PR003说明

提纲

- 作业时间表
  - PR003，今天~待定
- 实验任务
  - 中间代码生成
  - 汇编代码生成
- 模拟器

实验任务

- 任务：
  - 从AST进行生成中间代码
  - 由中间代码翻译为RISC-V汇编
- 难点：
  - 中间代码的设计
  - AST到中间代码的翻译
  - 中间代码到RISC-V汇编的翻译

实验任务——中间代码生成

- 中间代码的设计
  - 三地址码
  - SSA（不要求）
- 到中间代码的翻译
  - if-else如何翻译？
  - while如何翻译？
  - 变量声明、定义如何翻译？

实验任务——中间代码设计

- 定义函数
- 赋值操作
- 加、减、乘、除
- 取地址操作
  - 取数组元素

实验任务——中间代码设计

- 取某地址的内存单元的内容
- 向某地址的内存单元存内容
- 定义标号
  - if-else、while跳转
- 无条件跳转到标号
- 有条件跳转到标号
- 返回

实验任务——中间代码设计

- 传形参
- 调用函数
- 传实参

实验任务——中间代码生成

- 翻译模式
  - 基本表达式
  - 语句
  - 条件表达式
  - 函数调用
  - 函数参数
  - 数组
- 龙书上有

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC
  - 大概长什么样子
- 有哪些指令
- ABI
  - 调用约定
  - 寄存器使用
- 翻译的细节
  - float怎么表示？全局变量怎么表示?

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
1    int func(int a)
2    {
3        int b = 3;
4        a = a + b;
5        return a;
6    }
7
8    int main(void)
9    {
10       int a = 0;
11       int b = a;
12       func(a);
13       return 0;
14   }
```

```
1        .file   "testcode.c"
2        .option nopic
3        .attribute arch, "rv64i2p0_m2p0_a2p0_f2p0_d2p0_c2p0"
4        .attribute unaligned_access, 0
5        .attribute stack_align, 16
6        .text
7        .align  1
8        .globl  func
9        .type   func, @function
10 func:
11       addi    sp,sp,-48
12       sd  s0,40(sp)
13       addi    s0,sp,48
14       mv  a5,a0
15       sw  a5,-36(s0)
16       li  a5,3
17       sw  a5,-20(s0)
18       lw  a4,-36(s0)
19       lw  a5,-20(s0)
20       addw    a5,a4,a5
21       sw  a5,-36(s0)
22       lw  a5,-36(s0)
23       mv  a0,a5
24       ld  s0,40(sp)
25       addi    sp,sp,48
26       jr  ra
27       .size   func, .-func
28       .align  1
29       .globl  main
30       .type   main, @function
```

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
1  int func(int a)
2  {
3      int b = 3;
4      a = a + b;
5      return a;
6  }
7
8  int main(void)
9  {
10     int a = 0;
11     int b = a;
12     func(a);
13     return 0;
14 }
```

```
28      .align  1
29      .globl  main
30      .type   main, @function
31 main:
32      addi    sp,sp,-32
33      sd  ra,24(sp)
34      sd  s0,16(sp)
35      addi    s0,sp,32
36      sw  zero,-20(s0)
37      lw  a5,-20(s0)
38      sw  a5,-24(s0)
39      lw  a5,-20(s0)
40      mv  a0,a5
41      call    func
42      li  a5,0
43      mv  a0,a5
44      ld  ra,24(sp)
45      ld  s0,16(sp)
46      addi    sp,sp,32
47      jr  ra
48      .size   main, .-main
49      .ident  "GCC: (GNU) 10.2.0"
```

实验任务——汇编代码生成

- 目标机器： RISC-V 64GC

```
18  float func(float a, float b)
19  {
20      float c = a;
21      c = a + b;
22      return c;
23  }
24
25  int main(void)
26  {
27      float a = 3.2;
28      float b = 4.3;
29      int c = 4;
30      float d;
31      d = func(a, b);
32      return 0;
33  }
```

```
 5          .attribute stack_align, 16
 6          .text
 7          .align   1
 8          .globl   func
 9          .type    func, @function
10  func:
11          addi     sp,sp,-48
12          sd  s0,40(sp)
13          addi     s0,sp,48
14          fsw fa0,-36(s0)
15          fsw fa1,-40(s0)
16          flw fa5,-36(s0)
17          fsw fa5,-20(s0)
18          flw fa4,-36(s0)
19          flw fa5,-40(s0)
20          fadd.s   fa5,fa4,fa5
21          fsw fa5,-20(s0)
22          flw fa5,-20(s0)
23          fmv.s    fa0,fa5
24          ld  s0,40(sp)
25          addi     sp,sp,48
26          jr  ra
27          .size    func,  .-func
```

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
18  float func(float a, float b)
19  {
20      float c = a;
21      c = a + b;
22      return c;
23  }
24
25  int main(void)
26  {
27      float a = 3.2;
28      float b = 4.3;
29      int c = 4;
30      float d;
31      d = func(a, b);
32      return 0;
33  }
```

```
28      .align   1
29      .globl   main
30      .type    main, @function
31  main:
32      addi     sp,sp,-32
33      sd  ra,24(sp)
34      sd  s0,16(sp)
35      addi     s0,sp,32
36      lui a5,%hi(.LC0)
37      flw fa5,%lo(.LC0)(a5)
38      fsw fa5,-20(s0)
39      lui a5,%hi(.LC1)
40      flw fa5,%lo(.LC1)(a5)
41      fsw fa5,-24(s0)
42      li  a5,4
43      sw  a5,-28(s0)
44      flw fa1,-24(s0)
45      flw fa0,-20(s0)
46      call     func
47      fsw fa0,-32(s0)
48      li  a5,0
49      mv  a0,a5
50      ld  ra,24(sp)
51      ld  s0,16(sp)
52      addi     sp,sp,32
53      jr  ra
54      .size    main, .-main
55      .section    .rodata
56      .align   2
57  .LC0:
58      .word    1078774989
59      .align   2
60  .LC1:
61      .word    1082759578
62      .ident   "GCC: (GNU) 10.2.0"
```

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
39  int aa = 0;
40  const int bb = 2;
41  int a_array[5];
42  int b_array[5] = {1, 2, 3, 4, 5};
43  int c_array[5] = {1, 2, 3};
44  float fa = 1.2;
45  double da = 2.3;
46
47  float func(float a, float b)
48  {
49      float c = a;
50      c = a + b;
51      return c;
52  }
53
54  int main(void)
55  {
56      float a = 3.2;
57      float b = 4.3;
58      int c = 4;
59      float d;
60      d = func(a, b);
61      return 0;
62  }
```

```
 5      .attribute stack_align, 16
 6      .text
 7      .globl  aa
 8      .section    .sbss,"aw",@nobits
 9      .align  2
10      .type   aa, @object
11      .size   aa, 4
12  aa:
13      .zero   4
14      .globl  bb
15      .section    .srodata,"a"
16      .align  2
17      .type   bb, @object
18      .size   bb, 4
19  bb:
20      .word   2
21      .globl  a_array
22      .bss
23      .align  3
24      .type   a_array, @object
25      .size   a_array, 20
```

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
39    int aa = 0;
40    const int bb = 2;
41    int a_array[5];
42    int b_array[5] = {1, 2, 3, 4, 5};
43    int c_array[5] = {1, 2, 3};
44    float fa = 1.2;
45    double da = 2.3;
46
47    float func(float a, float b)
48    {
49        float c = a;
50        c = a + b;
51        return c;
52    }
53
54    int main(void)
55    {
56        float a = 3.2;
57        float b = 4.3;
58        int c = 4;
59        float d;
60        d = func(a, b);
61        return 0;
62    }
```

```
21        .globl  a_array
22        .bss
23        .align  3
24        .type   a_array, @object
25        .size   a_array, 20
26 a_array:
27        .zero   20
28        .globl  b_array
29        .data
30        .align  3
31        .type   b_array, @object
32        .size   b_array, 20
33 b_array:
34        .word   1
35        .word   2
36        .word   3
37        .word   4
38        .word   5
39        .globl  c_array
40        .align  3
41        .type   c_array, @object
42        .size   c_array, 20
43 c_array:
44        .word   1
45        .word   2
46        .word   3
47        .zero   8
```

实验任务——汇编代码生成

- 目标机器：RISC-V 64GC

```
39    int aa = 0;
40    const int bb = 2;
41    int a_array[5];
42    int b_array[5] = {1, 2, 3, 4, 5};
43    int c_array[5] = {1, 2, 3};
44    float fa = 1.2;
45    double da = 2.3;
46
47    float func(float a, float b)
48    {
49        float c = a;
50        c = a + b;
51        return c;
52    }
53
54    int main(void)
55    {
56        float a = 3.2;
57        float b = 4.3;
58        int c = 4;
59        float d;
60        d = func(a, b);
61        return 0;
62    }
```

```
48        .globl  fa
49        .section    .sdata,"aw"
50        .align  2
51        .type   fa, @object
52        .size   fa, 4
53 fa:
54        .word   1067030938
55        .globl  da
56        .align  3
57        .type   da, @object
58        .size   da, 8
59 da:
60        .word   1717986918
61        .word   1073899110
```

实验任务——汇编代码生成

- RISC-V 64GC指令

- Chapter 24

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

RV64I Base Instruction Set (in addition to RV32I)

imm[11:0]		rs1	110	rd	0000011	LWU
imm[11:0]		rs1	011	rd	0000011	LD
imm[11:5]	rs2	rs1	011	imm[4:0]	0100011	SD
000000	shamt	rs1	001	rd	0010011	SLLI
000000	shamt	rs1	101	rd	0010011	SRLI
010000	shamt	rs1	101	rd	0010011	SRAI
imm[11:0]		rs1	000	rd	0011011	ADDIW

实验任务——汇编代码生成

- RISC-V 64GC ABI
  - 寄存器约定
  - 调用约定
- Chapter 25

Integer Register Convention

Name	ABI Mnemonic	Meaning	Preserved across calls?
x0	zero	Zero	-- (Immutable)
x1	ra	Return address	No
x2	sp	Stack pointer	Yes
x3	gp	Global pointer	-- (Unallocatable)
x4	tp	Thread pointer	-- (Unallocatable)
x5-x7	t0-t2	Temporary registers	No
x8-x9	s0-s1	Callee-saved registers	Yes
x10-x17	a0-a7	Argument registers	No
x18-x27	s2-s11	Callee-saved registers	Yes
x28-x31	t3-t6	Temporary registers	No

实验任务——汇编代码生成

- RISC-V 64GC ABI
  - 寄存器约定
  - 调用约定
- Chapter 25

Floating-point Register Convention

Name	ABI Mnemonic	Meaning	Preserved across calls?
f0-f7	ft0-ft7	Temporary registers	No
f8-f9	fs0-fs1	Callee-saved registers	Yes*
f10-f17	fa0-fa7	Argument registers	No
f18-f27	fs2-fs11	Callee-saved registers	Yes*
f28-f31	ft8-ft11	Temporary registers	No

gnu工具链

- $riscv64-unknown-elf-gcc
  -S test.s test.c

```c
1 int main(void)
2 {
3     int a = 2;
4     int b = 3;
5     int c = a + b;
6     print_int(c);
7     return 0;
8 }
```

```asm
 1      .file   "test.c"
 2      .option nopic
 3      .attribute arch, "rv64i2p0_m2p0_a2p0_f2p0_d2p0_c2p0"
 4      .attribute unaligned_access, 0
 5      .attribute stack_align, 16
 6      .text
 7      .align  1
 8      .globl  main
 9      .type   main, @function
10 main:
11      addi    sp,sp,-32
12      sd   ra,24(sp)
13      sd   s0,16(sp)
14      addi    s0,sp,32
15      li   a5,2
16      sw   a5,-20(s0)
17      li   a5,3
18      sw   a5,-24(s0)
19      lw   a4,-20(s0)
20      lw   a5,-24(s0)
21      addw    a5,a4,a5
22      sw   a5,-28(s0)
23      lw   a5,-28(s0)
24      mv   a0,a5
25      call    print_int
26      li   a5,0
27      mv   a0,a5
28      ld   ra,24(sp)
29      ld   s0,16(sp)
30      addi    sp,sp,32
31      jr   ra
32      .size   main, .-main
33      .ident  "GCC: (GNU) 10.2.0"
```

模拟器

- $riscv64-unknown-elf-gcc -o test test.c -L./libcact –lcact
- $spike pk test

```
.
├── libcact
│   ├── cactlib.c
│   ├── cactlib.h
│   ├── cactlib.o
│   └── libcact.a
├── test
└── test.c
```

```
1 int main(void)
2 {
3     int a = 2;
4     int b = 3;
5     int c = a + b;
6     print_int(c);
7     return 0;
8 }
```

```
lishuaijiang@ACT100:~/workspace/demo$ spike pk test
bbl loader
5
```

Thanks!

Q&A