



Отчёт по лабораторной работе № 8-10 по курсу Фундаментальная информатика

Студент группы М8О-101БВ-24 Волков Алексей Александрович № по списку 3

Контакты www, e-mail, icq, skype volckovl3sch@yandex.ru

Работа выполнена: « » 20 г.

Преподаватель: каф.806

Входной контроль знаний с оценкой

Отчёт сдан « » 201 г., итоговая оценка

Подпись преподавателя

1. **Тема:** Построение многоуровневой программы, её поэтапная сборка и отладка

2. **Цель работы:** Изучение конкретной системы программирования в С и получение навыков подготовки и отладки программы

☐ 3. **Задание (вариант №):** Составить программу из нескольких файлов, содержащую прямую и косвенную рекурсию, продемонстрировать четыре этапа сборки и работу отладчика

4. **Оборудование(лабораторное):**
ЭВМ , процессор , имя узла сети с ОП Мб,
НМД Мб. Терминал адрес . Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор с ОП Мб, НМД Мб. Монитор
Другие устройства

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 , 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 на домашнем компьютере

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].

Создать несколько файлов исходного кода: main.c, funcs.c и заголовочный файл funcs.h

Ошибки на этапе препроцессинга:

- В main.c добавить несуществующий заголовочный файл #include "non_exist.h".
- В main.c внести ошибку в директиву препроцессора (например, #define VALUE 10).
- Исправить найденные ошибки.

Ошибки на этапе компиляции:

- В main.c допустить синтаксическую ошибку (убрать точку с запятой после printf).
- В main.c допустить ошибку несоответствия типов (умножение int на строку).
- Исправить ошибки (добавить точку с запятой, заменить строку на числовой литерал).

Ошибки на этапе ассемблирования:

- В main.c вставить некорректные инструкции (asm("move eax, 0")).
- В main.c вставить некорректную вставку (asm("fuhqwio"))
- Исправить ассемблерные инструкции на корректные.

Ошибки на этапе линковки:

- В main.c и funcs.c объявить переменную с одинаковым именем, что приведёт к конфликту (например, int twice = 10; и int twice = 20;).
- В funcs.c изменить сигнатуру функции, чтобы она не соответствовала объявлению в funcs.h.
- Исправить ошибки (удалить дубликат переменной или использовать extern, выровнять сигнатуру функции).

gdb:

- Запустить: gdb ./prog.
- Установить точки останова с помощью break, например break main.
- Использовать команды run, step, next, print, backtrace для пошагового анализа работы программы.
- Продемонстрировать разницу между step и next.

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).

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

Реализовал программу, состоящая из нескольких файлов и содержащая в себе как прямую, так и косвенную рекурсию

Работа подразумевает изучение этапов сборки программы, которые могут содержать как предупреждения, с которыми программа может запускаться, так и ошибки, которые не дают провести окончательную сборку, а значит запрещают дальнейший её запуск

Реализации точек остановки программы с помощью команды break, является одним из наиболее популярных способов нахождения ошибок в коде программы

Благодаря использованию команд step и next наглядно продемонстрированы возможности пошаговой отладки, а анализ стека вызовов позволил определить проблемы в рекурсивных функциях.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента

Волк