Санкт-Петербургский политехнический университет Петра Великого

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 4**

**Дисциплина:** Низкоуровневое программирование

**Тема:** Раздельная компиляция

Выполнил студент гр. 3530901/90004 Пудов Н. А.

(подпись)

Преподаватель Алексюк А.О.

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

**Содержание**

[1. ТЗ 3](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506730)

[2. Метод решения 3](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506731)

[3. Ход решения 3](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506732)

[3.1 Листинги программ на С 3](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506733)

[3.2 Сборка программ «по шагам», анализ промежуточных и результирующих файлов 4](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506734)

[3.3 Формирование статической библиотеки, разработка make-файлов для сборки библиотеки 27](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506735)

[4. Выводы 31](file:///D:\Закачки\Zotov_lab_4.docx#_Toc69506736)

# **1. ТЗ**

1) На языке C разработать функцию, выполняющую *вывод члена Фибоначчи по заданному номеру*. 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 и 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.

**2. Метод решения**

**Алгоритм решения:**

Алгоритм решения задачи достаточно прост и был описан в прошлых отчётах, поэтому в дополнительных комментариях не нуждается.

**3. Ход решения**

**3.1 Листинги программ на С**

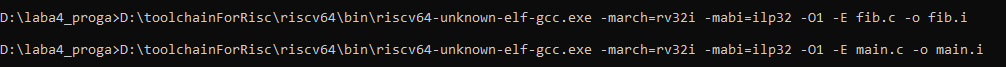
|  |
| --- |
|  |
| Рис.1. ''fib.h'' |

|  |
| --- |
|  |
| Рис.2. ''fib.c'' |

|  |
| --- |
|  |
| Рис.3. ''main.c'' |

**3.2 Сборка программ «по шагам», анализ промежуточных и результирующих файлов**

Приступим к пошаговой сборке написанных на языке C программ. Сначала препроцессируем исходные файлы fib.c и main.c в файлы fib.i и main.i:



Драйвер компилятора gcc riscv64-unknown-elf-gcc запускается с параметрами командной строки «-march=rv32i» и «-mabi=ilp32», которые указывают, что целевым является процессор с базовой архитектурой системы команд RV32I. Параметр «-O1» указывает, что надо выполнять простые оптимизации генерируемого кода, параметр «-E» – что надо остановить процесс сборки после препроцессирования.

Ниже представлены листинги полученных .i-файлов:

|  |
| --- |
|  |
| Рис.4. ''fib.i'' |

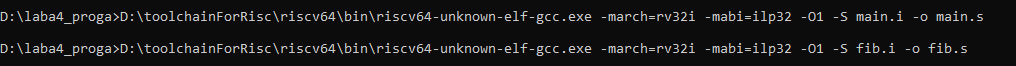
|  |
| --- |
|  |
| Рис.5. 1 часть ''main.i'' |

Между 1 и 2 частями файла “main.i’’ находится примерно 1600 строк кода с инструкциями по линковке имеющейся в файле “main.c” директивы “#include <stdio.h>” и “#include stdlib.h”.

|  |
| --- |
|  |
| Рис.6. 2 часть ''main.i'' |

Появившиеся нестандартные директивы, начинающиеся с символа “#”, используются для передачи информации об исходном тексте из препроцессора в компилятор. В файле “main.i” четвертая директива «# 1 “main.c”» информирует компилятор о том, что следующая строка является результатом обработки первой строки исходного файла “main.c”, где была прописана команда “#include “fib.h”, поэтому препроцессор произвел вставку содержимого этого заголовочного файла, то есть определение функции fib. Описание самого содержимого файла начинается после директивы “# 5 “main.c”. Исходный код тестирующей функции main после работы препроцессора остался без изменений, также как и исходный код функции fib из файла fib.i.

Далее проведем компиляцию файлов fib.i и main.i в код на языке ассемблера fib.s и main.s:



Драйвер компилятора riscv64-unknown-elf-gcc запускается с параметрами, где параметр “-S” указывает на необходимость остановить процесс сборки после компиляции.

|  |
| --- |
|  |
| Рис.7. ''fib.s'' |

Как мы можем заметить, начиная с метки fib начинается тело функции, выполняющей нахождение числа Фибоначчи по его номеру в последовательности.

|  |
| --- |
|  |
|  |
| Рис.8. ''main.s'' |

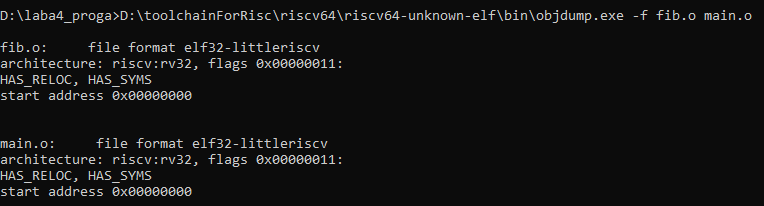
По метке main начинается тестовая программа, которая внутри себя уже вызывает функцию fib, однако эта функция ещё не определена, так как еще не было проведено ассемблирование.

Далее проведем ассемблирование файлов fib.s и main.s в объектные файлы fib.o и main.o



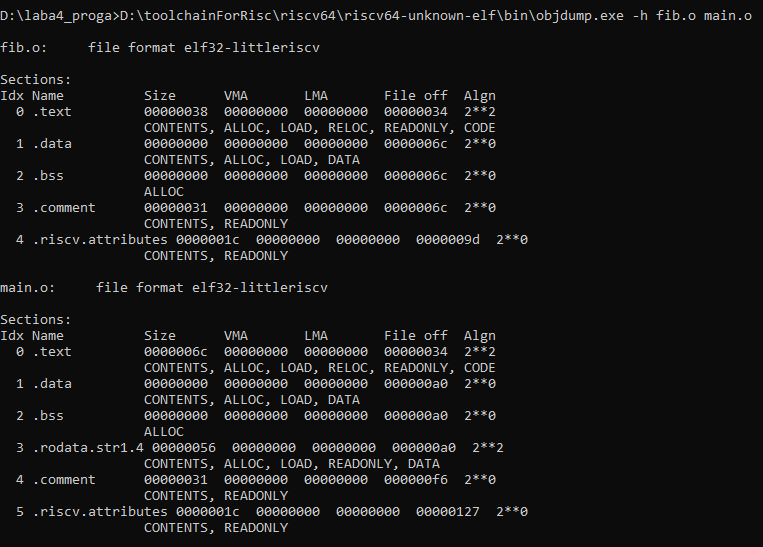
Драйвер компилятора riscv64-unknown-elf-gcc запускается с параметрами командной строки, где параметр “-c” указывает на необходимость остановить процесс сборки после ассемблирования.

Так как объектные файлы не являются текстовыми, для просмотра их содержимого используется objdump:

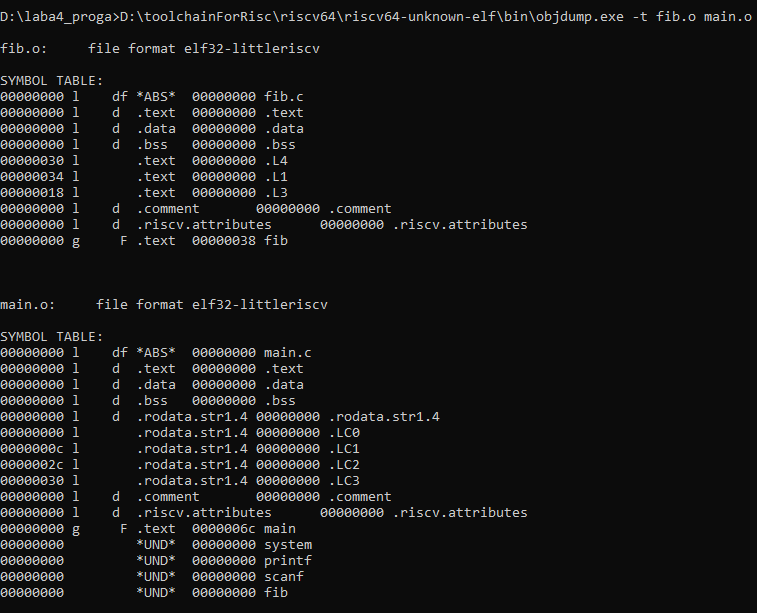


Оба файла содержат таблицу перемещений и таблицу символов (об этом нам говорят флаги HAS\_RELOC и HAS\_SYMS).

Выведем в консоль заголовки секций объектных файлов:

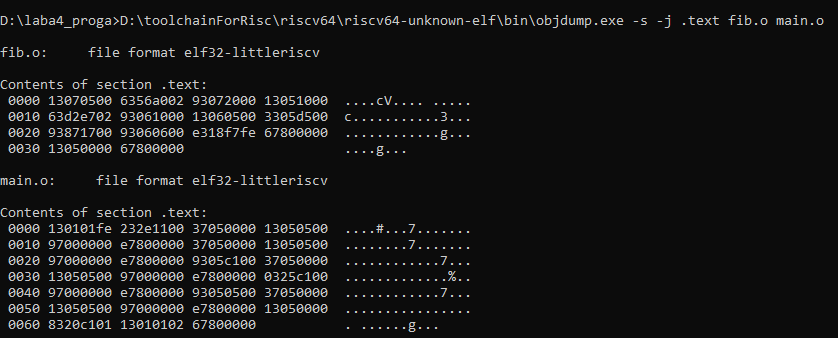


Далее просмотрим таблицы символов тех же файлов:

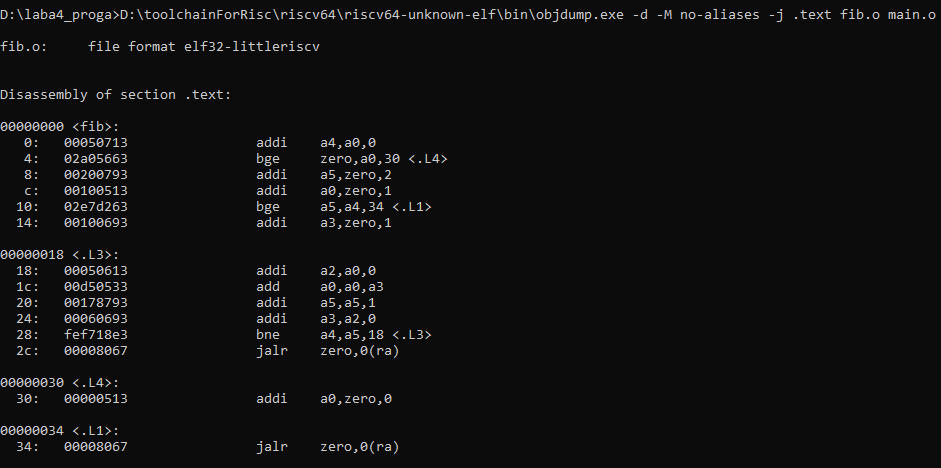


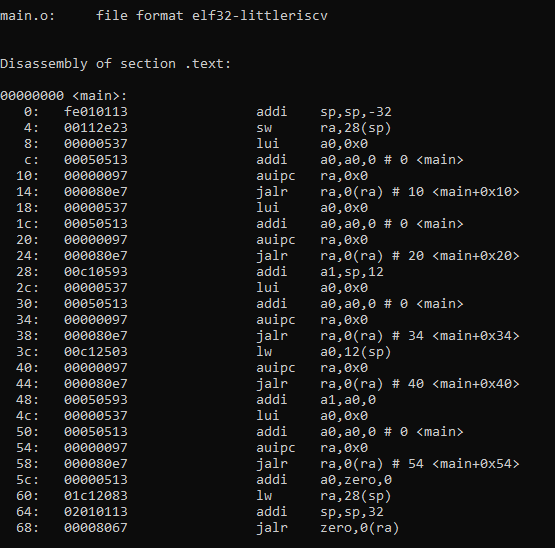
В таблице у fib.o только один глобальный (флаг “g”) символ типа «функция» (“F”) – для fib, так же как и в main.o для main.

Проанализируем секции .text объектных файлов, в которых содержатся коды инструкций:



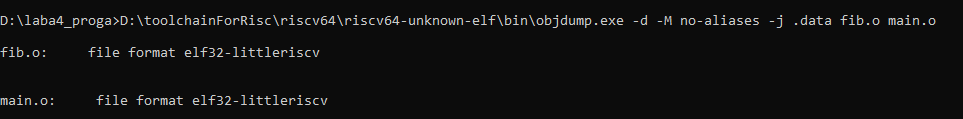
Процедура декодирования кодов инструкций является «механической», следовательно, разумно поручить ее выполнение ЭВМ, введя параметры “-d -M no-aliases”:





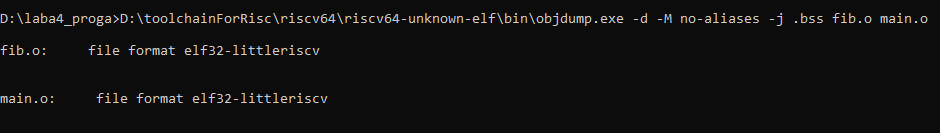
Дизассемблированный код очень похож на сгенерированный за исключением псевдо-инструкций.

Рассмотрим секции .data объектных файлов - секции инициализированных данных.

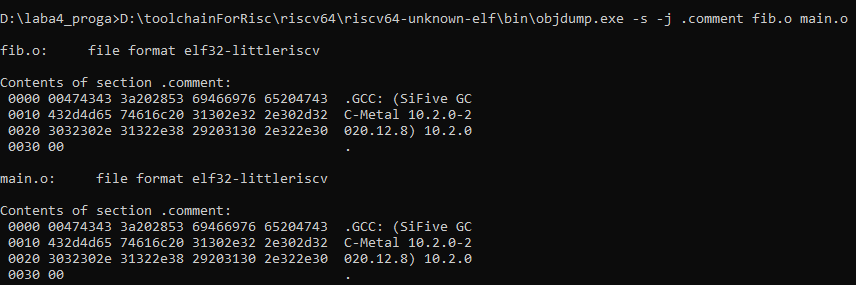


Размер секций равен нулю.

Секции .bss объектных файлов – секции данных, инициализированных нулями, тоже пусты:

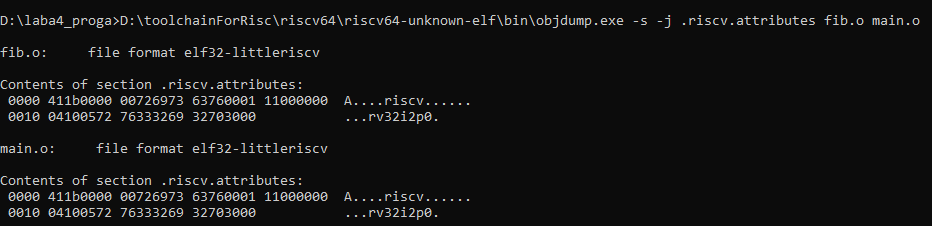


Рассмотрим секцию .comment – секцию данных о версиях:

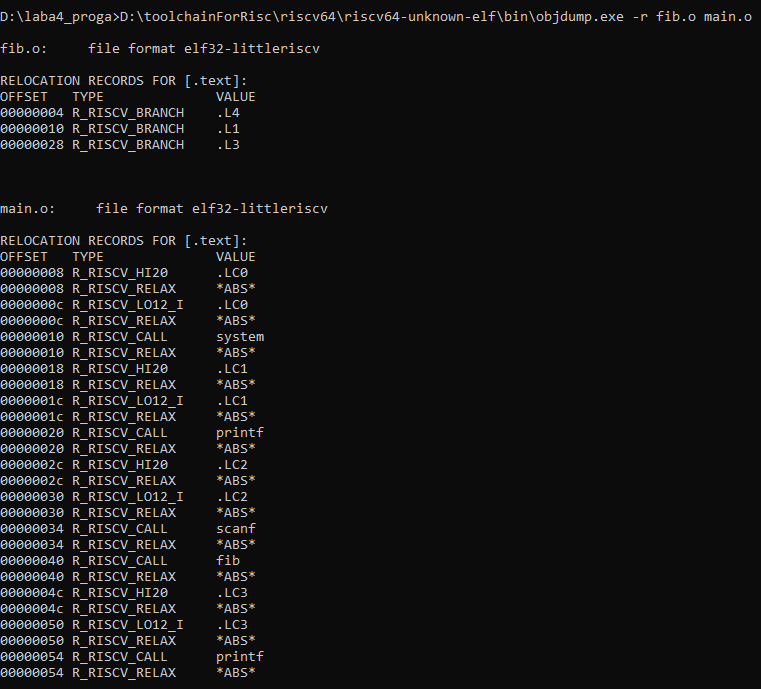


Как мы можем видеть, эта секция для обоих файлов содержит одни и те же значения: сведения о GCC.

Секция .riscv.attributes обоих объектных файлов содержит одну и ту же информацию об используемой архитектуре команд RV32I:



Рассмотрим таблицы перемещений объектных файлов:



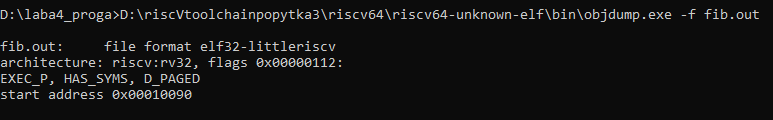
В таблицах перемещения fib.o содержится информация о ветвлениях(R\_RISCV\_BRANCH).

В таблицах перемещения main.o, есть R\_RISCV\_CALL. Записи типа R\_RISCV\_RELAX заносятся в таблицу перемещений в дополнение к записям типа R\_RISCV\_CAL.

Далее выполним компоновку и формирование исполняемых файлов программ:

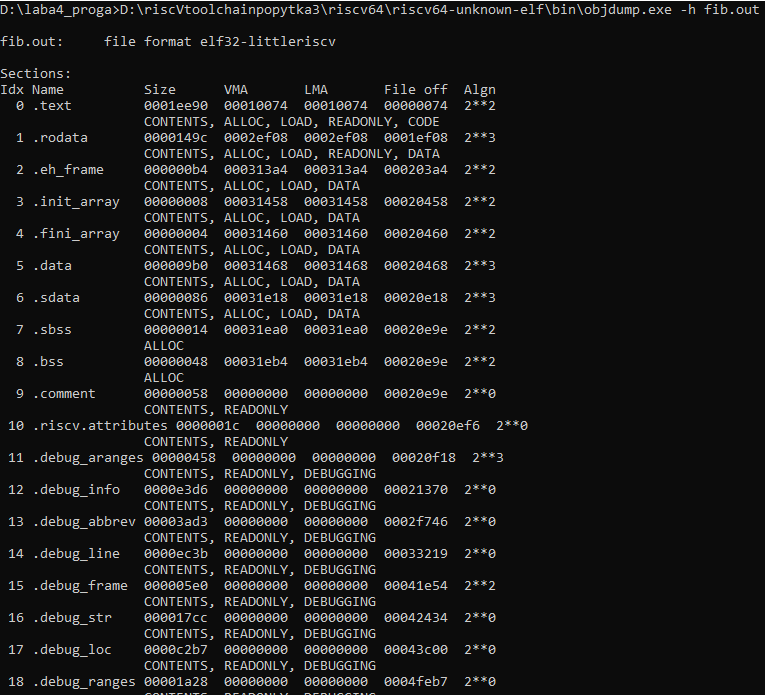


Сформированный компоновщиком файл “fib.out”, также является «бинарным», так что рассматриваем его с помощью “objdump”.



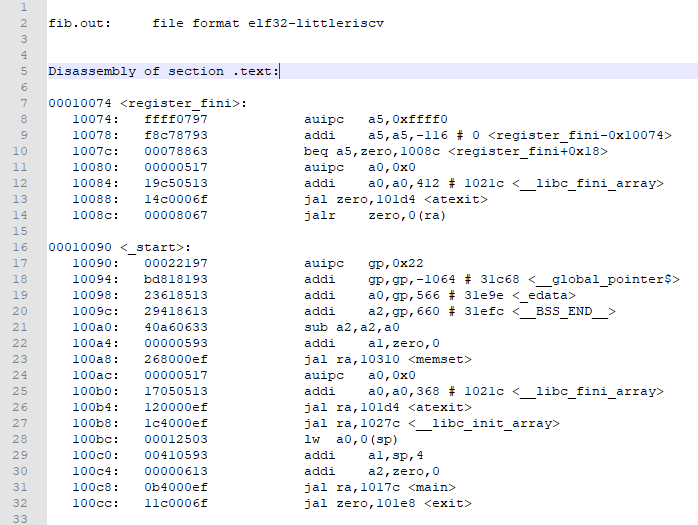
Флаг EXEC\_P указывает, что файл действительно является исполняемым, после загрузки его выполнение должно начаться с адреса 0x00010090 (entrypoint).

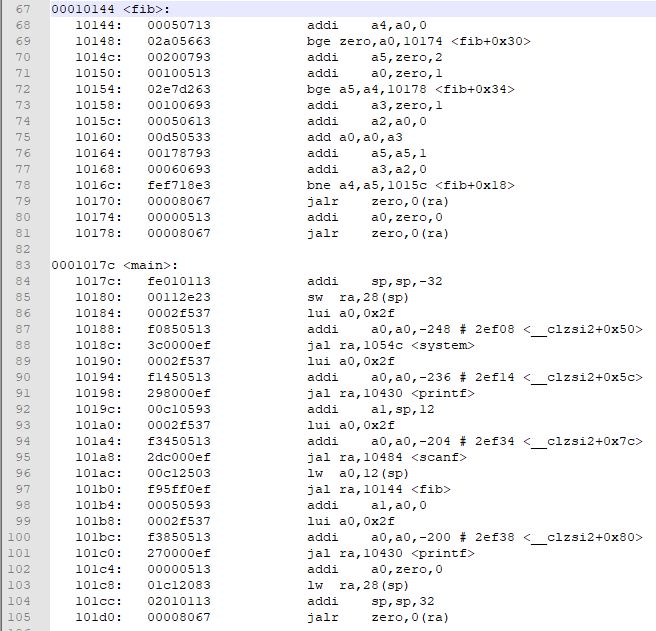
Откроем секции исполняемого файла:

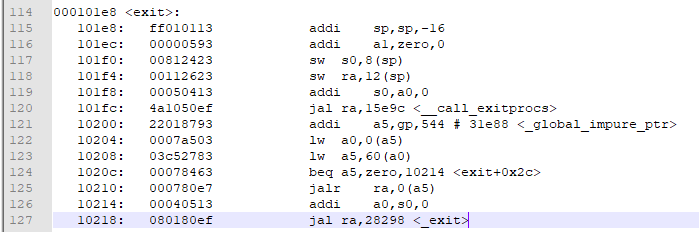


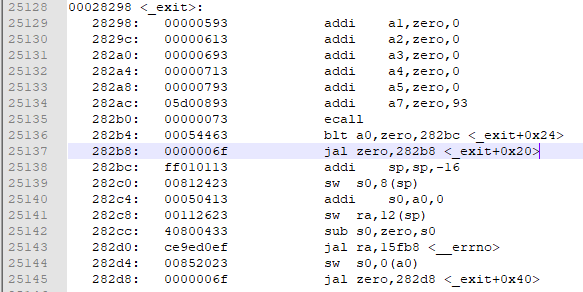
В данном файле производится слияние содержания секций обоих объектных файлов, а также расширение списка секций новыми блоками.

Выведем содержимое секции .text исполняемого файла в отдельный файл и проанализируем его содержимое (так как полученный файл имеет очень много информации, выведем самое главное):



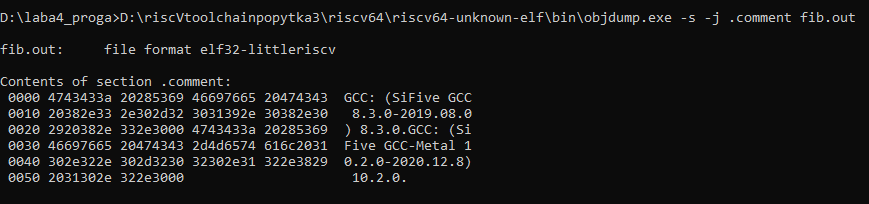




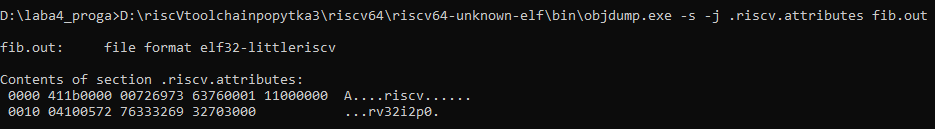


Компоновщик, заменил все переходы auipc+jalr на одну инструкцию jal и корректным адресом перехода.

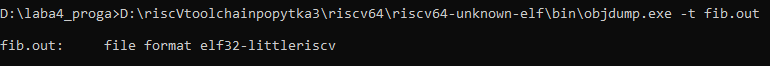
Секция .comment по-прежнему содержит информацию о версии GCC:



Секция .riscv.attributes так же содержит информацию об нашей архитектуре команд RV32I:



Откроем таблицу символов исполняемого файла:



Получим:

SYMBOL TABLE:

00010074 l d .text 00000000 .text

0002ef08 l d .rodata 00000000 .rodata

000313a4 l d .eh\_frame 00000000 .eh\_frame

00031458 l d .init\_array 00000000 .init\_array

00031460 l d .fini\_array 00000000 .fini\_array

00031468 l d .data 00000000 .data

00031e18 l d .sdata 00000000 .sdata

00031ea0 l d .sbss 00000000 .sbss

00031eb4 l d .bss 00000000 .bss

00000000 l d .comment 00000000 .comment

00000000 l d .riscv.attributes 00000000 .riscv.attributes

00000000 l d .debug\_aranges 00000000 .debug\_aranges

00000000 l d .debug\_info 00000000 .debug\_info

00000000 l d .debug\_abbrev 00000000 .debug\_abbrev

00000000 l d .debug\_line 00000000 .debug\_line

00000000 l d .debug\_frame 00000000 .debug\_frame

00000000 l d .debug\_str 00000000 .debug\_str

00000000 l d .debug\_loc 00000000 .debug\_loc

00000000 l d .debug\_ranges 00000000 .debug\_ranges

00000000 l df \*ABS\* 00000000 \_\_call\_atexit.c

00010074 l F .text 0000001c register\_fini

00000000 l df \*ABS\* 00000000 crtstuff.c

000313a4 l O .eh\_frame 00000000 \_\_EH\_FRAME\_BEGIN\_\_

000100d0 l F .text 00000000 \_\_do\_global\_dtors\_aux

00031eb4 l O .bss 00000001 completed.5434

00031460 l O .fini\_array 00000000 \_\_do\_global\_dtors\_aux\_fini\_array\_entry

00010120 l F .text 00000000 frame\_dummy

00031eb8 l O .bss 00000018 object.5439

0003145c l O .init\_array 00000000 \_\_frame\_dummy\_init\_array\_entry

00000000 l df \*ABS\* 00000000 fib.c

00000000 l df \*ABS\* 00000000 main.c

00000000 l df \*ABS\* 00000000 atexit.c

00000000 l df \*ABS\* 00000000 exit.c

00000000 l df \*ABS\* 00000000 fini.c

00000000 l df \*ABS\* 00000000 impure.c

00031468 l O .data 00000428 impure\_data

00000000 l df \*ABS\* 00000000 init.c

00000000 l df \*ABS\* 00000000 printf.c

00000000 l df \*ABS\* 00000000 scanf.c

00000000 l df \*ABS\* 00000000 system.c

00000000 l df \*ABS\* 00000000 vfprintf.c

00013080 l F .text 000000c0 \_\_sbprintf

0002f0cc l O .rodata 00000010 blanks.4504

0002f0dc l O .rodata 00000010 zeroes.4505

00000000 l df \*ABS\* 00000000 vfscanf.c

0002f4fc l O .rodata 00000022 basefix.4556

00000000 l df \*ABS\* 00000000 wsetup.c

00000000 l df \*ABS\* 00000000 \_\_atexit.c

00000000 l df \*ABS\* 00000000 ctype\_.c

00000000 l df \*ABS\* 00000000 errno.c

00000000 l df \*ABS\* 00000000 fflush.c

00000000 l df \*ABS\* 00000000 findfp.c

000162a8 l F .text 00000008 \_\_fp\_lock

000162bc l F .text 00000184 \_\_sinit.part.0

00016440 l F .text 00000008 \_\_fp\_unlock

00000000 l df \*ABS\* 00000000 fread.c

00000000 l df \*ABS\* 00000000 mallocr.c

00000000 l df \*ABS\* 00000000 fwalk.c

00000000 l df \*ABS\* 00000000 iswspace.c

00000000 l df \*ABS\* 00000000 ldtoa.c

00016e64 l F .text 00000068 eshdn1

00016ecc l F .text 00000070 eshup1

00016f3c l F .text 000000e0 m16m

0001701c l F .text 00000024 eisnan.part.0

00017040 l F .text 0000004c eneg

0001708c l F .text 00000040 eisneg

000170cc l F .text 000000e8 emovi

000171b4 l F .text 0000011c ecmp

000172d0 l F .text 0000001c eisinf.part.1

000172ec l F .text 000001ac eshift.part.3

00017498 l F .text 0000017c enormlz

00017614 l F .text 00000408 emdnorm

00017a1c l F .text 00000164 eiremain

00017b80 l F .text 000000b0 emovo.isra.6

00017c30 l F .text 00000408 emul

00018038 l F .text 00000504 ediv

0001853c l F .text 00000144 e113toe.isra.8

0002f62c l O .rodata 00000014 ezero

0002f640 l O .rodata 00000014 eone

0002f654 l O .rodata 00000104 etens

0002f758 l O .rodata 00000104 emtens

0002f85c l O .rodata 00000022 bmask

00000000 l df \*ABS\* 00000000 localeconv.c

00000000 l df \*ABS\* 00000000 locale.c

00000000 l df \*ABS\* 00000000 makebuf.c

00000000 l df \*ABS\* 00000000 malloc.c

00000000 l df \*ABS\* 00000000 mallocr.c

00000000 l df \*ABS\* 00000000 mbrtowc.c

00000000 l df \*ABS\* 00000000 mbtowc\_r.c

00000000 l df \*ABS\* 00000000 memchr.c

00000000 l df \*ABS\* 00000000 memcpy.c

00000000 l df \*ABS\* 00000000 mlock.c

00000000 l df \*ABS\* 00000000 mprec.c

0002f8b8 l O .rodata 0000000c p05.3298

00000000 l df \*ABS\* 00000000 realloc.c

00000000 l df \*ABS\* 00000000 mallocr.c

00000000 l df \*ABS\* 00000000 refill.c

0001be88 l F .text 0000001c lflush

00000000 l df \*ABS\* 00000000 sbrkr.c

00000000 l df \*ABS\* 00000000 sccl.c

00000000 l df \*ABS\* 00000000 sf\_nan.c

00000000 l df \*ABS\* 00000000 s\_frexp.c

00000000 l df \*ABS\* 00000000 sprintf.c

00000000 l df \*ABS\* 00000000 stdio.c

00000000 l df \*ABS\* 00000000 strcpy.c

00000000 l df \*ABS\* 00000000 strlen.c

00000000 l df \*ABS\* 00000000 strncpy.c

00000000 l df \*ABS\* 00000000 strtod.c

0001c844 l F .text 00000078 sulp

0002fab0 l O .rodata 00000014 fpi.3380

0002fac4 l O .rodata 00000014 fpinan.3416

0002fad8 l O .rodata 00000028 tinytens

00000000 l df \*ABS\* 00000000 strtold.c

00000000 l df \*ABS\* 00000000 strtoll.c

0001e0c0 l F .text 000002f0 \_strtoll\_l.isra.0

00000000 l df \*ABS\* 00000000 strtol.c

0001e3e4 l F .text 0000024c \_strtol\_l.isra.0

00000000 l df \*ABS\* 00000000 strtorx.c

00031e04 l O .data 00000014 fpi0.3163

00000000 l df \*ABS\* 00000000 strtoull.c

0001e7c0 l F .text 00000324 \_strtoull\_l.isra.0

00000000 l df \*ABS\* 00000000 strtoul.c

0001eb18 l F .text 00000284 \_strtoul\_l.isra.0

00000000 l df \*ABS\* 00000000 vfprintf.c

0002fc98 l O .rodata 00000010 blanks.4489

0002fca8 l O .rodata 00000010 zeroes.4490

00000000 l df \*ABS\* 00000000 ungetc.c

00000000 l df \*ABS\* 00000000 vfprintf.c

00021b1c l F .text 000000f0 \_\_sprint\_r.part.0

00022f0c l F .text 000000c0 \_\_sbprintf

0002fe24 l O .rodata 00000010 blanks.4480

0002fe34 l O .rodata 00000010 zeroes.4481

00000000 l df \*ABS\* 00000000 wctomb\_r.c

00000000 l df \*ABS\* 00000000 writer.c

00000000 l df \*ABS\* 00000000 mallocr.c

00000000 l df \*ABS\* 00000000 closer.c

00000000 l df \*ABS\* 00000000 fclose.c

00000000 l df \*ABS\* 00000000 fputwc.c

00000000 l df \*ABS\* 00000000 fstatr.c

00000000 l df \*ABS\* 00000000 fvwrite.c

00000000 l df \*ABS\* 00000000 gdtoa-gethex.c

00023998 l F .text 000000dc rshift

00000000 l df \*ABS\* 00000000 gdtoa-hexnan.c

00000000 l df \*ABS\* 00000000 isattyr.c

00000000 l df \*ABS\* 00000000 iswspace\_l.c

00000000 l df \*ABS\* 00000000 lnumeric.c

00031e9c l O .sdata 00000002 numempty

00000000 l df \*ABS\* 00000000 lseekr.c

00000000 l df \*ABS\* 00000000 memmove-stub.c

00000000 l df \*ABS\* 00000000 readr.c

00000000 l df \*ABS\* 00000000 reent.c

00000000 l df \*ABS\* 00000000 s\_nan.c

00000000 l df \*ABS\* 00000000 strncmp.c

00000000 l df \*ABS\* 00000000 strtodg.c

000249b4 l F .text 000000dc rshift

00024b80 l F .text 000003b0 rvOK

00030008 l O .rodata 0000005c fivesbits

00000000 l df \*ABS\* 00000000 vfprintf.c

000301d0 l O .rodata 00000010 blanks.4466

000301e0 l O .rodata 00000010 zeroes.4467

00000000 l df \*ABS\* 00000000 wbuf.c

00000000 l df \*ABS\* 00000000 wcrtomb.c

00000000 l df \*ABS\* 00000000 sys\_close.c

00000000 l df \*ABS\* 00000000 sys\_exit.c

00000000 l df \*ABS\* 00000000 sys\_fstat.c

00000000 l df \*ABS\* 00000000 sys\_isatty.c

00000000 l df \*ABS\* 00000000 sys\_lseek.c

00000000 l df \*ABS\* 00000000 sys\_read.c

00000000 l df \*ABS\* 00000000 sys\_sbrk.c

00031eb0 l O .sbss 00000004 heap\_end.1862

00000000 l df \*ABS\* 00000000 sys\_write.c

00000000 l df \*ABS\* 00000000 sys\_conv\_stat.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 adddf3.c

00000000 l df \*ABS\* 00000000 divdf3.c

00000000 l df \*ABS\* 00000000 eqdf2.c

00000000 l df \*ABS\* 00000000 gedf2.c

00000000 l df \*ABS\* 00000000 ledf2.c

00000000 l df \*ABS\* 00000000 muldf3.c

00000000 l df \*ABS\* 00000000 subdf3.c

00000000 l df \*ABS\* 00000000 unorddf2.c

00000000 l df \*ABS\* 00000000 fixdfsi.c

00000000 l df \*ABS\* 00000000 fixunsdfsi.c

00000000 l df \*ABS\* 00000000 floatsidf.c

00000000 l df \*ABS\* 00000000 floatunsidf.c

00000000 l df \*ABS\* 00000000 fixdfdi.c

00000000 l df \*ABS\* 00000000 floatdidf.c

00000000 l df \*ABS\* 00000000 eqtf2.c

00000000 l df \*ABS\* 00000000 getf2.c

00000000 l df \*ABS\* 00000000 letf2.c

00000000 l df \*ABS\* 00000000 multf3.c

00000000 l df \*ABS\* 00000000 subtf3.c

00000000 l df \*ABS\* 00000000 fixtfsi.c

00000000 l df \*ABS\* 00000000 floatsitf.c

00000000 l df \*ABS\* 00000000 lesf2.c

00000000 l df \*ABS\* 00000000 unordsf2.c

00000000 l df \*ABS\* 00000000 extenddftf2.c

00000000 l df \*ABS\* 00000000 truncdfsf2.c

00000000 l df \*ABS\* 00000000 trunctfdf2.c

00000000 l df \*ABS\* 00000000 multi3.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 libgcc2.c

00000000 l df \*ABS\* 00000000 crtstuff.c

00031454 l O .eh\_frame 00000000 \_\_FRAME\_END\_\_

00000000 l df \*ABS\* 00000000

00031464 l .fini\_array 00000000 \_\_fini\_array\_end

00031460 l .fini\_array 00000000 \_\_fini\_array\_start

00031460 l .init\_array 00000000 \_\_init\_array\_end

00031458 l .init\_array 00000000 \_\_preinit\_array\_end

00031458 l .init\_array 00000000 \_\_init\_array\_start

00031458 l .init\_array 00000000 \_\_preinit\_array\_start

0001b728 g F .text 0000009c \_mprec\_log10

0001b82c g F .text 00000078 \_\_any\_on

000244fc g F .text 00000054 \_isatty\_r

0002f990 g O .rodata 00000028 \_\_mprec\_tinytens

0001c63c g F .text 000000b0 strcpy

00024760 g F .text 00000040 cleanup\_glue

0002ee80 g F .text 00000038 .hidden \_\_ashldi3

00024580 g F .text 00000060 \_lseek\_r

0002b6c0 g F .text 00000144 .hidden \_\_getf2

0002a014 g F .text 0000008c .hidden \_\_eqdf2

0002b3e8 g F .text 0000020c .hidden \_\_floatdidf

0002b5f4 g F .text 000000cc .hidden \_\_eqtf2

00010430 g F .text 00000054 printf

00031c68 g \*ABS\* 00000000 \_\_global\_pointer$

00028138 g F .text 00000078 \_wcrtomb\_r

0001c450 g F .text 00000068 \_\_sseek

000165d8 g F .text 00000010 \_\_sinit

00027fa0 g F .text 00000184 \_\_swbuf\_r

000199f0 g F .text 0000007c \_setlocale\_r

00016448 g F .text 00000078 \_\_sfmoreglue

00021b08 g F .text 00000014 ungetc

0001a7f0 g F .text 00000004 \_\_malloc\_unlock

0001eae8 g F .text 00000018 strtoull\_l

0002e3a4 g F .text 00000188 .hidden \_\_floatsitf

0001dc9c g F .text 00000008 \_strtod\_r

000245e0 g F .text 00000120 memmove

000165c4 g F .text 00000014 \_cleanup

0001a7f4 g F .text 000000a8 \_Balloc

0002a0a0 g F .text 000000e4 .hidden \_\_gtdf2

00015fb8 g F .text 0000000c \_\_errno

0002851c g F .text 000000a4 \_conv\_stat

000199d8 g F .text 00000008 \_\_localeconv\_l

00023480 g F .text 0000005c \_fstat\_r

00031ef8 g O .bss 00000004 errno

00015c20 g F .text 00000018 \_\_svfscanf

0001e064 g F .text 0000005c strtold

0001c3c0 g F .text 00000008 \_\_seofread

00031e18 g .sdata 00000000 \_\_SDATA\_BEGIN\_\_

0001a6d0 g F .text 0000011c memcpy

00024290 g F .text 0000026c \_\_hexnan

000162b0 g F .text 0000000c \_cleanup\_r

0001edd0 g F .text 00002a98 \_svfprintf\_r

0002b1e4 g F .text 000000ac .hidden \_\_floatsidf

0002a184 g F .text 000000e4 .hidden \_\_ltdf2

0001e3b4 g F .text 00000018 strtoll\_l

0001a4e4 g F .text 0000009c mbrtowc

00015c38 g F .text 00000060 \_vfscanf\_r

0001b684 g F .text 000000a4 \_\_ratio

0001054c g F .text 00000030 system

00019c78 g F .text 00000010 malloc

000103ec g F .text 00000044 \_printf\_r

0002b804 g F .text 00000144 .hidden \_\_letf2

0002ed9c g F .text 00000048 .hidden \_\_udivsi3

000282dc g F .text 00000070 \_fstat

00016e5c g F .text 00000008 iswspace

0002f8c8 g O .rodata 000000c8 \_\_mprec\_tens

0001e3cc g F .text 00000018 strtoll

0002e52c g F .text 000000b4 .hidden \_\_lesf2

0002b168 g F .text 0000007c .hidden \_\_fixunsdfsi

00031eac g O .sbss 00000004 \_\_malloc\_top\_pad

0001edb8 g F .text 00000018 strtoul

00031e8c g O .sdata 00000000 .hidden \_\_dso\_handle

000199e0 g F .text 00000008 \_localeconv\_r

0001eae4 g F .text 00000004 \_strtoull\_r

000234dc g F .text 000004bc \_\_sfvwrite\_r

0001ac54 g F .text 00000034 \_\_i2b

0001e008 g F .text 0000005c strtold\_l

0001c0a8 g F .text 00000054 \_sbrk\_r

00024700 g F .text 00000060 \_read\_r

00023184 g F .text 00000110 \_fclose\_r

00016280 g F .text 00000028 fflush

00031ea8 g O .sbss 00000004 \_\_malloc\_max\_sbrked\_mem

0001051c g F .text 00000030 \_system\_r

00029084 g F .text 000007c0 .hidden \_\_adddf3

0001b428 g F .text 00000118 \_\_b2d

00028b94 g F .text 000004f0 .hidden \_\_umoddi3

0002834c g F .text 00000040 \_isatty

00031e88 g O .sdata 00000004 \_global\_impure\_ptr

0001b8b8 g F .text 000005d0 \_realloc\_r

0001027c g F .text 00000094 \_\_libc\_init\_array

0001e634 g F .text 00000018 strtol\_l

0002fe44 g O .rodata 00000100 \_\_hexdig

000285c0 g F .text 000005d4 .hidden \_\_udivdi3

000233d0 g F .text 0000002c \_fputwc\_r

0002f9b8 g O .rodata 00000028 \_\_mprec\_bigtens

0001aa18 g F .text 00000110 \_\_s2b

0002b290 g F .text 00000084 .hidden \_\_floatunsidf

0002842c g F .text 000000a0 \_sbrk

0001b188 g F .text 0000005c \_\_mcmp

00024a90 g F .text 000000f0 increment

000165f8 g F .text 00000014 \_\_fp\_lock\_all

0001e64c g F .text 00000018 strtol

0001021c g F .text 00000060 \_\_libc\_fini\_array

0001c8bc g F .text 000013e0 \_strtod\_l

0002424c g F .text 00000044 \_\_match

00024f6c g F .text 000000d8 set\_ones

000247a0 g F .text 00000110 \_reclaim\_reent

0001ab28 g F .text 00000074 \_\_hi0bits

0002b0e8 g F .text 00000080 .hidden \_\_fixdfsi

0001dcbc g F .text 00000018 strtod

0001e730 g F .text 00000090 \_strtorx\_l

0001dfa4 g F .text 00000064 \_strtold\_r

0002ee48 g F .text 00000038 .hidden \_\_lshrdi3

0002a184 g F .text 000000e4 .hidden \_\_ledf2

0002e254 g F .text 00000150 .hidden \_\_fixtfsi

0001dcd4 g F .text 00000168 strtof\_l

0001aef4 g F .text 00000148 \_\_pow5mult

0002ede4 g F .text 00000010 .hidden \_\_umodsi3

000104d8 g F .text 00000044 \_scanf\_r

0001e664 g F .text 000000cc ULtox

0002eeb8 g F .text 0000004c .hidden \_\_clzsi2

000165e8 g F .text 00000004 \_\_sfp\_lock\_acquire

0001a5f4 g F .text 000000dc memchr

0001c27c g F .text 0000006c \_sprintf\_r

000169f4 g F .text 000002f8 \_free\_r

00019a6c g F .text 00000008 \_\_locale\_mb\_cur\_max

00015e9c g F .text 0000011c \_\_call\_exitprocs

00031e94 g O .sdata 00000004 \_\_malloc\_sbrk\_base

00010090 g F .text 00000040 \_start

0001e3b0 g F .text 00000004 \_strtoll\_r

0002838c g F .text 00000050 \_lseek

0001b03c g F .text 0000014c \_\_lshift

0002b804 g F .text 00000144 .hidden \_\_lttf2

0002b5f4 g F .text 000000cc .hidden \_\_netf2

00026d04 g F .text 000001bc \_\_ssprint\_r

00015e00 g F .text 0000009c \_\_register\_exitproc

000248c4 g F .text 000000f0 strncmp

00019974 g F .text 00000064 \_ldcheck

00025044 g F .text 00001cc0 \_strtodg\_l

0001ac88 g F .text 0000026c \_\_multiply

0002ece0 g F .text 00000024 .hidden \_\_mulsi3

0001c778 g F .text 000000cc strncpy

00031ed0 g O .bss 00000028 \_\_malloc\_current\_mallinfo

0001b540 g F .text 00000144 \_\_d2b

0001b8a4 g F .text 00000014 realloc

0001ed9c g F .text 00000004 \_strtoul\_r

0002e624 g F .text 00000208 .hidden \_\_extenddftf2

00023130 g F .text 00000054 \_close\_r

00015c98 g F .text 00000168 \_\_swsetup\_r

00029844 g F .text 000007d0 .hidden \_\_divdf3

000164c0 g F .text 00000104 \_\_sfp

0001b7c4 g F .text 00000068 \_\_copybits

00031efc g .bss 00000000 \_\_BSS\_END\_\_

000319fc g O .data 00000408 \_\_malloc\_av\_

000165f4 g F .text 00000004 \_\_sinit\_lock\_release

0002a268 g F .text 00000670 .hidden \_\_muldf3

0001c364 g F .text 0000005c \_\_sread

000168b0 g F .text 0000001c fread

0001a7ec g F .text 00000004 \_\_malloc\_lock

00016220 g F .text 00000060 \_fflush\_r

00023068 g F .text 000000c8 \_calloc\_r

0001eb00 g F .text 00000018 strtoull

00031e9e g .sbss 00000000 \_\_bss\_start

00010310 g F .text 000000dc memset

0001017c g F .text 00000058 main

00031ea4 g O .sbss 00000004 \_\_malloc\_max\_total\_mem

00028124 g F .text 00000014 \_\_swbuf

0002e52c g F .text 000000b4 .hidden \_\_ltsf2

00016620 g F .text 00000290 \_fread\_r

0001c4b8 g F .text 00000008 \_\_sclose

00023294 g F .text 00000010 fclose

0001e630 g F .text 00000004 \_strtol\_r

0002e82c g F .text 00000180 .hidden \_\_truncdfsf2

00019c98 g F .text 000007cc \_malloc\_r

00022fd8 g F .text 00000030 \_\_ascii\_wctomb

00021868 g F .text 000000f0 \_\_submore

00016cec g F .text 000000b0 \_fwalk

0001a580 g F .text 0000000c \_mbtowc\_r

0002ed94 g F .text 00000084 .hidden \_\_divsi3

000168cc g F .text 00000128 \_malloc\_trim\_r

0001c4c0 g F .text 0000017c strcmp

00022ef4 g F .text 00000018 vfiprintf

0001eda0 g F .text 00000018 strtoul\_l

0002b948 g F .text 0000136c .hidden \_\_multf3

0002a014 g F .text 0000008c .hidden \_\_nedf2

0001c2e8 g F .text 0000007c sprintf

000302a4 g O .rodata 00000100 .hidden \_\_clz\_tab

00031ea0 g O .sbss 00000004 \_PathLocale

0002b314 g F .text 000000d4 .hidden \_\_fixdfdi

0001de3c g F .text 00000168 strtof

0001dca4 g F .text 00000018 strtod\_l

000101d4 g F .text 00000014 atexit

00023008 g F .text 00000060 \_write\_r

00024550 g F .text 00000028 iswspace\_l

00019a74 g F .text 00000014 setlocale

00010484 g F .text 00000054 scanf

0001c1bc g F .text 00000010 nanf

00031e90 g O .sdata 00000004 \_impure\_ptr

00015fc4 g F .text 0000025c \_\_sflush\_r

0001a464 g F .text 00000080 \_mbrtowc\_r

0002b6c0 g F .text 00000144 .hidden \_\_gttf2

00026ec0 g F .text 000010e0 \_svfiprintf\_r

0001bea4 g F .text 00000204 \_\_srefill\_r

0001a58c g F .text 00000068 \_\_ascii\_mbtowc

0002ccb4 g F .text 000015a0 .hidden \_\_subtf3

0001b3c8 g F .text 00000060 \_\_ulp

0001660c g F .text 00000014 \_\_fp\_unlock\_all

0002ff44 g O .rodata 0000000c \_C\_numeric\_locale

000248b0 g F .text 00000014 nan

000199e8 g F .text 00000008 localeconv

00019a88 g F .text 000000d0 \_\_swhatbuf\_r

00031468 g .data 00000000 \_\_DATA\_BEGIN\_\_

000284cc g F .text 00000050 \_write

00031e9e g .sdata 00000000 \_edata

00031efc g .bss 00000000 \_end

0002e5e0 g F .text 00000044 .hidden \_\_unordsf2

000232a4 g F .text 0000012c \_\_fputwc

00015bb8 g F .text 00000068 vfscanf

0001c3c8 g F .text 00000088 \_\_swrite

00031e98 g O .sdata 00000004 \_\_malloc\_trim\_threshold

000101e8 g F .text 00000034 exit

00021c24 g F .text 000012d0 \_vfiprintf\_r

00016d9c g F .text 000000c0 \_fwalk\_reent

0001b1e4 g F .text 000001e4 \_\_mdiff

0002ee18 g F .text 00000030 .hidden \_\_modsi3

000165ec g F .text 00000004 \_\_sfp\_lock\_release

00018680 g F .text 000012f4 \_ldtoa\_r

0001c0fc g F .text 000000c0 \_\_sccl

0002f528 g O .rodata 00000101 \_ctype\_

000283dc g F .text 00000050 \_read

0002b09c g F .text 0000004c .hidden \_\_unorddf2

00028298 g F .text 00000044 \_exit

0002ed04 g F .text 00000090 .hidden \_\_muldi3

00019b58 g F .text 00000120 \_\_smakebuf\_r

0001c6ec g F .text 0000008c strlen

0002a0a0 g F .text 000000e4 .hidden \_\_gedf2

00023a74 g F .text 000007d8 \_\_gethex

00021c0c g F .text 00000018 \_\_sprint\_r

00024578 g F .text 00000008 \_\_numeric\_load\_locale

00022fcc g F .text 0000000c \_wctomb\_r

0002a8d8 g F .text 000007c4 .hidden \_\_subdf3

0001057c g F .text 00002aec \_vfprintf\_r

0001ab9c g F .text 000000b8 \_\_lo0bits

00013140 g F .text 00002a78 \_\_svfscanf\_r

00021958 g F .text 000001b0 \_ungetc\_r

000281b0 g F .text 00000090 wcrtomb

00024f30 g F .text 0000003c decrement

0001c1cc g F .text 000000b0 frexp

00031890 g O .data 0000016c \_\_global\_locale

00013068 g F .text 00000018 vfprintf

0002e9ac g F .text 00000334 .hidden \_\_trunctfdf2

000233fc g F .text 00000084 fputwc

00028240 g F .text 00000058 \_close

00019c88 g F .text 00000010 free

000165f0 g F .text 00000004 \_\_sinit\_lock\_acquire

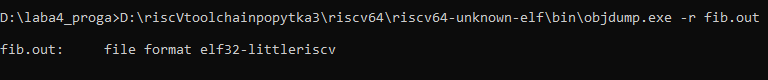
0001a8c0 g F .text 00000158 \_\_multadd

0001a89c g F .text 00000024 \_Bfree

00010144 g F .text 00000038 fib

Как мы можем заметить, таблица символов содержит множество дополнительных вхождений, однако определяет все нужные секции, метки и адреса. Функции *fib (00010144 g F .text 00000038 fib)* и *main (0001017c g F .text 00000058 main)* помечены флагом F, но в отличие от стадии ассемблирования все они являются определенными и содержатся по корректным адресам, чтобы их можно было успешно вызвать из других участков программы.

Откроем таблицу перемещений исполняемого файла:

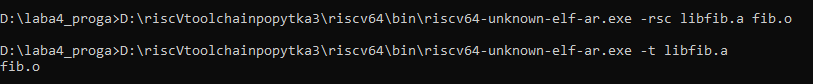


Как мы можем заметить, она пуста. Все необходимые релокации, оптимизации и замены инструкций были успешно проведены компоновщиком.

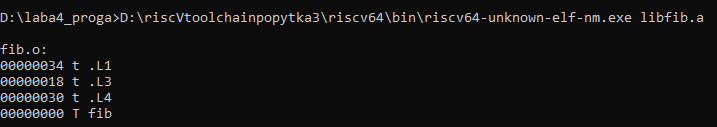
Итогом сборки по шагам является исполняемый на процессорах архитектуры RISC-V файл, выполняющий нахождение члена Фибоначчи по заданному номеру в последовательности.

**3.3. Формирование статической библиотеки, разработка make-файлов для сборки библиотеки.**

Статическая библиотека - архив объектных файлов, среди которых компоновщик выбирает «полезные» для данной программы: объектный файл можно считать «полезным», если в нем определяется еще не разрешенный компоновщиком символ. Разработанная функция определения числа Фибоначчи содержится в единственном исходном файле на языке C. Выделим этот файл в статическую библиотеку:



В выводе утилиты nm кодом T обозначаются символы, определенные в соответствующем объектном файле.



Как мы видим, символ функции fib является основным символом, определяемым в этом объектном файле, остальные символы определяют лишь локальные для этого файла метки.

Поиск библиотеки и сборка программы:



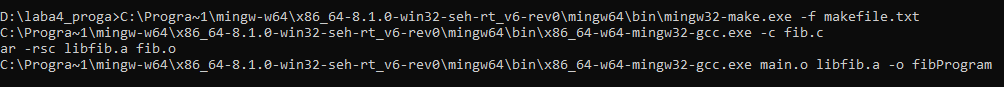
После выполнения получим файл fib.

Выполним сборку через make-файл:

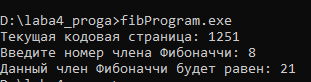
*Makefile* - это набор инструкций для программы *make*, которая позволяет собирать проекты, состоящие из большого числа “\*.c” и “\*.h” файлов. Обычно эта программа используется в связке с системами сборки, например *cmake*, позволяя вести проекты модульно (т.е. проект с включенными подпроектами).

Листинг написанного make-файла приведен ниже:

Запуск make-файла и процесс сборки:



Запуск созданного fibProgram.exe:



Программа работает корректно.

Что происходит в Make-файле:

1. Создается объектный файл main.o из исходного main.c
2. Создается объектный файл fib.o из исходного fib.c
3. Архивируется объектный файл fib.o (создаём статическую библиотеку libfib.a)
4. Компонуем статическую библиотеку libfib.a с объектным файлом main.o и получаем исполняемый файл fibProgram.exe

**4. Выводы**

В ходе выполнения лабораторной работы были закреплены знания языка C, ассемблера RISC-V, получены навыки работы с препроцессором, компилятором, ассемблером и компоновщиком пакета GCC и драйвером компилятора riscv64-unknown-elf-gcc. Были изучены особенности каждого этапа пошаговой сборки набора программ, а также инструменты, позволяющие выделить разработанные программы в статическую библиотеку и автоматизировать сборку этой библиотеки.