i++;

table.name[j]='\0';

j=0;

while(dt[i]!=';') {table.tip[j]=dt[i]; i++; j++;}

table.tip[j]='\0';

zhash=hash(table.name,31);

if (hashh[zhash]==0)

{sch++;

hashh[zhash]=sch;

tabl[sch].hash=zhash;

strcpy(tabl[sch].name,table.name);

strcpy(tabl[sch].tip,table.tip);

}

else

{j=hashh[zhash];

do {if(strcmp(table.name,tabl[j].name)==0) l++; z=j; j=tabl[j].chain;} while(j!=0);

if(l!=0) printf("\nOshibka: povtorenie - %s - parametra\n",table.name);

else

{sch++;

tabl[sch].hash=zhash;

tabl[z].chain=sch;

strcpy(tabl[sch].name,table.name);

strcpy(tabl[sch].tip,table.tip);

}

}

i+=2; }while(dt[i]!='\0');

printf("\nHash:");

for(i=0;i<10;i++) printf(" %u",hashh[i]);

for(i=1;i<=sch;i++) printf("\n%u\t%-2.5s\t%-10s\tCepochka=%u\t\tHash=%u",i,tabl[i].name,tabl[i].tip,tabl[i].chain,tabl[i].hash);

getch();

return 0;}

**Результат работы программы:**