Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и компьютерных технологий

**Отчёт по лабораторной работе № 4**

Дисциплина: Низкоуровневое программирование.

Тема: раздельная компиляция

Выполнил студент гр. 3530901/10003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Рубцов Е.А.

(подпись)

Принял старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Коренев Д.А.

(подпись)

“ ” 2022 г.

Санкт-Петербург

2022

Оглавление.

ТЗ………………………………………………………………………….......3 стр.

Метод решения………………………………………………………………3 стр.

Листинг программ………………………………………...………….…...…4 стр.

Препроцессирование…………………………………………………....…...6 стр.

Компиляция………………………………………………………....………..8 стр.

Ассемблирование……………………………...……………………………11 стр.

Компоновка…..……………………………………………………………...19 стр.

Создание статической библиотеки и make файлов………………………20 стр.

Вывод……………………………………………………………………..…21 стр.

Формулировка задачи**:**

1) На языке C разработать функцию, реализующую определенную вариантом задания функциональность. Поместить определение функции в отдельный исходный файл, оформить заголовочный файл. Разработать тестовую программу на языке C.

2) Собрать программу «по шагам». Проанализировать выход препроцессора и компилятора. Проанализировать состав и содержимое секций, таблицы символов, таблицы перемещений и отладочную информацию, содержащуюся в объектных файлах исполняемом файле.

3) Выделить разработанную функцию в статическую библиотеку. Разработать make-файлы для сборки библиотеки и использующей ее тестовой программы. Проанализировать ход сборки библиотеки и программы, созданные файлы зависимостей.

Вариант задания**:** объединить два отсортированных массива в один

# Метод решения:

Создаются итераторы для обоих массивов, а также указатель на выходной массив. Сравниваются элементы из обоих массивов, в выходной массив копируется наибольший элемент, после чего увеличивается итеретор для массива, из которого было скопировано заначение. После того, как один из итераторов достигнет значения размера соответствующего массива, программа копирует оставшиеся элементы из другого массива и завершает выполнение.

Листинг программ**:**

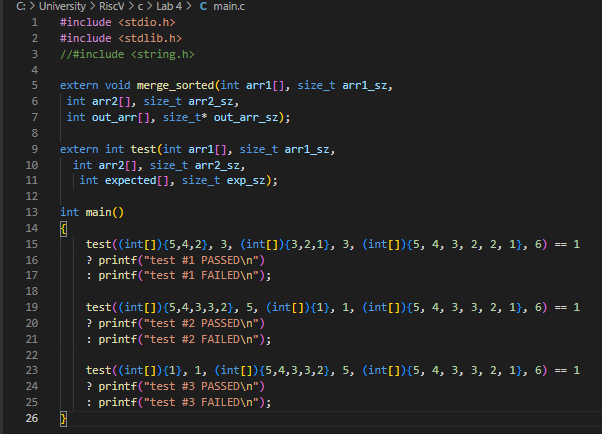


Рис.1 файл *main.c*

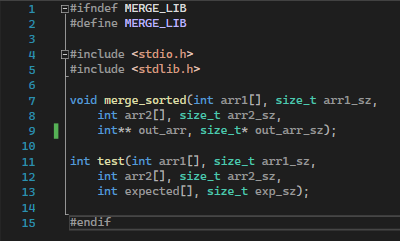
**

Рис.2 файл *merge\_l.h*

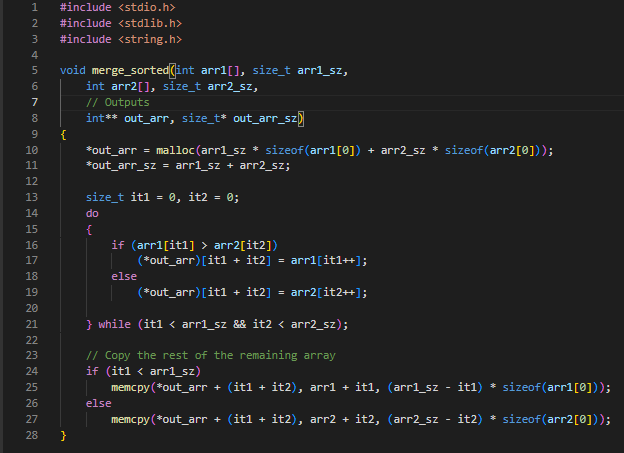


Рис.3 файл array\_merge.c

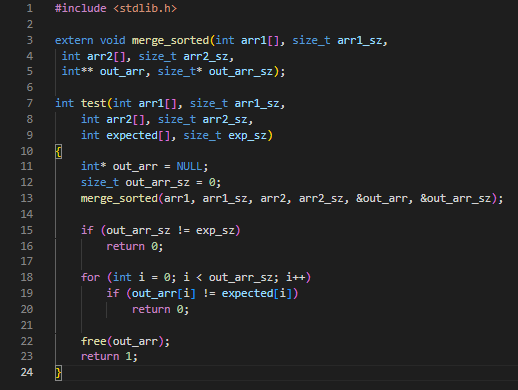


Рис.4 файл test\_merge*.c*

В файле array\_merge реализована функция merge\_sorted, в которую передуются два входных массива, их размеры, а также указатель на выходной массив

В файле test\_merge описана функция test, которая реализовывает тестирование функции merge\_sorted.

Препроцессирование**:**

Начнем сборку программы на C по шагам. Первый этап – препроцессирование файлов с исходным кодом в файлы “main.i”, “test\_merge” и “array\_merge.i”.

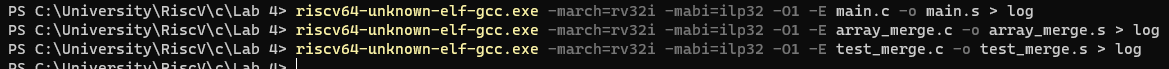


Рис.5 комманды для препроцессирования

Результаты препроцессирования:

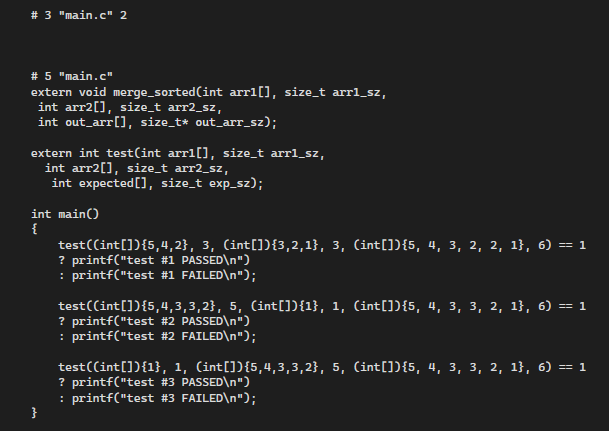


Рис.6 файл main.i

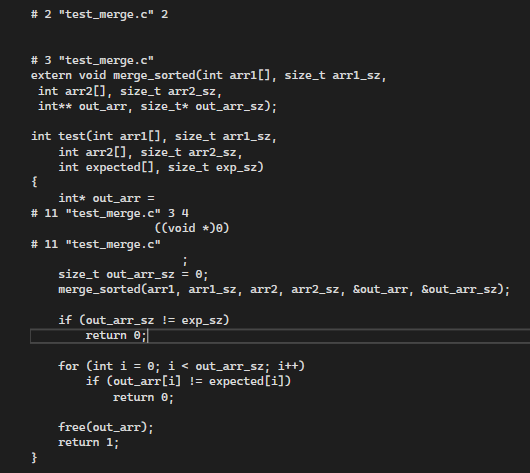


Рис.7 файл test\_merge*.i*

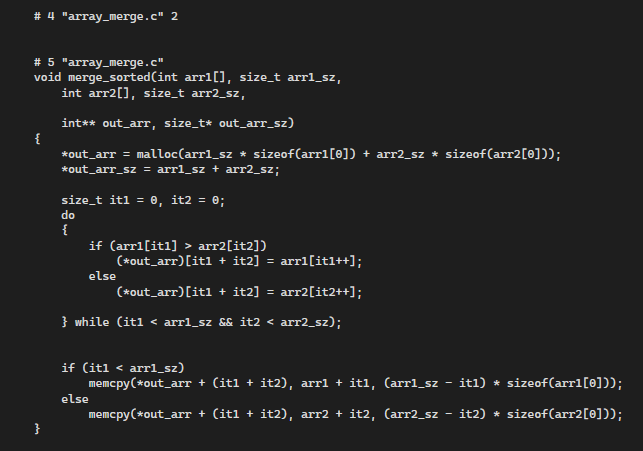


Рис.8 файл array\_merge*.i*

Компиляция**:**

Следующим шагом является компиляция файлов *.i* в текст на языке ассемблера



Рис.9 комманды для компиляции

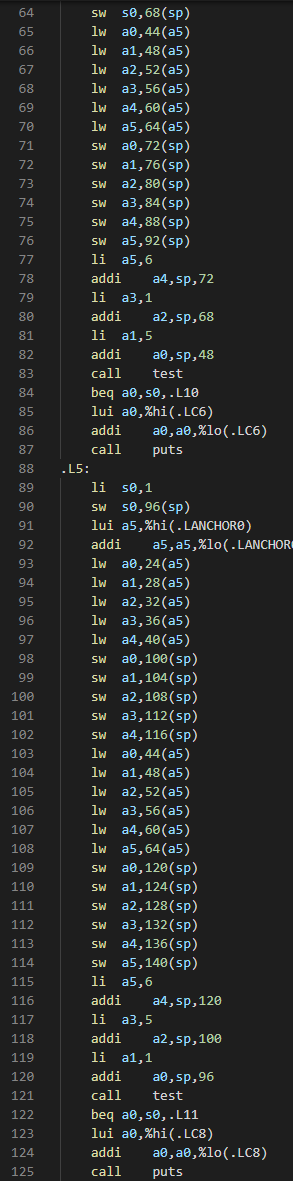
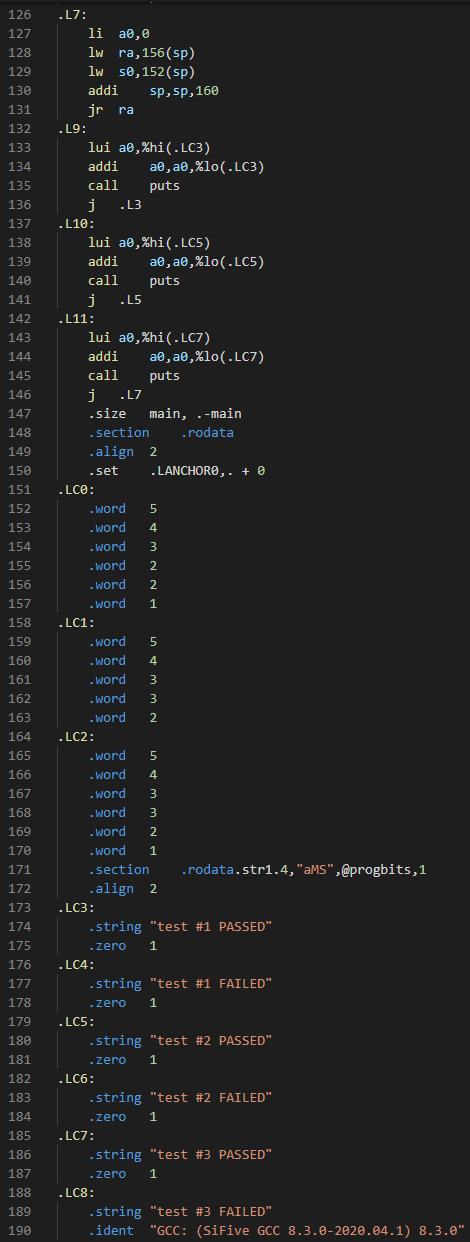
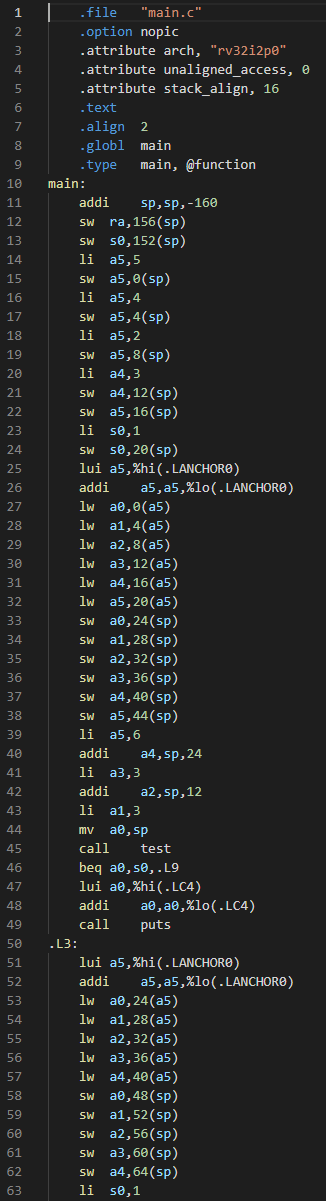
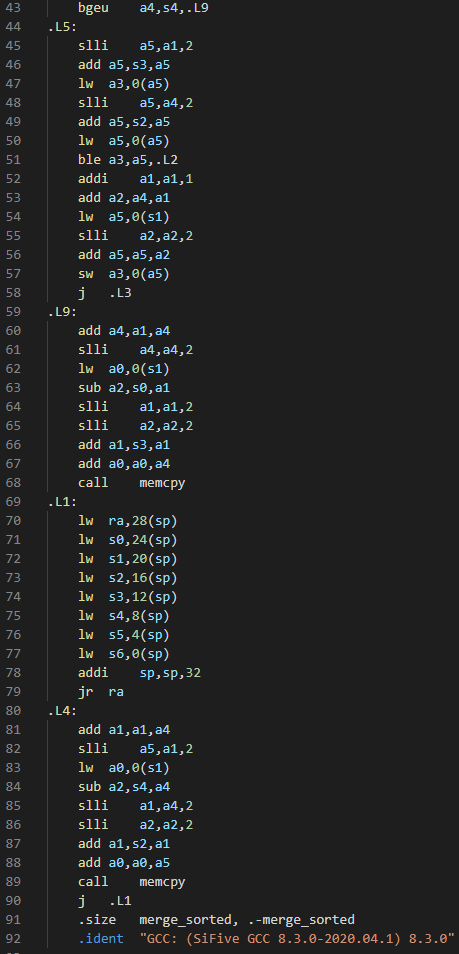
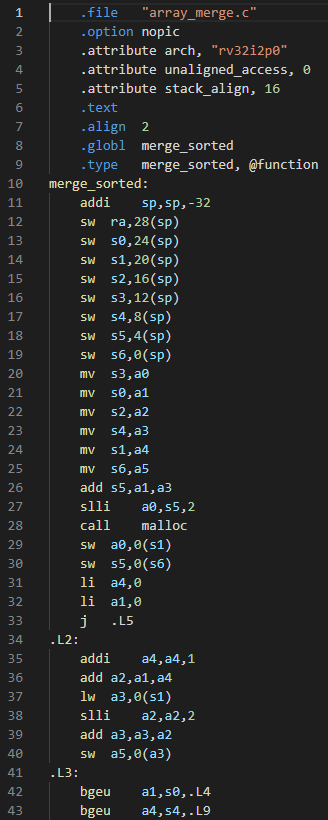
Результаты компиляции файлов:

Рис. 10 файл *main.s*

****Рис. 11 файл *array\_merge.s*

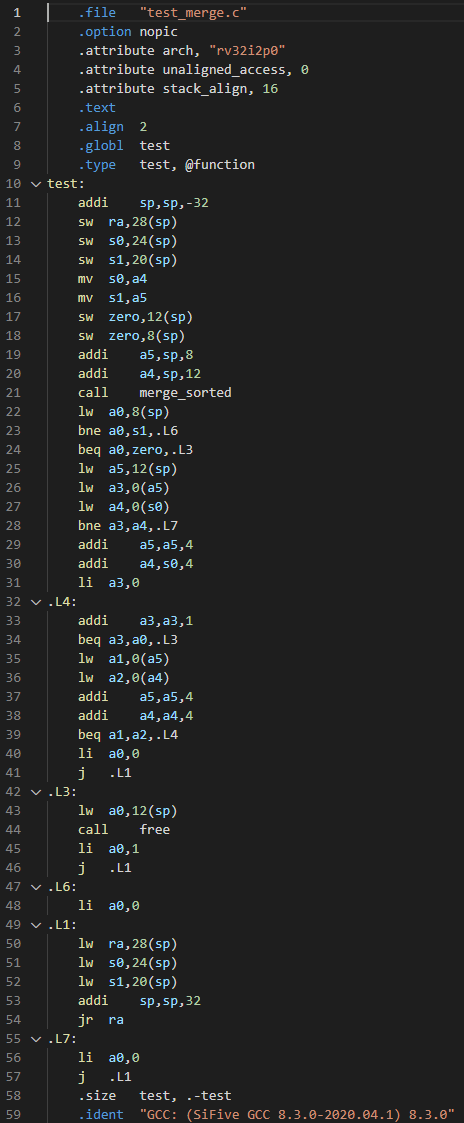
****

Рис. 12 файл *test\_merge.s*

Ассемблирование:



Рис.13 комманды для ассемблирования

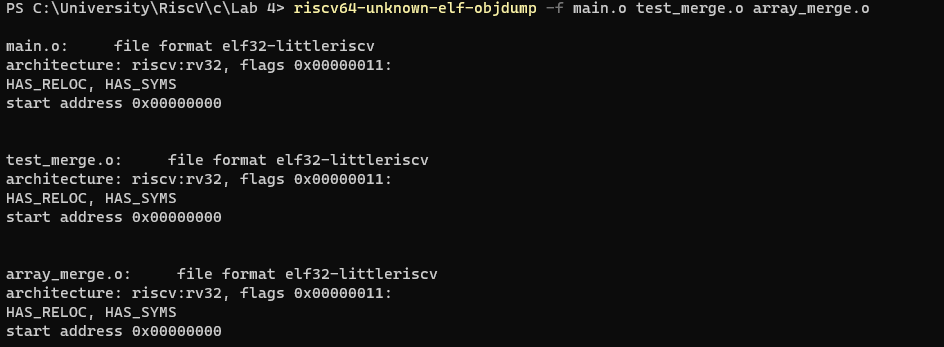


Рис.14 вывод информации о файлах

Файлы содержат символы (флаг HAS\_SYMS), содержат таблицу перемещений (флаг HAS\_RELOC). Объектный файл не должен содержать адрес, с которого начинается исполнение программы, такого же формата и исполняемый файл, поэтому здесь это поле равно нулю.

Заголовки файлов:

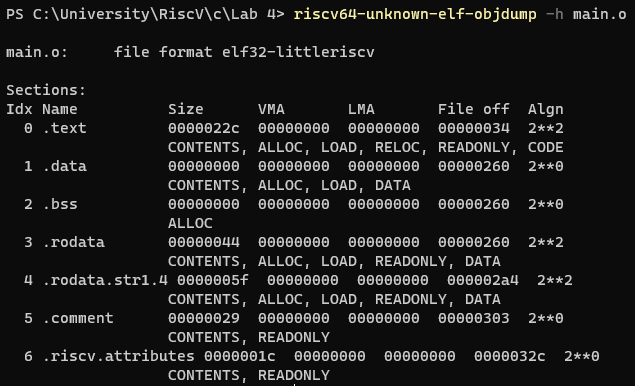


Рис.15 секция заголовком для файла main.o

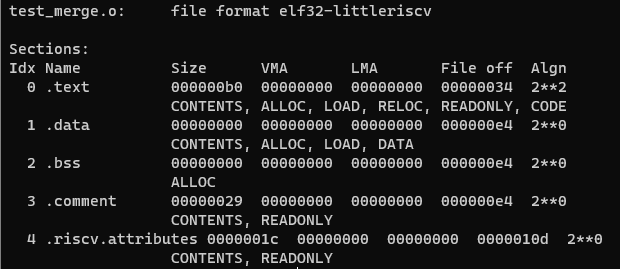


Рис.16 секция заголовком для файла test\_merge.o

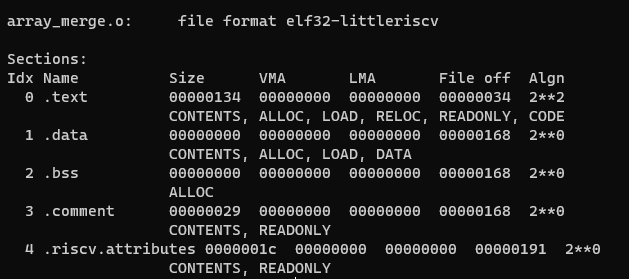


Рис.17 секция заголовком для файла array\_merge.o

Таблицы символов файлов:

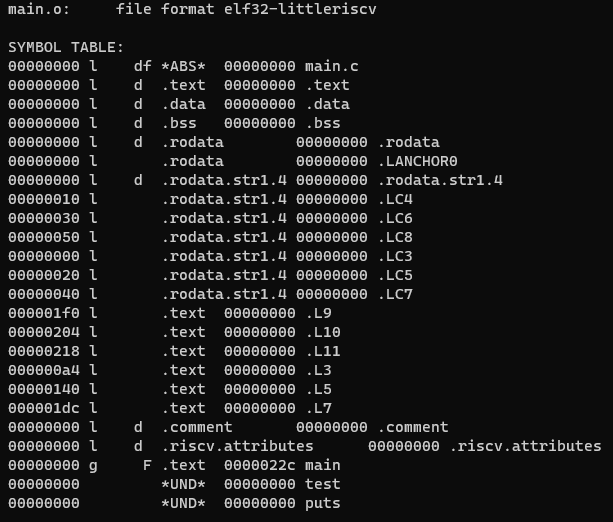


Рис.18 таблица символов для файла main.o

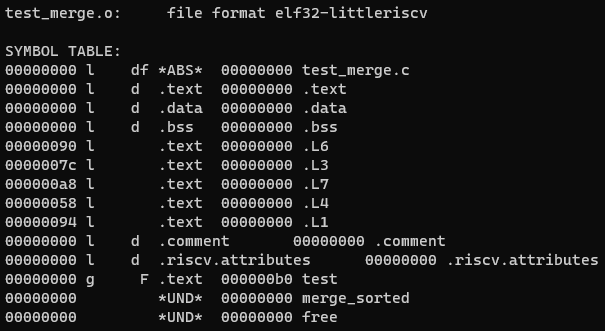


Рис.19 таблица символов для файла test\_merge.o

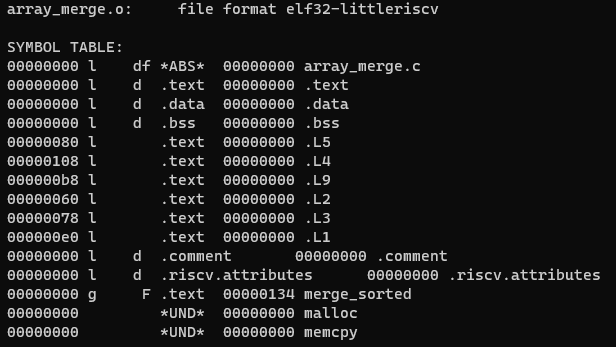


Рис.20 таблица символов для файла array\_merge.o

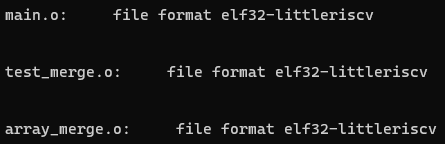


Рис.21 содержание секций .bss для всех файлов

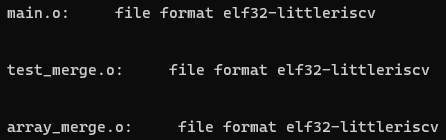


Рис.22 содержание секций .data для всех файлов

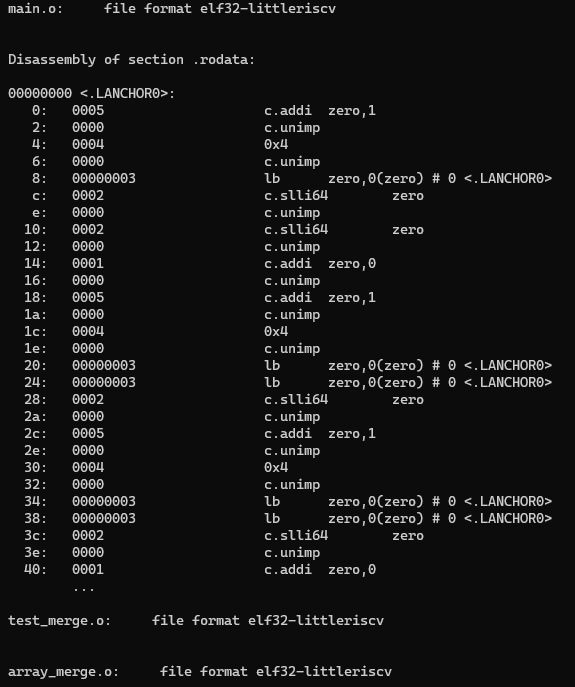


Рис.23 содержание секций .rodata для всех файлов

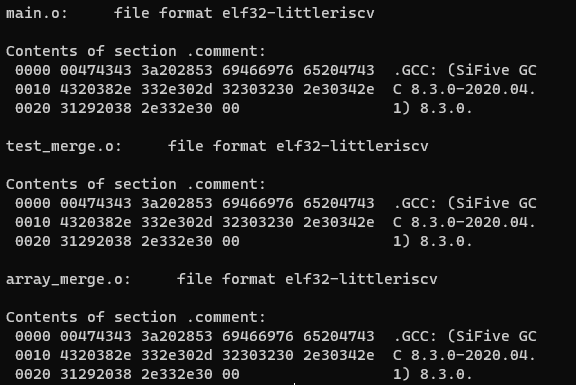


Рис.24 содержание секций .comment для всех файлов

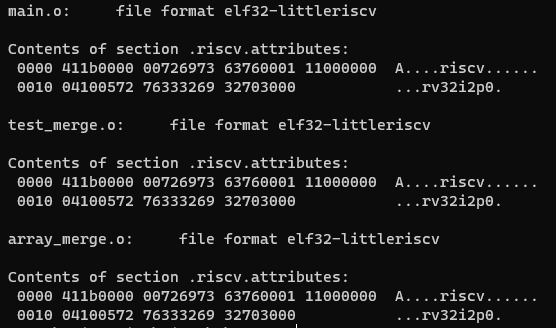


Рис.25 содержание секций .riscv.attributes для всех файлов

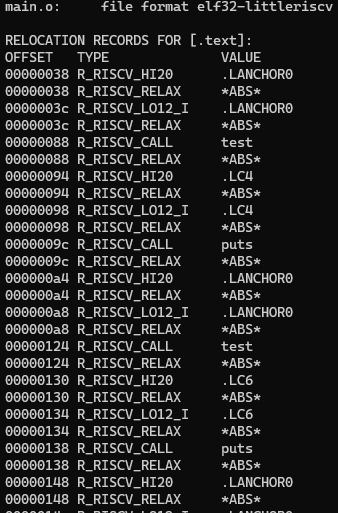
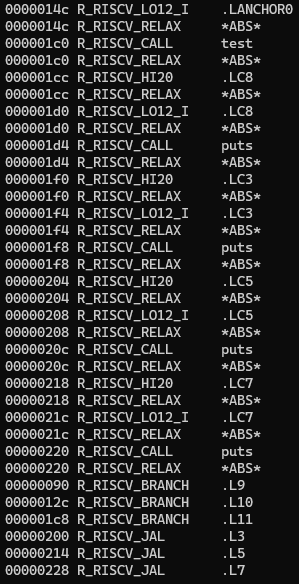
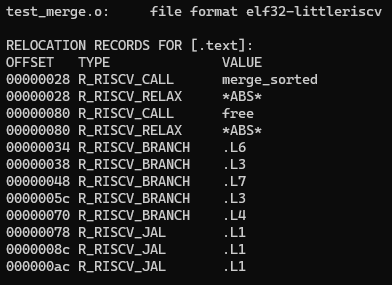
Рис.26 таблица перемещений для файла main.o

Рис.27 таблицы перемещений для файлов array\_merge.o и test\_merge.o

Компоновка**:**



Рис.28 команда для компоновки объекстных файлов

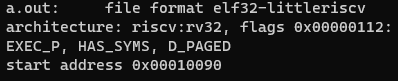


Рис.29 информация о файле a.out

Флаг EXEC\_P указывает, что файл является исполняемым, после загрузки его выполнение должно начаться с адреса 0x00010090

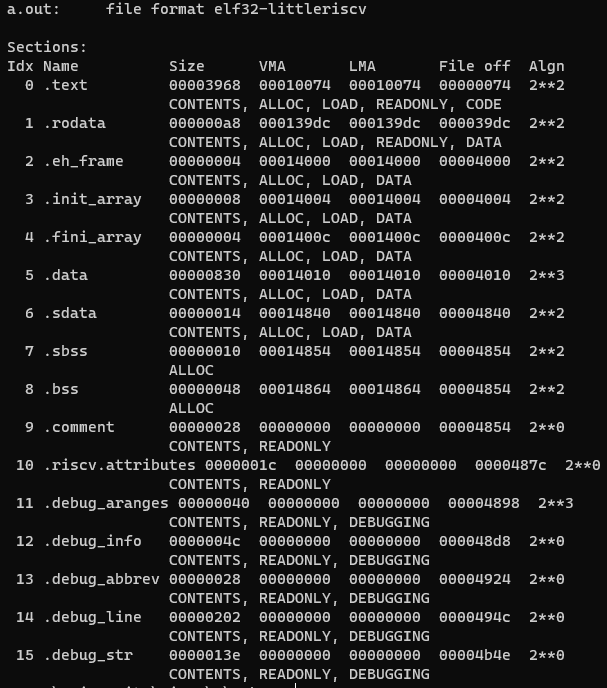
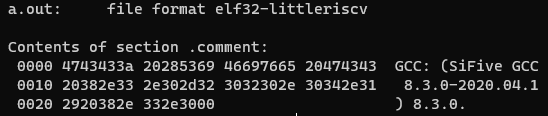
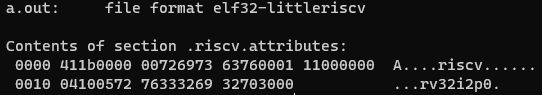


Рис.30 заголовки секций файла a.out

В исполняемом файле производится слияние содержания секций обоих объектных файлов, а также значительное расширение списка секций новыми блоками.





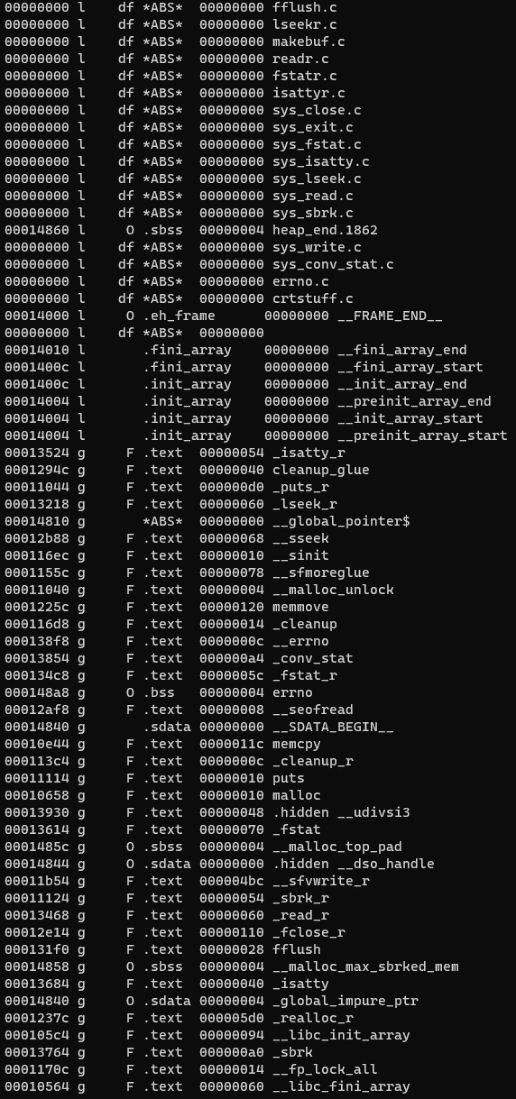
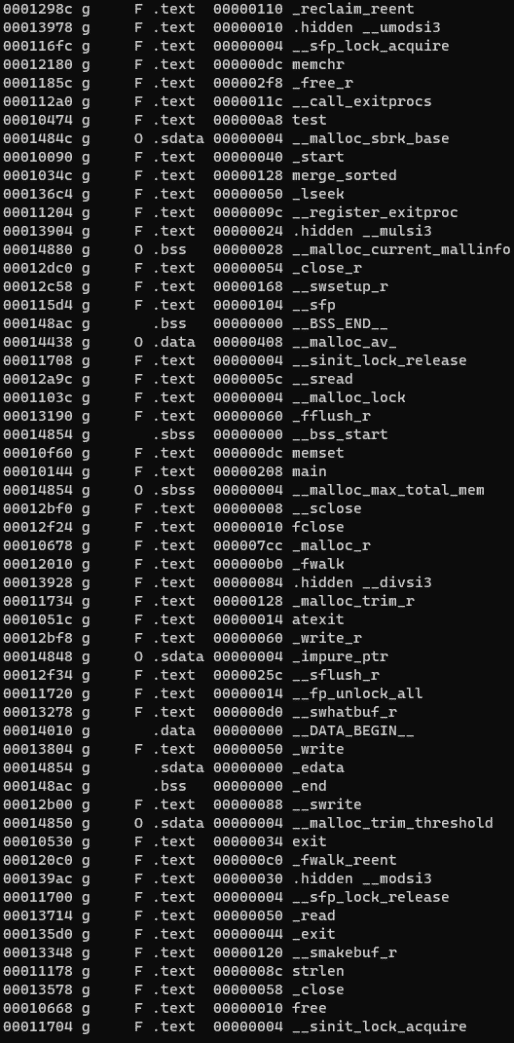
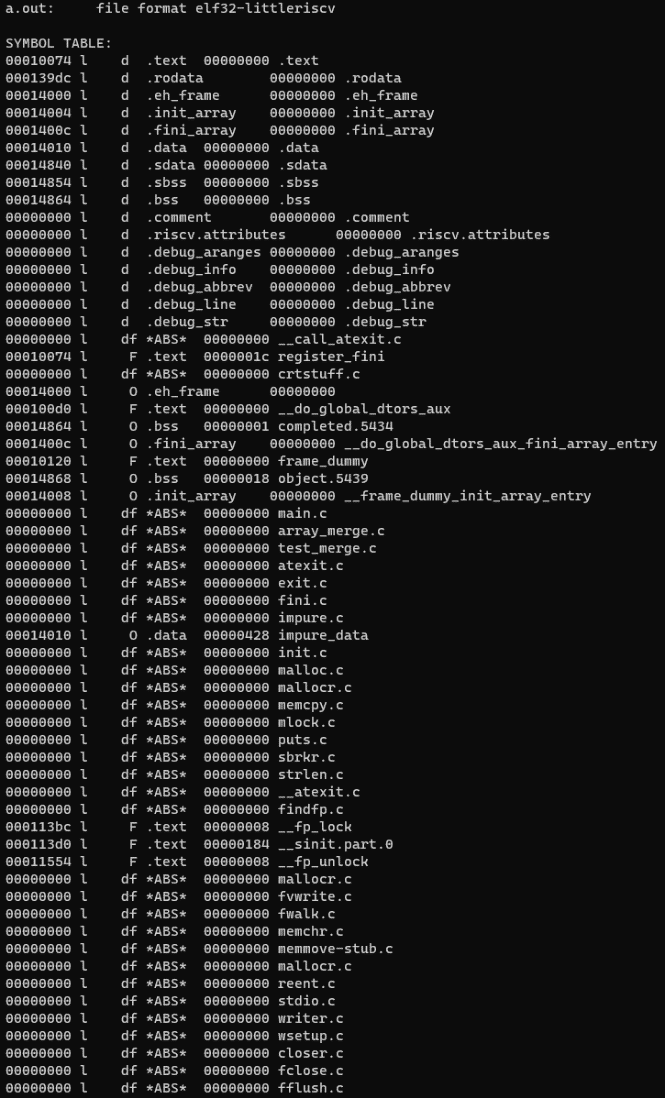
Рис. 31 содержание секций *.comment* и *.riscv.attributes* в файле *a.out*

Рис. 32 таблица символов *a.out*

Таблица перемещений файла пуста, все нееобходимые перемещения были записаны в код компоновщиком

Формирование статической библиотеки, разработка make-файлов для сборки библиотеки**:**

Статическая библиотека – это архив объектных файлов, каждый из которых может быть импортирован компоновщиком отдельно, если он неообходим для компоновки программы.

Создание статической бибилиотеки из файлов array\_merge.o и test\_merge.o:

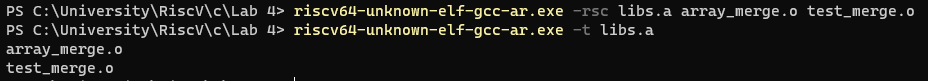


Рис. 33 сборка библиотеки

Используя получившуюся библиотеку, соберем исполняемый файл:



Рис. 33 сборка исполняемого файла

Процесс формирования статической библиотеки и сборки исполняемого файла можно автоматизировать при помощи утилиты make. Для этого понадобится создать Makefile:

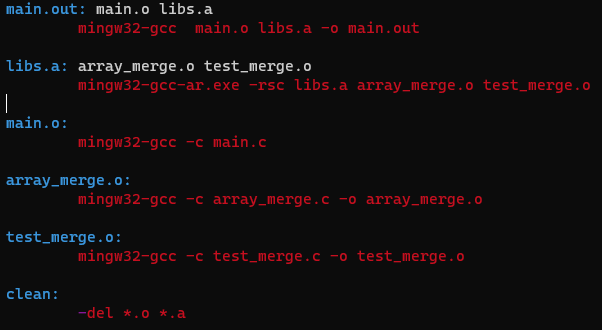


Рис. 34 текст файла “Makefile”

Вывод**:**

В ходе выполнения работы была изучена пошаговая компиляция программы на языке С. Также была создана статическая библиотека и произведена сборка программы с помощью Makefile.