ФИО	Завьялов Никита Аркадьевич
Номер группы	M3138
Название работы	ISA

2. Ссылка на репозиторий: https://github.com/NEKAfk/comp-ach
3. Инструментарий: Работа была выполнена с использованием среды разработки VSCode, язык программирования C++, компилятор g++ (Debian 12.2.0-14) 12.2.0.

4. Результат работы программы на тестовых данных:

```
.text

00010074     <main>:
   10074:    ff010113        addi      sp, sp, -16
   10078:    00112623          sw      ra, 12(sp)
   1007c:    030000ef         jal      ra, 0x100ac <mmul>
   10080:    00c12083          lw      ra, 12(sp)
   10084:    00000513        addi      a0, zero, 0
   10088:    01010113        addi      sp, sp, 16
   1008c:    00008067        jalr      zero, 0(ra)
   10090:    00000013        addi      zero, zero, 0
   10094:    00100137         lui      sp, 0x100
   10098:    fddff0ef         jal      ra, 0x10074 <main>
   1009c:    00050593        addi      a1, a0, 0
   100a0:    00a00893        addi      a7, zero, 10
   100a4:    0ff0000f       fence      iorw, iorw
   100a8:    00000073       ecall

000100ac     <mmul>:
   100ac:    00011f37         lui      t5, 0x11
   100b0:    124f0513        addi      a0, t5, 292
   100b4:    65450513        addi      a0, a0, 1620
   100b8:    124f0f13        addi      t5, t5, 292
   100bc:    e4018293        addi      t0, gp, -448
   100c0:    fd018f93        addi      t6, gp, -48
   100c4:    02800e93        addi      t4, zero, 40

000100c8     <L2>:
   100c8:    fec50e13        addi      t3, a0, -20
   100cc:    000f0313        addi      t1, t5, 0
   100d0:    000f8893        addi      a7, t6, 0
   100d4:    00000813        addi      a6, zero, 0

000100d8     <L1>:
   100d8:    00088693        addi      a3, a7, 0
   100dc:    000e0793        addi      a5, t3, 0
   100e0:    00000613        addi      a2, zero, 0

000100e4     <L0>:
   100e4:    00078703          lb      a4, 0(a5)
   100e8:    00069583          lh      a1, 0(a3)
   100ec:    00178793        addi      a5, a5, 1
   100f0:    02868693        addi      a3, a3, 40
```

```
   100f4:    02b70733         mul      a4, a4, a1
   100f8:    00e60633         add      a2, a2, a4
   100fc:    fea794e3         bne      a5, a0, 0x100e4, <L0>
   10100:    00c32023          sw      a2, 0(t1)
   10104:    00280813        addi      a6, a6, 2
   10108:    00430313        addi      t1, t1, 4
   1010c:    00288893        addi      a7, a7, 2
   10110:    fdd814e3         bne      a6, t4, 0x100d8, <L1>
   10114:    050f0f13        addi      t5, t5, 80
   10118:    01478513        addi      a0, a5, 20
   1011c:    fa5f16e3         bne      t5, t0, 0x100c8, <L2>
   10120:    00008067        jalr      zero, 0(ra)
```

```
.symtab

Symbol Value              Size Type     Bind     Vis       Index Name
[   0] 0x0                   0 NOTYPE   LOCAL    DEFAULT   UNDEF
[   1] 0x10074               0 SECTION  LOCAL    DEFAULT       1
[   2] 0x11124               0 SECTION  LOCAL    DEFAULT       2
[   3] 0x0                   0 SECTION  LOCAL    DEFAULT       3
[   4] 0x0                   0 SECTION  LOCAL    DEFAULT       4
[   5] 0x0                   0 FILE     LOCAL    DEFAULT     ABS test.c
[   6] 0x11924               0 NOTYPE   GLOBAL   DEFAULT     ABS __global_pointer$
[   7] 0x118F4             800 OBJECT   GLOBAL   DEFAULT       2 b
[   8] 0x11124               0 NOTYPE   GLOBAL   DEFAULT       1 __SDATA_BEGIN__
[   9] 0x100AC             120 FUNC     GLOBAL   DEFAULT       1 mmul
[  10] 0x0                   0 NOTYPE   GLOBAL   DEFAULT   UNDEF _start
[  11] 0x11124            1600 OBJECT   GLOBAL   DEFAULT       2 c
[  12] 0x11C14               0 NOTYPE   GLOBAL   DEFAULT       2 __BSS_END__
[  13] 0x11124               0 NOTYPE   GLOBAL   DEFAULT       2 __bss_start
[  14] 0x10074              28 FUNC     GLOBAL   DEFAULT       1 main
[  15] 0x11124               0 NOTYPE   GLOBAL   DEFAULT       1 __DATA_BEGIN__
[  16] 0x11124               0 NOTYPE   GLOBAL   DEFAULT       1 _edata
[  17] 0x11C14               0 NOTYPE   GLOBAL   DEFAULT       2 _end
[  18] 0x11764             400 OBJECT   GLOBAL   DEFAULT       2 a
```

5. Были реализованы два набора комманд RV32I и RV32M. Для разбора файла, он был скопирован в массив. Для удобного считывания нескольких байтов был написан метод readNBytes(int n)

```
uint32_t getNBytes(int n) {
        uint32_t res = 0;
        uint32_t pow16 = 1;
        for (int i = 0; i < n; i++) {
                res += buffer[ind] * pow16;
                pow16 *= 256;
                ind++;
        }
        return res;
}
```

Единственная фиксированная часть elf файла – его header, располагающийся в самом начале файла и хранящий информацию о нем.

Сначала мы считываем все с приставкой e_sh* по рис. 1, элементы с данной приставкой хранят информацию об section table. e_shoff – оффсет от начала файла, показывающий расположение section header-ов, e_shentsize и e_shnum показывают сколько занимает(в байтах) одна секция и сколько их всего, e_shstrndx – индекс секциии с именами всех секций.

```
#define EI_NIDENT       16

typedef struct {
        unsigned char   e_ident[EI_NIDENT];
        Elf32_Half      e_type;
        Elf32_Half      e_machine;
        Elf32_Word      e_version;
        Elf32_Addr      e_entry;
        Elf32_Off       e_phoff;
        Elf32_Off       e_shoff;
        Elf32_Word      e_flags;
        Elf32_Half      e_ehsize;
        Elf32_Half      e_phentsize;
        Elf32_Half      e_phnum;
        Elf32_Half      e_shentsize;
        Elf32_Half      e_shnum;
        Elf32_Half      e_shstrndx;
} Elf32_Ehdr;
```

Puc.1. Oracle® Solaris 11.1 Linkers and Libraries, ElfHeader cmp. 300

Потом мы перемещаем указатель на e_shoff, ищем секцию с именем(sh_name) ".strtab" согласно рис. 2 и запоминаем оффсет таблицы строк, которая отвечает за имена элементов ".symtab".

Дальше мы находим секцию ".symtab", переходим на неё по sh_offset и парсим объекты как на рис. 3, сохраняя пары st_value – st_name. Также находим st_type = ((st_info) & 0xf) и st_bind = ((info) >> 4) st_vis = ((st_other) & 0x3)

Дальше парсим у каждого объекта значения как на рис. 4, рис. 5 и рис.6.

```
typedef struct {
        elf32_Word      sh_name;
        Elf32_Word      sh_type;
        Elf32_Word      sh_flags;
        Elf32_Addr      sh_addr;
        Elf32_Off       sh_offset;
        Elf32_Word      sh_size;
        Elf32_Word      sh_link;
        Elf32_Word      sh_info;
        Elf32_Word      sh_addralign;


Chapter 12 • Object File Format




        Elf32_Word      sh_entsize;
} Elf32_Shdr;
```

Puc.2. Oracle® Solaris 11.1 Linkers and Libraries, SectionTabEntries cmp. 309

```
typedef struct {
        Elf32_Word      st_name;
        Elf32_Addr      st_value;
        Elf32_Word      st_size;
        unsigned char   st_info;
        unsigned char   st_other;
        Elf32_Half      st_shndx;
} Elf32_Sym;
```

Puc.3. Oracle® Solaris 11.1 Linkers and Libraries, SymTabEntries cmp. 356

TABLE 12–19 ELF Symbol Binding, ELF32_ST_BIND and ELF64_ST_BIND

Name	Value
STB_LOCAL	0
STB_GLOBAL	1
STB_WEAK	2
STB_LOOS	10
STB_HIOS	12

TABLE 12–19 ELF Symbol Binding, ELF32_ST_BIND and ELF64_ST_BIND *(Continued)*

Name	Value
STB_LOPROC	13
STB_HIPROC	15

Puc.4. Oracle® Solaris 11.1 Linkers and Libraries, ST_BIND cmp. 357

TABLE 12–20 ELF Symbol Types, ELF32_ST_TYPE and ELF64_ST_TYPE

Name	Value
STT_NOTYPE	0
STT_OBJECT	1
STT_FUNC	2
STT_SECTION	3
STT_FILE	4
STT_COMMON	5
STT_TLS	6
STT_LOOS	10
STT_HIOS	12
STT_LOPROC	13
STT_SPARC_REGISTER	13
STT_HIPROC	15

Puc.5. Oracle® Solaris 11.1 Linkers and Libraries, ST_TYPE cmp. 359

TABLE 12–21 ELF Symbol Visibility

Name	Value
STV_DEFAULT	0
STV_INTERNAL	1
STV_HIDDEN	2
STV_PROTECTED	3
STV_EXPORTED	4
STV_SINGLETON	5
STV_ELIMINATE	6

Puc.6. Oracle® Solaris 11.1 Linkers and Libraries, ST_TYPE cmp. 360

31 27	26 25	24 20	19 15	14 12	11 7	6 0	
funct7		rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]			rs1	funct3	rd	opcode	I-type
imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode	B-type
imm[31:12]					rd	opcode	U-type
imm[20|10:1|11|19:12]					rd	opcode	J-type

RV32I Base Instruction Set						
imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU

imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm	pred succ	rs1	000	rd	0001111	FENCE
0000	0001 0000	00000	000	00000	0001111	PAUSE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK

RV32M Standard Extension						
0000001	rs2	rs1	000	rd	0110011	MUL
0000001	rs2	rs1	001	rd	0110011	MULH
0000001	rs2	rs1	010	rd	0110011	MULHSU
0000001	rs2	rs1	011	rd	0110011	MULHU
0000001	rs2	rs1	100	rd	0110011	DIV
0000001	rs2	rs1	101	rd	0110011	DIVU
0000001	rs2	rs1	110	rd	0110011	REM
0000001	rs2	rs1	111	rd	0110011	REMU

Register	ABI Name	Description	Saver
x0	zero	Hard-wired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5–7	t0–2	Temporaries	Caller
x8	s0/fp	Saved register/frame pointer	Callee
x9	s1	Saved register	Callee
x10–11	a0–1	Function arguments/return values	Caller
x12–17	a2–7	Function arguments	Caller
x18–27	s2–11	Saved registers	Callee
x28–31	t3–6	Temporaries	Caller

Рис.7 ABI naming conventions

Дальше переходим в секцию “.text”. Комманды расположены последовательно друг за другом и все занимают 4 байта. Расширение RV32I – стандартное расширение для работы с целыми числами, содержащее 40 инструкций. Расширение RV32M – это дополнение к RV32I нужное для умножения и деления целых чисел. Парсим записанные в секции комманды, предварительно найдя все метки и определив их имена (если значения метки нет в “.symtab”, то ей присваивается имя Ln, где n – первое не использованное неотрицательное число). Метки используют комманды jal и beq/bne/blt/bge/bltu/bgeu. Также имена регистров соответствуют рис. 7.

Приведем пример разбора инструкции: $fd018f93_{16}$ = $11111101000000011000111110010011_2$ opcode = 0010011, что соответствует коммандам addi, sltiu и др. Значит наша инструкция декодируется так, 11111101000000011000111110010011. Пурпурный участок равен 000_2, следовательно перед нами комманда addi. Регистрам $11111_2 = 31_{10}$, $00011_2 = 3_{10}$ соответствуют регистры t6 и gp, число выделенное голубым, это целой число, записанное в дополнении до двух. 111111010000_2 = 2000 -2^11 = -48_{10}. Команда: addi t6, gp, -48.

6. Список литературы.
1. Oracle® Solaris 11.1 Linkers and Libraries Guide
2. The RISC-V Instruction Set Manual Volume I - Unprivileged Architecture - Editors: Andrew waterman, Krste Asanovic, SiFive, Inc., CS Division, EECS Department,
University of California, Berkeley