

Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Фомин Иван Дмитриевич, № 22

Контакты email: grenka388@gmail.com,
Telegram: @Haliaven

Работа выполнена: «24» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «28» ____11____2022 г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.

2. **Цель работы:** Составить программу на Си, выполняющую анализ и обработку вводимого текста в соответствии с заданием.

3. **Задание:** Вариант 6. Подсчитать количество слов – правильных идентификаторов языков Pascal и C.

4. **Оборудование** (студента):

Процессор AMD Ryzen 7 4800HS with Radeon Graphics с ОП 8 Гб, SSD 512 Гб. Монитор 1920x1080

5. **Программное обеспечение** (студента):

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04 *jammy*

интерпретатор команд: *bash* версия 5.1.16(1)-release

Система программирования CLion версия 2021.1.3, редактор текстов *nano* версия 6.2

Утилиты операционной системы WinRar, Microsoft Word

Прикладные системы и программы Ubuntu wsl, CLion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере ~/home

6. Идея, метод, алгоритм

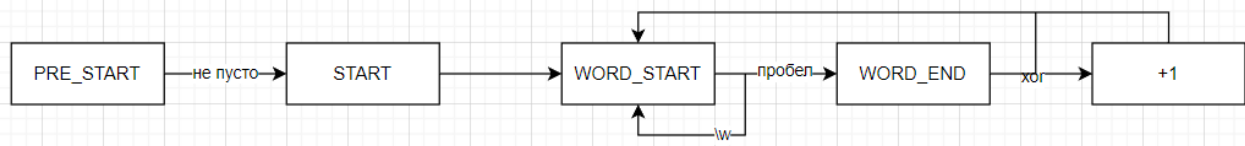
Сначала считываем слово. Словом я называю последовательность НЕ из специальных символов, ограниченная с двух сторон пробелом.

Считанное слово проверяем побуквенно, в зависимости от его длины, с зарезервированными словами из Си и Паскаля.

Если слово есть +1, если нет +0.

7. Сценарий выполнения работы

| Входные данные | Выходные данные | Описание тестируемого случая |
|---|-----------------|---|
| one two then three | 1 | Из всех слов then входит в искомые |
| prog ram | 1 | Проверка на игнор специальных символов |
| xor 872634 ugefuyg wiuehf case repeat g | 3 | Проверка, если вводятся подходящие слова разной длины |
| | 0 | Пустой ввод |



8. Распечатка протокола

```
#include <stdio.h>
#include <ctype.h>
#include <locale.h>
#include <assert.h>

const char answersLen3[1][3] = {
    "xor"
};

const char answersLen4[6][4] = {
    "case",
    "else",
    "then",
    "lock",
    "goto",
    "type"
};

const char answersLen5[2][5] = {
    "break",
    "using"
};

const char answersLen6[9][6] = {
    "repeat",
    "sizeof",
    "typeof",
    "static",
    "inline",
    "struct",
    "return",
    "signed",
    "extern"
};

const char answersLen7[2][7] = {
    "program",
    "default"
};

const char answersLen8[1][8] = {
    "register"
};

const int answersLenArr[9] = {
    0, 0, 0, 1, 6, 2, 9, 2, 1
};

const int escapeCharacters[6] = {
    '\n', '\t', '\r', '\v', '\n', '\b'
};

int isEqualStr(const char str1[], const char str2[], int lenEdge) {
    //printf("\n 1: %s \n 2: %s \n", str1, str2);

    for (int i = 0; i < lenEdge; ++i)
        if (str1[i] != str2[i]) return 0;

    return 1;
}
```

```

int isReservedWord(char currentCh[], int len) {
    switch (len) {
        case 3:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen3[i], len)) return 1;
            return 0;

        case 4:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen4[i], len)) return 1;
            return 0;

        case 5:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen5[i], len)) return 1;
            return 0;

        case 6:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen6[i], len)) return 1;
            return 0;

        case 7:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen7[i], len)) return 1;
            return 0;

        case 8:
            for (int i = 0; i < answersLenArr[len]; ++i)
                if (isEqualStr(currentCh, answersLen8[i], len)) return 1;
            return 0;

        default:
            return 0;
    }
}

int isSpecial(char input) {
    for (int i = 0; i < sizeof escapeCharacters; ++i)
        if (input == escapeCharacters[i]) return 1;

    return 0;
}

void resetCurrentCh(char current[10]) {
    for (int i = 0; i < 10; ++i)
        current[i] = 0;
}

void test_isSpecial() {
    assert(isSpecial('\n') == 1);
    assert(isSpecial('\r') == 1);
    assert(isSpecial('\t') == 1);
    assert(isSpecial('\v') == 1);
    assert(isSpecial('\\') == 0);
    assert(isSpecial('b') == 0);
}

void test_isReservedWord() {
    assert(isReservedWord("bebra", 5) == 0);
    assert(isReservedWord("xor", 3) == 1);
    assert(isReservedWord("babus", 5) == 0);
    assert(isReservedWord("program", 7) == 1);
}

```

```

    assert(isReservedWord("progr.m", 7) == 0);
    assert(isReservedWord("progr.m", 7) == 0);
    assert(isReservedWord("lock", 4) == 1);
}

void test_isEqualStr() {
    assert(isEqualStr("str1", "str2", 4) == 0);
    assert(isEqualStr("then", answersLen4[2], 4) == 1);
    assert(isEqualStr("register", answersLen8[0], 8) == 1);
    assert(isEqualStr("registered", answersLen8[0], 8) == 0);
    assert(isEqualStr("\nthen", answersLen5[0], 5) == 0);
}

void doAllTests() {
    test_isSpecial();
    test_isEqualStr();
    test_isReservedWord();
}

typedef enum {
    PRE_START,
    START,
    WORD_START,
    WORD_END
} State;

int main() {
    setlocale(LC_ALL, "ru_RU.UTF-8");
    setlocale(LC_NUMERIC, "C");

    doAllTests();

    char ch;
    char currentCh[10] = {-1};
    int len = 0;
    int res = 0;

    State currentState = PRE_START;

    /*
     * РЕГУЛЯРКА ДЛЯ ЗАДАЧИ
     * 1. Удалить все специальные знаки [a-zA-Z0-9 ]+
     * 2. Найти все подходящие слова \W(xor|break|using|program|...)\W
     */

    while ((ch = tolower(getchar())) != '1') {
        if (isSpecial(ch)) continue;
        if (ch == '\\0' && currentState != WORD_END && currentState != WORD_START)
            break;

        switch (currentState) {
            case PRE_START:
                ++currentState;

                if (ch == ' ') break;

            case START:
                currentCh[currentCh[0] == -1 ? 0 : 1] = ch;

                if (len == 0) len = 1;
                else ++len;

                ++currentState;

```

```

        break;

    case WORD_START:
        if (len > 10) {
            len = 0;
            currentState = START;
            break;
        }

        if (ch != ' ' && ch != '\0') {
            currentCh[len] = ch;
            ++len;
            break;
        } else {
            ++currentState;
            // mb delete
            break;
        }

    case WORD_END:
        printf("\n %s END \n", currentCh);

        res += isReservedWord(currentCh, len);

        resetCurrentCh(currentCh);
        currentCh[0] = ch;
        len = 1;

        currentState = START;
        break;
    }
}

printf("\nres: %d", res);

return 0;
}

```

9. Дневник отладки

| № | Лаб. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|---|---------------------|----------------------------|--------------|--------------------|----------------------------|--------------|
| 1 | дом | 18.11.2022 – 24.11.2022 | Весь день | Ничего не работает | Пытался дебажить | Ненавижу Си. |

10. Замечания автора

Ну, если честно, в этой лабе ничего хорошего нет, работа неактуальна, потому что я не умею пока писать нормальный (насколько это возможно для этого языка конечно) код на Си.

11. Выводы

Это было ужасно. После того, как я всю жизнь прогал на высокоуровневых ЯП (даже не питон), переходить на Си – это чистая боль. 80% всего времени выполнения лабы я сидел в гугле или смотрел на очередную ошибку, которую компилятор мне дружелюбно представил в виде 0x07987907347 потому что это очень понятно 🙌

Конечно, понятно, что нас обучают Си для того, чтобы мы поняли, как всё устроено изнутри, но это уже садизм. Пока того, что нам давали на лабах, было недостаточно, чтобы сделать это задание (в основном указатели, многомерные массивы).

В общем и целом, у меня горит. На этот раз без анекдота

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента _____