|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации Мытищинский филиал**  **Федерального государственного бюджетного образовательного учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МФ МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_космический\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_К-3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**отчет**

***К ЛАБОРАТОРНОЙ РАБОТЕ***

***№ 6***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

# *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*по ДИСЦИПЛИНЕ

**«Системное программное ОБЕСПЕЧЕНИЕ»**

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент К3-46Б Чернов В.Д.

(Подпись,

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Чернышов А.В.

*2024 г.*

**Задание**

Целью лабораторной работы является построение хеш-таблицы по выбранному списку ключей и осуществление поиска по таблице выбранного ключа. В качестве ключей рассматриваются мнемокоды команд процессора и директивы ассемблера.

Набором директив является набор I . Хеширование выполняется методом деления. Для разрешения коллизий используется метод открытой адрессации. Количество ключей — 17.

**Текст программы**

**commands.h**

#ifndef COMMANDS\_H

# define COMMANDS\_H

# define MAX\_KEYS 17

const char\* const commands[] = {

"START", "END", "BYTE", "WORD", "RESB", "RESW", "HLT", "ADD", "MUL", "MOV", "LSR", "LSL", "JZ", "JE", "JNZ", "JNE", "JC"

};

#endif

**hash.c**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include "Commands.h"

#define MAX\_SIZE 50

typedef struct s\_node {

char\* cmd;

} t\_node;

unsigned int hash(const char\* str) {

unsigned int h = 0;

while (\*str) {

h ^= \*str++;

}

return h % MAX\_SIZE;

}

int add(t\_node\* table[], const char\* str) {

unsigned int h = hash(str);

while (table[h] != NULL) {

h = (h + 1) % MAX\_SIZE;

}

t\_node\* node = (t\_node\*)malloc(sizeof(t\_node));

node->cmd = strdup(str);

table[h] = node;

return h;

}

t\_node\* get(t\_node\* table[], const char\* cmd) {

unsigned int h = hash(cmd);

while (table[h] != NULL) {

if (strcmp(table[h]->cmd, cmd) == 0) {

return table[h];

}

h = (h + 1) % MAX\_SIZE;

}

return NULL;

}

int main() {

unsigned int h;

t\_node\* table[MAX\_SIZE];

for (int i = 0; i < MAX\_SIZE; ++i) {

table[i] = NULL;

}

for (int i = 0; i < MAX\_KEYS; ++i) {

printf("%s ", commands[i]);

unsigned int key = add(table, commands[i]);

printf("key: %i\n", key);

}

for (int i = 0; i < MAX\_KEYS; ++i) {

t\_node\* node = get(table, commands[i]);

if (node) {

printf("Found: %s\n", node->cmd);

} else {

printf("Not found\n");

}

}

for (int i = 0; i < MAX\_SIZE; ++i) {

if (table[i] != NULL) {

free(table[i]->cmd);

free(table[i]);

}

}

return 0;

}

**Результат работы программы**

START key: 14

END key: 29

BYTE key: 10

WORD key: 15

RESB key: 6

RESW key: 19

HLT key: 30

ADD key: 16

MUL key: 34

MOV key: 35

LSR key: 27

LSL key: 33

JZ key: 17

JE key: 18

JNZ key: 44

JNE key: 20

JC key: 9

Found: START

Found: END

Found: BYTE

Found: WORD

Found: RESB

Found: RESW

Found: HLT

Found: ADD

Found: MUL

Found: MOV

Found: LSR

Found: LSL

Found: JZ

Found: JE

Found: JNZ

Found: JNE

Found: JC