Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Компиляторные технологии

ОТЧЕТ

по лабораторной работе

на тему:

ДВОИЧНОЕ, ВОСЬМЕРИЧНОЕ ИЛИ 16-ЧНОЕ ЧИСЛО В СИНТАКСИСЕ FASM (ПОСТФИКСНАЯ ФОРМА ЗАПИСИ)

|  |  |
| --- | --- |
| Выполнил  Студент гр. 351003 | Е. Д. Гузаев |
| Проверил | С. В. Болтак |

Минск, 2024

**Вариант 5**

**Задание:**

Двоичное, восьмеричное или 16-чное число в синтаксисе FASM (постфиксная форма записи). 16-чное число должно начинаться с цифры от 0 до 9.

• регулярное выражение;

Классы:

Digit ::= [0-9];

Digit\_0 ::= [0];

Digit\_1\_9 ::= [1-9];

Letters::= [a-f A-F];

Hex ::= [Hh];

Bin ::= [Bb];

Oct ::= [Oo];

Oct\_Digit ::= [0-7];

Bin\_Digit ::= [01];

Регулярное выражение:

Number ::= (Bin\_Digit+ Bin) | (Oct\_Digit+ Oct) | (Digit\_1\_9+ Letters+ Hex) | (Digit\_0+ Letters+ Digit+ Hex)

• НКА для этого регулярного выражения;



• эквивалентный ДКА и его таблицу переходов;

• программное средство, реализующее работу этого ДКА.

#include <stdio.h>

#include <string.h>

#define MAX\_PATH\_LENGTH 256

// Функция проверки, является ли символ цифрой

int is\_digit(char ch) {

return ch >= '0' && ch <= '9';

}

// Функция преобразования символа к верхнему регистру (для шестнадцатеричных чисел)

char to\_upper(char ch) {

if (ch >= 'a' && ch <= 'f') {

return ch - ('a' - 'A');

}

return ch;

}

// Функция проверки двоичного числа

int is\_valid\_binary(const char \*num) {

size\_t len = strlen(num);

if (len < 2 || num[len - 1] != 'B') {

return 0;

}

for (size\_t i = 0; i < len - 1; i++) {

if (num[i] != '0' && num[i] != '1') {

return 0;

}

}

return 1;

}

// Функция проверки восьмеричного числа

int is\_valid\_octal(const char \*num) {

size\_t len = strlen(num);

if (len < 2 || (num[len - 1] != 'o' && num[len - 1] != 'O')) {

return 0;

}

for (size\_t i = 0; i < len - 1; i++) {

if (num[i] < '0' || num[i] > '7') {

return 0;

}

}

return 1;

}

// Функция проверки шестнадцатеричного числа

int is\_valid\_hex(const char \*num) {

size\_t len = strlen(num);

if (len < 2 || (num[len - 1] != 'H' && num[len - 1] != 'h')) {

return 0;

}

if (!is\_digit(num[0])) { // Число должно начинаться с цифры

return 0;

}

for (size\_t i = 0; i < len - 1; i++) {

char upper\_char = to\_upper(num[i]);

if (!is\_digit(num[i]) && (upper\_char < 'A' || upper\_char > 'F')) {

return 0;

}

}

return 1;

}

// Функция проверки корректности числа в синтаксисе FASM

int is\_valid\_number(const char \*num) {

return is\_valid\_binary(num) || is\_valid\_octal(num) || is\_valid\_hex(num);

}

// Функция для обработки ввода чисел из файла

void process\_file\_input() {

char filename[MAX\_PATH\_LENGTH];

char number[MAX\_PATH\_LENGTH];

FILE \*file = NULL;

printf("Введите имя файла (в той же директории, что и .exe): ");

fgets(filename, MAX\_PATH\_LENGTH, stdin);

filename[strcspn(filename, "\n")] = 0; // Удаляем символ новой строки

file = fopen(filename, "r");

if (file == NULL) {

printf("Не удалось открыть файл %s. Убедитесь, что файл находится в той же директории.\n", filename);

return;

}

while (fgets(number, MAX\_PATH\_LENGTH, file) != NULL) {

number[strcspn(number, "\n")] = 0; // Удаляем символ новой строки

if (is\_valid\_number(number)) {

printf("Число корректно: %s\n", number);

} else {

printf("Число некорректно: %s\n", number);

}

}

fclose(file);

}

// Функция для ручного ввода чисел

void process\_manual\_input() {

char number[MAX\_PATH\_LENGTH];

printf("Введите число в синтаксисе FASM (или 'q' для выхода):\n");

int exit\_flag = 0;

while (!exit\_flag) {

printf("> ");

fgets(number, MAX\_PATH\_LENGTH, stdin);

// Убедимся, что строка не превышает допустимую длину

if (strchr(number, '\n') == NULL) {

printf("Строка слишком длинная. Попробуйте снова.\n");

// Очищаем буфер

while (getchar() != '\n');

continue;

}

// Удаляем символ новой строки, если он есть

number[strcspn(number, "\n")] = 0;

// Проверяем, хочет ли пользователь выйти

if (strcmp(number, "q") == 0) {

exit\_flag = 1;

} else {

if (is\_valid\_number(number)) {

printf("Число корректно.\n");

} else {

printf("Число некорректно. Попробуйте снова.\n");

}

}

}

}

// Основная функция программы

int main() {

char choice;

printf("Выберите режим ввода:\n");

printf("1. Считывать числа из файла\n");

printf("2. Вводить числа вручную\n");

printf("> ");

choice = getchar();

getchar(); // Игнорируем символ новой строки после выбора

if (choice == '1') {

process\_file\_input();

} else if (choice == '2') {

process\_manual\_input();

} else {

printf("Неверный выбор.\n");

return 1;

}

printf("Программа завершена.\n");

return 0;

}