

第十一章习题参考答案

包含题目：习题 11.8、11.10、11.12、11.14、11.16 和 11.17

题目 11.8 对于如下程序

(1) 构建符号表

(2) 该程序实现了什么

```

                                . data      x00007000
SaveR2:                        . space      4
SaveR4:                        . space      4
HelloWorld:                    . asciiz     " Hello, World!"
;

                                . text      x40000000
                                . global     main
main:                          SW           SaveR2( R0 ), r2
                                SW           SaveR4( R0 ), r4
;

                                ADDI         R2, R0, HelloWorld
LOOP:                          LB           R4, 0( R2 )
                                BEQZ        R4, Return
                                TRAP         x07
                                ADDI        R2, R2, #1
                                J            LOOP
;
Return:                        LW           R2, SaveR2( R0 )
                                LW           R4, SaveR4( R0 )
                                TRAP         x00

```

解答

(1) 构建符号表：

| 符号 | 地址 |
|------------|------------|
| SaveR2 | x0000 7000 |
| SaveR4 | x0000 7004 |
| HelloWorld | x0000 7008 |
| main | x4000 0000 |
| LOOP | x4000 000C |
| Return | x4000 0020 |

(2) 该程序在不改变 R2 和 R4 值的情况下，输出“Hello, World!”。

题目 11.10 对于如下程序

(1) 将该程序翻译为机器语言程序。

(2) 假设在这个程序执行之前，在 NUM1 中设置了一个正整数的值，该程序实现了什么？

```

                                . data      x30000000
NUM1:                          . space     4
NUM2:                          . space     4
                                ;

                                . text      x40000000
                                . global    main
main:                          ADDI        R1,R0,#0
                                LW          R2,NUM1(R0)
                                ANDI        R3,R2,#1
                                BNEZ        R3,NEXT
                                SUBI        R2,R2,#1
NEXT:                          ADD         R1,R1,R2
                                SUBI        R2,R2,#2
                                SLEI        R3,R2,#0
                                BEQZ        R3,NEXT
                                SW          NUM2(R0),R1
                                TRAP        x00

```

解答

(1) 第一步，建立符号表

| 符号 | 地址 |
|------|------------|
| NUM1 | x3000 0000 |
| NUM2 | x3000 0004 |
| main | x4000 0000 |
| NEXT | x4000 0014 |

第二步，翻译为机器语言

| 地址 | 数据 | 解释 |
|------------|--|------------------|
| x4000 0000 | 000001 00000 00001 0000 0000 0000 0000 | ADDI R1, R0, #0 |
| x4000 0004 | 001100 00000 00100 0011 0000 0000 0000 | LHI R4, x3000 |
| x4000 0008 | 011100 00100 00010 0000 0000 0000 0000 | LW R2, x0000(R4) |
| x4000 000C | 001001 00010 00011 0000 0000 0000 0001 | ANDI R3, R2, #1 |
| x4000 0010 | 101001 00011 00000 0000 0000 0000 0100 | BNEZ R3, NEXT |
| x4000 0014 | 000011 00010 00010 0000 0000 0000 0001 | SUBI R2, R2, #1 |
| x4000 0018 | 000000 00001 00010 00001 00000 000001 | ADD R1, R1, R2 |
| x4000 001C | 000011 00010 00010 0000 0000 0000 0010 | SUBI R2, R2, #2 |
| x4000 0020 | 010010 00010 00011 0000 0000 0000 0000 | SLEI R3, R2, #0 |
| x4000 0024 | 101000 00011 00000 1111 1111 1111 0000 | BEQZ R3, NEXT |
| x4000 0028 | 011101 00100 00001 0000 0000 0000 0100 | SW x0004(R4), R1 |
| x4000 002C | 110000 00000 00000 0000 0000 0000 0000 | TRAP x00 |

(2) 该程序将不大于 NUM1 中的所有奇数求和，存入 NUM2 中。

题目 11.12 如下程序存在一个错误，指出该错误并进行修复。该错误可以在汇编时还是在运行时被检测出来？

```

                . data      x30000000
NUM:           . space     4
;
                . text      x40000000
                . global    main
main:          ADDI        R1, R0, #100000
                SW         NUM(R0), R1
                TRAP        x00

```

解答 该程序的错误在 `main: ADDI R1, R0, #100000` 中，I-型 DLX 指令中的立即数为 16 位补码整数，能表示的十进制范围为 $-2^{15} \sim 2^{15} - 1$ ，即 $-32768 \sim 32767$ ，100000 在此范围之外。

要修复该错误，可以将该指令替换为 `LHI R2, x0001; ADDI R1, R2, x86A0` 或 `ADDI R1, R0, #20000; ADDI R1, R0, #20000`

由于汇编时立即数溢出，可知该错误是在汇编时被检测出来的。

题目 11.14 如下 DLX 程序用于判断一个字符串是否是“回文”（正向读和反向读都相同的字符串），例如，`strts` 就是回文。假设字符串起始于存储单元 `x30000000`。如果该字符串是回文，程序以 `R1` 的值为 1 结束，否则 `R1` 为 0。填空，将程序补充完整。

| | | |
|---------|----------|--------------|
| | . data | x30000000 |
| STRING: | . ASCIIZ | " string" |
| ; | | |
| | . text | x40000000 |
| | . global | main |
| main: | ADDI | R1,R0,#0 |
| | ADDI | R2,R0,STRING |
| | ADDI | R3,R2,#0 |
| LOOP1: | LB | R4,0(R2) |
| | BEQZ | R4,NEXT |
| | ADDI | R2,R2,#1 |
| | J | LOOP1 |
| NEXT: | | |
| LOOP2: | LB | R4,0(R2) |
| | | |
| | SEQ | R6,R4,R5 |
| | BEQZ | R6,EXIT |
| | | |
| | SLE | R6,R2,R3 |
| | BEQZ | R6,LOOP2 |
| | ADDI | R1,R1,#1 |
| EXIT: | TRAP | x00 |

解答 填入的代码依次为:

```

1  LB  R5, 0(R3)
2  ADDI R2, R2, #1
3  ADDI R3, R3, #1
4  ADDI R1, R1, #1 或 ADDI R1, R6, #0

```

题目 11.16 使用 DLX 的 C 编译器对如下局部变量声明进行编译, 写出 DLX 汇编代码。

```

1  char c = 'c';
2  int x = 5;
3  int y;

```

解答

```

1  ADDI R5, R0, x63 或 ADDI R5, R0, #99
2  SW  -4(R30), R5
3  ADDI R5, R0, #5
4  SW  -8(R30), R5
5  ADDI R5, R0, #0
6  SW  -12(R30), R5

```

题目 11.17 使用 DLX 的 C 编译器对如下 switch 语句进行编译, 写出 DLX 汇编代码。

```

1  int i, j, k, x;

```

```
2  ...
3  switch (i){
4      case 0:
5          x = k + j;
6          break;
7      case 1:
8          x = k - j;
9          break;
10     case 2:
11         x = i + k;
12         break;
13     default:
14         x = i - k;
15 }
```

解答

```
1      ADDI R16, R0, #0; i
2      ADDI R17, R0, #0; j
3      ADDI R18, R0, #0; k
4      ADDI R19, R0, #0; x
5      .....
6      SEQI R1, R16, #0
7      BNEZ R1, Case_0
8      SEQI R1, R16, #1
9      BNEZ R1, Case_1
10     SEQI R1, R16, #2
11     BNEZ R1, Case_2
12     J Case_default
13 Case_0:  ADD R19, R18, R17
14         J Done
15 Case_1:  SUB R19, R18, R17
16         J Done
17 Case_2:  ADD R19, R16, R18
18         J Done
19 Case_default: SUB R19, R16, R18
20         J Done
21 Done:   .....
```