

Отчет по лабораторной работе № 11 по курсу Архитектура компьютера и информационных сетей

Студент группы М8О-103Б-22 Клименко Виталий Максимович, № по списку 11

Контакты www, e-mail, icq, skype vitalikklimenko96@gmail.com

Работа выполнена: 14 ноября 2022 г.

Преподаватель: доцент Никулин С.П.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ 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 языке Си, выполняющую анализ и обработку вводимого текста.

3. **Задание (вариант № 29):** Удалить десятичные числа, не превышающие INT_MAX

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel 4x 3.5GHz с ОП 16 Гб НМД HDD 200 Гб . Монитор Встроенный 1920x1080
Другие устройства Touchpad Synaptics

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

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Pop!_OS версия 22.04 jammy
интерпретатор команд bash версия 5.1.16
Система программирования GNU Compiler Collection (GCC) версия 11.2.0
Редактор текстов Visual Studio Code версия 1.72.2
Утилиты операционной системы _____

Прикладные системы и программы gcc, gdb

Местонахождение и имена файлов программ и данных на домашнем компьютере _____

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

Читать числа посимвольно для будущего корректного перевода строки в число и для обработки конца файла

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

Создать файл со своей структурой string, подобную string из Си++, там создать несколько функций для упрощенного взаимодействия с кодом как для программиста так и для проверяющего. Написать файл, который будет выполнять выданное задание

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```
vitos@vitos-pop:~/Studying/mai/labs/l11$ cat makefile
all:
    gcc -std=c99 -g -Wall -Wextra -o main main.c string.c
    ./main
vitos@vitos-pop:~/Studying/mai/labs/l11$ cat string.h
#ifndef STRING_H
#define STRING_H

#include <stdio.h>
#include <stdlib.h>
#include <inttypes.h>

typedef struct string {
    char *values;
    uint64_t last_element;
    uint64_t memory_size;
} string;

// Initialize string with default capacity
int init_string(string* s);

// Sets all struct's parameters to zero
int free_string(string* s);

// Adds memory
int string_resize(string* s, uint64_t capacity);

// Adds a char to string, adding more memory if needed
int add_char(string* s, char value);

// Reads string char by char returning 1 if EOF
int read_string(string* s);

#endif
vitos@vitos-pop:~/Studying/mai/labs/l11$ cat string.c
#include "string.h"

// Maybe deprecated, but those are for error handling
#define UNDEFINED 0
#define SUCCESS 1
#define ERROR -1

// How big initial memory is
const uint64_t INIT_CAPACITY = 2;
// How big relative to previous allocated memory new memory should be
const uint64_t EXTENDED_CAPACITY = 2;

// Memory-related macros
#define new(a, n) (a*)calloc(n, sizeof(a) * n)
#define reallocate(ptr, a, n) (a*)realloc(ptr, sizeof(a) * n)

#define max(a, b) (a > b ? a : b)
#define min(a, b) (a < b ? a : b)

// Initialize string with default capacity
int init_string(string* s) {
    int result = UNDEFINED;
    s->memory_size = INIT_CAPACITY;
    s->values = new(char, INIT_CAPACITY);
    s->last_element = 0;
    return result;
}

// Sets all struct's parameters to zero
int free_string(string* s) {
    s->memory_size = INIT_CAPACITY;
    s->values = new(char, 0);
    s->last_element = 0;
    return SUCCESS;
}

// Adds memory
int string_resize(string* s, uint64_t capacity) {
    uint64_t new_capacity = s->memory_size * capacity;
    s->memory_size = new_capacity;
    s->values = reallocate(s->values, char, new_capacity);
}
```


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 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: Я научился писать программы на Си связанные с обработкой литер из терминала. По пути реализовывания данной лабораторной работы я также начал изучение хэдэр файлов, структур данных и мейкфайлов для Си

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента