

Отчёт по лабораторной работе №4

Дисциплина: Низкоуровневое программирование

Тема: Программирование RISC-V

Выполнил студент гр. 3530901/10005 _____ Кашин В.А.
(подпись)

Преподаватель _____ Корнев Д. А.
(подпись)

“ ” _____ 2022 г

Санкт-Петербург

2022

Текст программы

```
#include <stddef.h>

void sorted(int *array, size_t size){
    int min, temp;
    for(int i = 0; i < size; i++){
        min = i;
        for (int k = i + 1; k < size; k++){
            if(array[k] < array[min])
                min = k;
        }
        temp = array[i];
        array[i] = array[min];
        array[min] = temp;
    }
}
```

sorted.c(Основной файл)

```
#include <stddef.h>

void sorted (int *array, size_t size);
```

sorted.h(Заголовочный файл)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned const array[4] = {4, 3, 2, 1};
6      size_t arraySize = sizeof(array) / sizeof(array[0]);
7      sorted(array, arraySize);
8      for (int i = 0; i < arraySize; i++)
9          printf("%d ", array[i]);
10     return 0;
11 }
12
```

main.c(Тестовый файл)

Сборка программы

1.Препроцессирование

Препроцессирование файлов main.c и sorted.c, чтобы получить файлы main.i и sorted.i:

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -E 4\main.c -o 4\main.i
```

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -E 4\sorted.c -o 4\sorted.i
```

Параметры:

- march=rv32ic -mabi=ilp32 – целевым является процессор с базовой архитектурой системы команд RV32I;
- O1 – выполнять простые оптимизации генерируемого кода;
- E – остановить процесс сборки после препроцессирования;
- o – путь к выходному файлу.

```
# 2 "4\\sorted.h" 2

# 5 "4\\sorted.h"
extern void sorted (int *array, size_t size);
# 2 "4\\sorted.c" 2
# 1 "c:\\riscv64-unknown-elf\\lib\\gcc\\riscv64-unknown-elf\\10.2.0\\include\\stddef.h" 1 3 4
# 3 "4\\sorted.c" 2

void sorted(int *array, size_t size){
    int min, temp;
    for(int i =0; i < size; i++){
        min = i;
        for (int k = i + 1; k < size; k++)
            if(array[k] < array[min])
                min = k;
        temp = array[i];
        array[i] = array[min];
        array[min] = temp;
    }
}
```

(sorted.i)

```
# 2 "4\\main.c" 2
# 1 "4\\sorted.h" 1
# 1 "c:\\riscv64-unknown-elf\\lib\\gcc\\riscv64-unknown-elf\\10.2.0\\include\\stddef.h" 1 3 4
# 2 "4\\sorted.h" 2

# 5 "4\\sorted.h"
extern void sorted (int *array, size_t size);
# 3 "4\\main.c" 2

int main(void) {
    int array[7];
    for (int i = 0; i < 7; i++)
        scanf("%i", array[i]);
    sorted(array, 7);
    for (int i = 0; i < 7; i++)
        printf("%i ", &array[i]);
    return 0;
}
```

(main.i)

2. Компиляция

Компиляция файлов main.i и sorted.i сохранив результат (на языке ассемблера) в файлы main.s и sieve.s:

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -S -fpreprocessed 4\main.i -o 4\main.s
```

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -S -fpreprocessed 4\sorted.i -o 4\sorted.s
```

-S – остановить процесс сборки после компиляции, не запуская ассемблер;

-fpreprocessed – выполняется компиляция файла, уже отработанного процесса;

```
1  .file "main.c"
2  .option nopic
3  .attribute arch, "rv32i2p0_c2p0"
4  .attribute unaligned_access, 0
5  .attribute stack_align, 16
6  .text
7  .section .rodata.str1.4,"aMS",@progbits,1
8  .align 2
9  .LC0:
10 .string "%i"
11 .align 2
12 .LC1:
13 .string "%i "
14 .text
15 .align 1
16 .globl main
17 .type main, @function
18 main:
19 addi sp,sp,-64
20 sw ra,60(sp)
21 sw s0,56(sp)
22 sw s1,52(sp)
23 sw s2,48(sp)
24 sw s3,44(sp)
25 addi s0,sp,4
26 addi s2,sp,32
27 mv s1,s0
28 lui s3,%hi(.LC0)
29 .L2:
30 lw a1,0(s1)
31 addi a0,s3,%lo(.LC0)
32 call scanf
33 addi s1,s1,4
34 bne s1,s2,.L2
35 li a1,7
36 addi a0,sp,4
37 call sorted
38 lui s1,%hi(.LC1)
39 .L3:
40 mv a1,s0
41 addi a0,s1,%lo(.LC1)
42 call printf
43 addi s0,s0,4
44 bne s0,s2,.L3
45 li a0,0
46 lw ra,60(sp)
47 lw s0,56(sp)
48 lw s1,52(sp)
49 lw s2,48(sp)
50 lw s3,44(sp)
51 addi sp,sp,64
52 jr ra
53 .size main, .-main
54 .ident "GCC: (SiFive GCC-Metal 10.2.0-2020.12.8) 10.2.0"
55
```

(main.s)

```

1  .file "sorted.c"
2  .option nopic
3  .attribute arch, "rv32i2p0_c2p0"
4  .attribute unaligned_access, 0
5  .attribute stack_align, 16
6  .text
7  .align 1
8  .globl sorted
9  .type sorted, @function
10 sorted:
11     beq a1,zero,.L1
12     mv t1,a0
13     mv t3,a1
14     li a7,0
15     j .L6
16     .L4:
17     addi a4,a4,1
18     addi a3,a3,4
19     bleu a1,a4,.L3
20     .L5:
21     slli a5,a2,2
22     add a5,a0,a5
23     lw a6,4(a3)
24     lw a5,0(a5)
25     bge a6,a5,.L4
26     mv a2,a4
27     j .L4
28     .L3:
29     lw a5,0(t1)
30     slli a2,a2,2
31     add a2,a0,a2
32     lw a4,0(a2)
33     sw a4,0(t1)
34     sw a5,0(a2)
35     addi t1,t1,4
36     beq a7,t3,.L1
37     .L6:
38     mv a2,a7
39     addi a7,a7,1
40     bleu a1,a7,.L3
41     mv a3,t1
42     mv a4,a7
43     j .L5
44     .L1:
45     ret
46     .size sorted,.-sorted
47     .ident "GCC: (SiFive GCC-Metal 10.2.0-2020.12.8) 10.2.0"
48

```

(sorted.s)

3. Ассемблирование

Ассемблирование файлов main.s и sorted.s:

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -c 4\main.s -o 4\main.o
```

```
riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 -O1 -c 4\sorted.s -o 4\sorted.o
```

riscv64-unknown-elf-objdump -h 4\main.o:

Sections:						
Idx	Name	Size	VMA	LMA	File off	Algn
0	.text	00000060	00000000	00000000	00000034	2**1
		CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE				
1	.data	00000000	00000000	00000000	00000094	2**0
		CONTENTS, ALLOC, LOAD, DATA				
2	.bss	00000000	00000000	00000000	00000094	2**0
		ALLOC				
3	.rodata.str1.4	00000008	00000000	00000000	00000094	2**2
		CONTENTS, ALLOC, LOAD, READONLY, DATA				
4	.comment	00000031	00000000	00000000	0000009c	2**0
		CONTENTS, READONLY				
5	.riscv.attributes	00000021	00000000	00000000	000000cd	2**0
		CONTENTS, READONLY				

riscv64-unknown-elf-objdump -h 4\sorted.o

```
Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          0000004c  00000000  00000000  00000034  2**1
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data           00000000  00000000  00000000  00000080  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  00000080  2**0
    ALLOC
  3 .comment        00000031  00000000  00000000  00000080  2**0
    CONTENTS, READONLY
  4 .riscv.attributes 00000021  00000000  00000000  000000b1  2**0
    CONTENTS, READONLY
```

В файлах main.o и sorted.o появились такие секции как:

text – секция кода, в которой содержатся коды инструкций (название секции обусловлено историческими причинами);

.data – секция инициализированных данных;

.bss – секция неинициализированных статических переменных (название секции также обусловлено историческими причинами);

.rodata – аналог .data для неизменяемых данных

.comment – секция данных о версиях размером 12 байт

.riscv.attributes – информация про RISC-V

riscv64-unknown-elf-objdump -t 4\main.o:

```
4\main.o:      file format elf32-littleriscv

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 main.c
00000000 l     d  .text  00000000 .text
00000000 l     d  .data  00000000 .data
00000000 l     d  .bss   00000000 .bss
00000000 l     d  .rodata.str1.4 00000000 .rodata.str1.4
00000000 l     d  .rodata.str1.4 00000000 .LC0
00000004 l     d  .rodata.str1.4 00000000 .LC1
00000018 l     d  .text  00000000 .L2
0000003c l     d  .text  00000000 .L3
00000000 l     d  .comment      00000000 .comment
00000000 l     d  .riscv.attributes      00000000 .riscv.attributes
00000000 g     F  .text  00000060 main
00000000      *UND*  00000000 scanf
00000000      *UND*  00000000 sorted
00000000      *UND*  00000000 printf
```

riscv64-unknown-elf-objdump -t 4\sorted.o:

```
4\sorted.o:      file format elf32-littleriscv

SYMBOL TABLE:
00000000 l      df *ABS* 00000000 sorted.c
00000000 l      d  .text 00000000 .text
00000000 l      d  .data 00000000 .data
00000000 l      d  .bss 00000000 .bss
0000004a l      .text 00000000 .L1
0000003c l      .text 00000000 .L6
00000026 l      .text 00000000 .L3
0000000a l      .text 00000000 .L4
00000012 l      .text 00000000 .L5
00000000 l      d  .comment 00000000 .comment
00000000 l      d  .riscv.attributes 00000000 .riscv.attributes
00000000 g      F  .text 0000004c sorted
```

.text

riscv64-unknown-elf-objdump -s -j .text 4\sorted.o 4\main.o:

```
Contents of section .text:
 0000 a9c52a83 2e8e8148 15a80507 9106637c  ..*....H.....c|
 0010 b7009317 2600aa97 03a84600 9c43e356  ....&.....F..C.V
 0020 f8fe3a86 ddb78327 03000a06 2a961842  ..:.....'....*..B
 0030 2320e300 1cc21103 6389c801 46868508  # .....C...F...
 0040 e3f3b8fe 9a864687 e9b78280  ....F.....

4\main.o:      file format elf32-littleriscv

Contents of section .text:
 0000 397106de 22dc26da 4ad84ed6 40001309  9q..".&.J.N.@...
 0010 0102a284 b7090000 8c401385 09009700  .....@.....
 0020 0000e780 00009104 e39824ff 9d454800  .....$..EH.
 0030 97000000 e7800000 b7040000 a2851385  .....
 0040 04009700 0000e780 00001104 e31824ff  .....$.
 0050 0145f250 6254d254 4259b259 21618280  .E.PbT.TBY.Y!a..
```

riscv64-unknown-elf-objdump -d -M no-aliases -j .text 4\sorted.o 4\main.o:

```
4\sorted.o:      file format elf32-littleriscv

Disassembly of section .text:

00000000 <sorted>:
   0:   c5a9          c.beqz    a1,4a <.L1>
   2:   832a          c.mv     t1,a0
   4:   8e2e          c.mv     t3,a1
   6:   4881          c.li     a7,0
   8:   a815          c.j      3c <.L6>

0000000a <.L4>:
   a:   0705          c.addi    a4,1
   c:   0691          c.addi    a3,4
   e:   00b77c63      bgeu     a4,a1,26 <.L3>

00000012 <.L5>:
  12:   00261793      slli     a5,a2,0x2
  16:   97aa          c.add     a5,a0
  18:   0046a803      lw       a6,4(a3)
  1c:   439c          c.lw     a5,0(a5)
  1e:   fef856e3      bge      a6,a5,a <.L4>
  22:   863a          c.mv     a2,a4
  24:   b7dd          c.j      a <.L4>

00000026 <.L3>:
  26:   00032783      lw       a5,0(t1)
  2a:   060a          c.slli    a2,0x2
  2c:   962a          c.add     a2,a0
  2e:   4218          c.lw     a4,0(a2)
  30:   00e32023      sw       a4,0(t1)
  34:   c21c          c.sw     a5,0(a2)
  36:   0311          c.addi    t1,4
  38:   01c88963      beq      a7,t3,4a <.L1>

0000003c <.L6>:
  3c:   8646          c.mv     a2,a7
  3e:   0885          c.addi    a7,1
  40:   feb8f3e3      bgeu     a7,a1,26 <.L3>
  44:   869a          c.mv     a3,t1
  46:   8746          c.mv     a4,a7
  48:   b7e9          c.j      12 <.L5>

0000004a <.L1>:
  4a:   8082          c.jr     ra
```

(sorted.o)

4\main.o: file format elf32-littleriscv

Disassembly of section .text:

00000000 <main>:

0:	7139	c.addi16sp	sp,-64
2:	de06	c.swsp	ra,60(sp)
4:	dc22	c.swsp	s0,56(sp)
6:	da26	c.swsp	s1,52(sp)
8:	d84a	c.swsp	s2,48(sp)
a:	d64e	c.swsp	s3,44(sp)
c:	0040	c.addi4spn	s0,sp,4
e:	02010913	addi	s2,sp,32
12:	84a2	c.mv	s1,s0
14:	000009b7	lui	s3,0x0

00000018 <.L2>:

18:	408c	c.lw	a1,0(s1)
1a:	00098513	addi	a0,s3,0 # 0 <main>
1e:	00000097	auipc	ra,0x0
22:	000080e7	jalr	ra,0(ra) # 1e <.L2+0x6>
26:	0491	c.addi	s1,4
28:	ff2498e3	bne	s1,s2,18 <.L2>
2c:	459d	c.li	a1,7
2e:	0048	c.addi4spn	a0,sp,4
30:	00000097	auipc	ra,0x0
34:	000080e7	jalr	ra,0(ra) # 30 <.L2+0x18>
38:	000004b7	lui	s1,0x0

0000003c <.L3>:

3c:	85a2	c.mv	a1,s0
3e:	00048513	addi	a0,s1,0 # 0 <main>
42:	00000097	auipc	ra,0x0
46:	000080e7	jalr	ra,0(ra) # 42 <.L3+0x6>
4a:	0411	c.addi	s0,4
4c:	ff2418e3	bne	s0,s2,3c <.L3>
50:	4501	c.li	a0,0
52:	50f2	c.lwsp	ra,60(sp)
54:	5462	c.lwsp	s0,56(sp)
56:	54d2	c.lwsp	s1,52(sp)
58:	5942	c.lwsp	s2,48(sp)
5a:	59b2	c.lwsp	s3,44(sp)
5c:	6121	c.addi16sp	sp,64
5e:	8082	c.jr	ra

(main.o)

4. Компоновка

riscv64-unknown-elf-gcc -march=rv32ic -mabi=ilp32 4\main.o 4\sorted.o -o 4\res

riscv64-unknown-elf-objdump -j .text -d -M no-aliases 4\res >4\res.ds

```

00010110 <main>:
10110: 7139          c.addi16sp sp,-64
10112: de06          c.swsp ra,60(sp)
10114: dc22          c.swsp s0,56(sp)
10116: da26          c.swsp s1,52(sp)
10118: d84a          c.swsp s2,48(sp)
1011a: d64e          c.swsp s3,44(sp)
1011c: 0040          c.addi4spn s0,sp,4
1011e: 02010913      addi s2,sp,32
10122: 84a2          c.mv s1,s0
10124: 000269b7      lui s3,0x26
10128: 408c          c.lw a1,0(s1)
1012a: 6c098513      addi a0,s3,1728 # 266c0 <__clzsi2+0x3e>
1012e: 2add          c.jal 10324 <scanf>
10130: 0491          c.addi s1,4
10132: ff249be3      bne s1,s2,10128 <main+0x18>
10136: 459d          c.li a1,7
10138: 0048          c.addi4spn a0,sp,4
1013a: 2015          c.jal 1015e <sorted>
1013c: 000264b7      lui s1,0x26
10140: 85a2          c.mv a1,s0
10142: 6c448513      addi a0,s1,1732 # 266c4 <__clzsi2+0x42>
10146: 2a55          c.jal 102fa <printf>
10148: 0411          c.addi s0,4
1014a: ff241be3      bne s0,s2,10140 <main+0x30>
1014e: 4501          c.li a0,0
10150: 50f2          c.lwsp ra,60(sp)
10152: 5462          c.lwsp s0,56(sp)
10154: 54d2          c.lwsp s1,52(sp)
10156: 5942          c.lwsp s2,48(sp)
10158: 59b2          c.lwsp s3,44(sp)
1015a: 6121          c.addi16sp sp,64
1015c: 8082          c.jr ra

0001015e <sorted>:
1015e: c5a9          c.beqz a1,101a8 <sorted+0x4a>
10160: 832a          c.mv t1,a0
10162: 8e2e          c.mv t3,a1
10164: 4081          c.li a7,0
10166: a815          c.j 1019a <sorted+0x3c>
10168: 0705          c.addi a4,1
1016a: 0691          c.addi a3,4
1016c: 00b77c63      bgeu a4,a1,10184 <sorted+0x26>
10170: 00261793      slli a5,a2,0x2
10174: 97aa          c.add a5,a0
10176: 0046a803      lw a6,4(a3)
1017a: 439c          c.lw a5,0(a5)
1017c: fef856e3      bge a6,a5,10168 <sorted+0xa>
10180: 863a          c.mv a2,a4
10182: b7dd          c.j 10168 <sorted+0xa>
10184: 00032783      lw a5,0(t1) # 10106 <frame_dummy+0x12>
10188: 060a          c.slli a2,0x2
1018a: 962a          c.add a2,a0
1018c: 4218          c.lw a4,0(a2)
1018e: 00e32023      sw a4,0(t1)
10192: c21c          c.sw a5,0(a2)
10194: 0311          c.addi t1,4
10196: 01c88963      beq a7,t3,101a8 <sorted+0x4a>
1019a: 8646          c.mv a2,a7
1019c: 0885          c.addi a7,1
1019e: feb8f3e3      bgeu a7,a1,10184 <sorted+0x26>
101a2: 869a          c.mv a3,t1
101a4: 8746          c.mv a4,a7
101a6: b7e9          c.j 10170 <sorted+0x12>
101a8: 8082          c.jr ra

```

Создание статистической библиотек

riscv64-unknown-elf-ar -rsc 4\lib.a 4\sorted.o

Используем makefile:

```
output: main.o lib.a
    gcc main.o lib.a -o output
main.o: main.c
    gcc -c main.c

lib.a: sorted.o sorted.h
    ar -rsc lib.a sorted.o

sorted.o:
    gcc -c sorted.c

clean:
    del .o.a
```