

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

Студент группы М8О-109Б-22 Нгуен Ньы Хоанг Ань, № по списку 12

Контакты vk, telegram @tng00

Работа выполнена: «20» ноября 2022 г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » 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:** Научиться обрабатывать последовательности литер входного текстового текста.

3. **Задание (вариант 19):** Перевести все мерные длины из дюймов (in) в миллиметры (mm). Например, 10in → 254mm.

4. **Оборудование** (студента):

Процессор Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz с ОП 5,88 Гб, НМД 1024 Гб. Монитор 1920x1080

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

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04.1 LTS

Интерпретатор команд: *bash* версия 5.1.16.

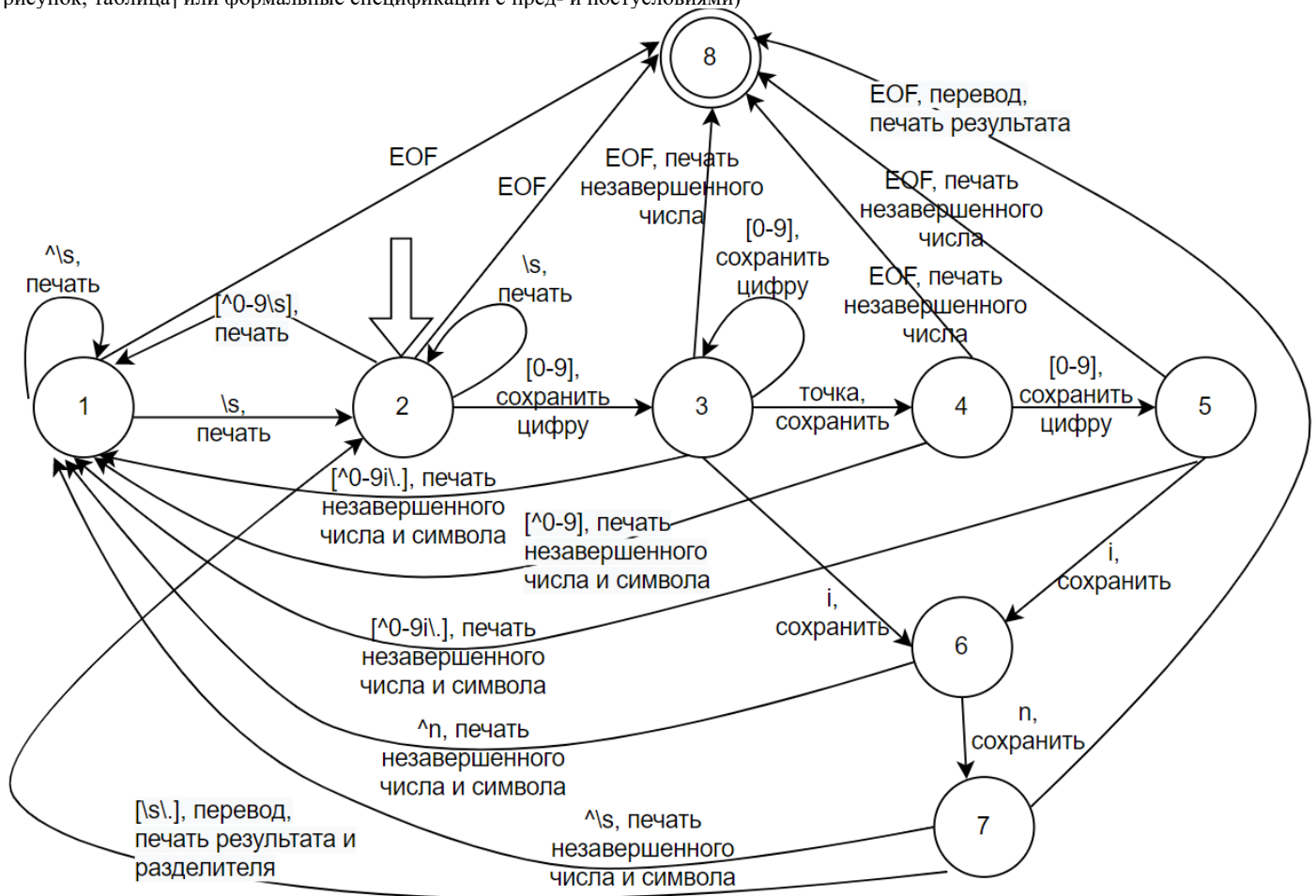
Система программирования -- версия --, редактор текстов *emacs* версия 28.2

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере /home/tng00

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)



7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

1. Изучить материалы лекции
2. Составить алгоритм действий, переходов состояний
3. Написать программу, удовлетворяющую условиям лабораторной работы
4. Провести unit-тестирование
5. Запротолировать сеанс

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
tng00@DESKTOP-QIGNL3K:~/lab11$ cat lab11.c
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

#define MM_IN_INCH 25.4

int IsDigit(char c) {
    return c >= '0' && c <= '9';
}

int IsSeparator(char c) {
    return c == ' ' || c == '\n' || c == '\t' || c == ',' || c == EOF;
}

struct Inch {
    char *num;
    size_t size;
};

void InitInch(struct Inch *inch) {
    inch->num = NULL;
    inch->size = 0;
    assert(inch->num == NULL && inch->size == 0);
}

void AddDigitToInch(struct Inch *inch, char c) {
    ++inch->size;
    if (inch->num == NULL) {
        inch->num = malloc(sizeof(char));
    } else {
        inch->num = realloc(inch->num, sizeof(char) * inch->size);
    }
    inch->num[inch->size - 1] = c;
    assert(inch->num[inch->size - 1] == c);
}

void ResetInch(struct Inch *inch) {
    free(inch->num);
    inch->num = NULL;
    inch->size = 0;
    assert(inch->num == NULL && inch->size == 0);
}

void Final(struct Inch *inch) {
    double res = strtod(inch->num, NULL) * MM_IN_INCH;
    printf("%gmm", res);
    assert(res == strtod(inch->num, NULL) * MM_IN_INCH);
    ResetInch(inch);
    assert(inch->num == NULL && inch->size == 0);
}

void PrintUnfinishedInch(struct Inch *inch) {
    printf("%s", inch->num);
    ResetInch(inch);
    assert(inch->num == NULL && inch->size == 0);
}

typedef enum {
    SEARCH, FIRST, DOT, SECOND, CHAR_I, CHAR_N
} State;

void UnitTests() {
```

```
assert(IsDigit('c') == 0);
assert(IsDigit('9') == 1);
assert(IsSeparator('r') == 0);
assert(IsSeparator('\t') == 1);
};
```

```
int main() {
    UnitTests();
    char c;
    char last_c = EOF;
    struct Inch inch;
    InitInch(&inch);
    State state = SEARCH;

    while ((c = getchar()) != EOF) {

        switch (state) {
            case SEARCH:
                if (IsDigit(c) && IsSeparator(last_c)) {
                    AddDigitToInch(&inch, c);
                    state = FIRST;
                } else {
                    printf("%c", c);
                }
                break;

            case FIRST:
                if (IsDigit(c) && inch.size > 0) {
                    AddDigitToInch(&inch, c);
                } else if (c == '.' && inch.size > 0) {
                    state = DOT;
                    AddDigitToInch(&inch, c);
                } else if (c == 'i' && inch.size > 0) {
                    state = CHAR_I;
                    AddDigitToInch(&inch, c);
                } else {
                    PrintUnfinishedInch(&inch);
                    printf("%c", c);
                    state = SEARCH;
                }
                break;

            case DOT:
                if (IsDigit(c) && inch.size > 0) {
                    AddDigitToInch(&inch, c);
                    state = SECOND;
                } else {
                    PrintUnfinishedInch(&inch);
                    printf("%c", c);
                    state = SEARCH;
                }
                break;

            case SECOND:
                if (IsDigit(c) && inch.size > 0) {
                    AddDigitToInch(&inch, c);
                } else if (c == 'i') {
                    AddDigitToInch(&inch, c);
                    state = CHAR_I;
                } else {
                    PrintUnfinishedInch(&inch);
                    printf("%c", c);
                    state = SEARCH;
                }
                break;

            case CHAR_I:
                if (c == 'n') {
```

```

        AddDigitToInch(&inch, c);
        state = CHAR_N;
    } else {
        PrintUnfinishedInch(&inch);
        printf("%c", c);
        state = SEARCH;
    }
    break;

case CHAR_N:
    if (IsSeparator(c) || c == '.') {
        Final(&inch);
        printf("%c", c);
    } else {
        PrintUnfinishedInch(&inch);
        printf("%c", c);
    }
    state = SEARCH;
    break;

default:
    break;
}

last_c = c;
}
}
tng00@DESKTOP-QIGNL3K:~/lab11$ cat test1.txt
12in, .9in 2.5in 6inn 5.7in 7n 0in. bla-bla-bla 89.9in

```

```

tng00@DESKTOP-QIGNL3K:~/lab11$ cc lab11.c
tng00@DESKTOP-QIGNL3K:~/lab11$ ./a.out < test1.txt
304.8mm, .9in 63.5mm 6inn 144.78mm 7n 0mm. bla-bla-bla 2283.46mm

```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы

11. Выводы

Выяснил, что из себя представляет конечный автомат. Научился обрабатывать последовательность литер, используя язык Си, познакомился с диаграммами состояний и переходов, их использованием для решения задач.

Подпись студента

