

Projeto de Compilador

Etapa 6 - Assembly

Matheus Lima

Programa de exemplo

```
etapa6 > ≡ testeasm.txt
1 // UFRGS - Compiladores - Marcelo Johann - 2024/1
2
3 char a:'a';
4 bool b:true;
5 char c:'x';
6 int d:0;
7 int number:199;
8 int i:1;
9 int v[10]: 'a' 1 'b' 3 4 5 6 7 8 9;
10 int matrix[100];
11 float f:2.5; // 0.0 .0 0. are all allowed
12 bool condicao:true;
13
14 int main ()
15 {
16     print "digite o valor de d: \n";
17     read int d;
18
19     i = a + i;
20     print "i = a + i -> ";
21     print int i;
22
23     d = d - 5;
24     print "d = d - 5 -> ";
25     print int d;
26
27     f = 2.0 * 9.;
28     print "f = 2.0 * 9. -> ";
29     print float f;
30
31     f = 5.5 / 2.;
32     print "f = 5.5 / 2. -> ";
33     print float f;
34
35     b = ~b;
36     print "b = ~b -> ";
37     print int b;
38
39     condicao = b | true;
```

```
40
41     if (condicao){
42         i = 0;
43         print "i = 0 -> ";
44         print int i;
45
46         while (i<10){
47             print "i = ";
48             print int i;
49
50             i = i + 1;
51         }
52
53         number = v[8];
54         print "number = v[8] -> ";
55         print int number;
56     }
57
58     return 0;
59 }
60
61 // end of file
62
63
64
```

Assembly gerado - Strings

```
etapa6 > asm out.s
1  ## FIXED INIT
2  .section  __TEXT,__cstring,cstring_literals
3  printIntStr: .asciz "%d\n"
4  printFloatStr: .asciz  "%f\n"
5  readIntStr: .asciz  "%d"
6  readFloatStr: .asciz  "%f"
7  printStr: .asciz  "%s"
8  _STRING16041765196113750995: .asciz "b = ~b -> "
9  _STRING12328719795569653691: .asciz "d = d - 5 -> "
10 _STRING6048186195195000451: .asciz "digite o valor de d: \n"
11 _STRING11676679939604644714: .asciz "i = 0 -> "
12 _STRING9790939547036807759: .asciz "i = a + i -> "
13 _STRING5937925058351471584: .asciz "number = v[8] -> "
14 _STRING10042128117035228350: .asciz "f = 5.5 / 2. -> "
15 _STRING6950845483759: .asciz "i = "
16 _STRING10688820508414138744: .asciz "f = 2.0 * 9. -> "
17
18 .section  __TEXT,__text,regular,pure_instructions
19
```

Assembly gerado - Código

```
20  ## TAC_BEGINFUN
21  .globl _main
22  _main:
23  pushq %rbp
24  movq  %rsp, %rbp
25
26  ## TAC_PRINTSTR
27  leaq  printStr(%rip), %rdi
28  leaq  _STRING6048186195195000451(%rip), %rsi
29  movb  $0, %al
30  callq _printf
31
32  ## TAC_READ
33  leaq  readIntStr(%rip), %rdi
34  leaq  _INTd(%rip), %rsi
35  movl  $0, %eax
36  callq _scanf
37
38  ## TAC_ADD  xorl  %eax, %eax
39  movq  _INTmYWeeirT_emp0@GOTPCREL(%rip), %rcx
40  movl  _CHARa(%rip), %edx
41  addl  _INTi(%rip), %edx
42  movl  %edx, (%rcx)
43
44  ## TAC_COPY
45  xorl  %eax, %eax
46  movq  _INTi@GOTPCREL(%rip), %rcx
47  movl  _INTmYWeeirT_emp0(%rip), %edx
48  movl  %edx, (%rcx)
```

```
50  ## TAC_PRINTSTR
51  leaq  printStr(%rip), %rdi
52  leaq  _STRING9790939547036807759(%rip), %rsi
53  movb  $0, %al
54  callq _printf
55
56  ## TAC_PRINTINT
57  leaq  printIntStr(%rip), %rdi
58  movl  _INTi(%rip), %esi
59  movb  $0, %al
60  callq _printf
61
62  ## TAC_SUB  xorl  %eax, %eax
63  movq  _INTmYWeeirT_emp1@GOTPCREL(%rip), %rcx
64  movl  _INTd(%rip), %edx
65  subl  _INT5(%rip), %edx
66  movl  %edx, (%rcx)
67
68  ## TAC_COPY
69  xorl  %eax, %eax
70  movq  _INTd@GOTPCREL(%rip), %rcx
71  movl  _INTmYWeeirT_emp1(%rip), %edx
72  movl  %edx, (%rcx)
73
74  ## TAC_PRINTSTR
75  leaq  printStr(%rip), %rdi
76  leaq  _STRING12328719795569653691(%rip), %rsi
77  movb  $0, %al
78  callq _printf
79
```

Assembly gerado - Código

```
80  ## TAC_PRINTINT
81      leaq    printIntStr(%rip), %rdi
82      movl    _INTD(%rip), %esi
83      movb    $0, %al
84      callq   _printf
85
86  ## TAC_MUL    xorl    %eax, %eax
87      movq    _FLOATmYWeeirT_emp2@GOTPCREL(%rip), %rdx
88      movss   _FLOAT2.0(%rip), %xmm0
89      mulss   _FLOAT9.0(%rip), %xmm0
90      movss   %xmm0, (%rdx)
91
92  ## TAC_COPY
93      xorl    %eax, %eax
94      movq    _FLOATf@GOTPCREL(%rip), %rcx
95      movl    _FLOATmYWeeirT_emp2(%rip), %edx
96      movl    %edx, (%rcx)
97
98  ## TAC_PRINTSTR
99      leaq    printStr(%rip), %rdi
100     leaq    _STRING10688820508414138744(%rip), %rsi
101     movb    $0, %al
102     callq   _printf
103
104  ## TAC_PRINTREAL
105     movss   _FLOATf(%rip), %xmm0
106     cvtss2sd %xmm0, %xmm0
107     leaq    printFloatStr(%rip), %rdi
108     movb    $1, %al
109     callq   _printf
110
```

```
111  ## TAC_DIV    xorl    %eax, %eax
112     movq    _FLOATmYWeeirT_emp3@GOTPCREL(%rip), %rcx
113     movss   _FLOAT5.5(%rip), %xmm0
114     divss   _FLOAT2.0(%rip), %xmm0
115     movss   %xmm0, (%rcx)
116
117  ## TAC_COPY
118     xorl    %eax, %eax
119     movq    _FLOATf@GOTPCREL(%rip), %rcx
120     movl    _FLOATmYWeeirT_emp3(%rip), %edx
121     movl    %edx, (%rcx)
122
123  ## TAC_PRINTSTR
124     leaq    printStr(%rip), %rdi
125     leaq    _STRING10042128117035228350(%rip), %rsi
126     movb    $0, %al
127     callq   _printf
128
129  ## TAC_PRINTREAL
130     movss   _FLOATf(%rip), %xmm0
131     cvtss2sd %xmm0, %xmm0
132     leaq    printFloatStr(%rip), %rdi
133     movb    $1, %al
134     callq   _printf
135
136  ## TAC_NOT
137     movl    _BOOLb(%rip), %eax
138     xorl    $1, %eax
139     movl    %eax, _BOOLmYWeeirT_emp4(%rip)
140
```

Assembly gerado - Código

```
141  ## TAC_COPY
142      xorl    %eax, %eax
143      movq    _BOOLb@GOTPCREL(%rip), %rcx
144      movl    _BOOLmYWeeirT_emp4(%rip), %edx
145      movl    %edx, (%rcx)
146
147  ## TAC_PRINTSTR
148      leaq    printStr(%rip), %rdi
149      leaq    _STRING16041765196113750995(%rip), %rsi
150      movb    $0, %al
151      callq   _printf
152
153  ## TAC_PRINTBOOL
154      movb    _BOOLb(%rip), %al
155      andb    $1, %al
156      movzbl  %al, %esi
157      leaq    printIntStr(%rip), %rdi
158      movb    $0, %al
159      callq   _printf
160
161  ## TAC OR
162      movl    $0, %eax
163      testl   %eax, %eax
164      jnz     .Ltrue1
165      movl    $1, %eax
166      jmp     .Lend1
167  .Ltrue1:
168      movl    $1, %eax
169  .Lend1:
170      movl    %eax, _BOOLmYWeeirT_emp5(%rip)
171
```

```
172  ## TAC_COPY
173      xorl    %eax, %eax
174      movq    _BOOLcondicao@GOTPCREL(%rip), %rcx
175      movl    _BOOLmYWeeirT_emp5(%rip), %edx
176      movl    %edx, (%rcx)
177
178  ## TAC JFALSE
179      movl    _BOOLcondicao(%rip), %eax
180      testl   %eax, %eax
181      jz      .mYLabe_l2
182
183  ## TAC_COPY
184      xorl    %eax, %eax
185      movq    _INTi@GOTPCREL(%rip), %rcx
186      movl    _INT0(%rip), %edx
187      movl    %edx, (%rcx)
188
189  ## TAC_PRINTSTR
190      leaq    printStr(%rip), %rdi
191      leaq    _STRING11676679939604644714(%rip), %rsi
192      movb    $0, %al
193      callq   _printf
194
195  ## TAC_PRINTINT
196      leaq    printIntStr(%rip), %rdi
197      movl    _INTi(%rip), %esi
198      movb    $0, %al
199      callq   _printf
200
```

Assembly gerado - Código

```
201  ## TAC LABEL
202  .mYLabe_l0:
203
204  ## TAC LESS
205      movl    _INTi(%rip), %eax
206      cmpl    _INT10(%rip), %eax
207      setl    %al
208      movzbl  %al, %eax
209      movl    %eax, _BOOLmYWeeirT_emp6(%rip)
210
211  ## TAC JFALSE
212      movl    _BOOLmYWeeirT_emp6(%rip), %eax
213      testl   %eax, %eax
214      jz      .mYLabe_l1
215
216  ## TAC_PRINTSTR
217      leaq    printStr(%rip), %rdi
218      leaq    _STRING6950845483759(%rip), %rsi
219      movb    $0, %al
220      callq   _printf
221
222  ## TAC_PRINTINT
223      leaq    printIntStr(%rip), %rdi
224      movl    _INTi(%rip), %esi
225      movb    $0, %al
226      callq   _printf
227
```

```
228  ## TAC_ADD  xorl    %eax, %eax
229      movq    _INTmYWeeirT_emp7@GOTPCREL(%rip), %rcx
230      movl    _INTi(%rip), %edx
231      addl    _INT1(%rip), %edx
232      movl    %edx, (%rcx)
233
234  ## TAC_COPY
235      xorl    %eax, %eax
236      movq    _INTi@GOTPCREL(%rip), %rcx
237      movl    _INTmYWeeirT_emp7(%rip), %edx
238      movl    %edx, (%rcx)
239
240  ## TAC JUMP
241      jmp     .mYLabe_l0
242
243  ## TAC LABEL
244  .mYLabe_l1:
245
246  ## TAC_VEC
247      xorl    %eax, %eax
248      movq    _INTmYWeeirT_emp8@GOTPCREL(%rip), %rcx
249      movl    _INTv+32(%rip), %edx
250      movl    %edx, (%rcx)
251
252  ## TAC_COPY
253      xorl    %eax, %eax
254      movq    _INTnumber@GOTPCREL(%rip), %rcx
255      movl    _INTmYWeeirT_emp8(%rip), %edx
256      movl    %edx, (%rcx)
257
```

Assembly gerado - Código

```
258  ## TAC_PRINTSTR
259      leaq    printStr(%rip), %rdi
260      leaq    _STRING5937925058351471584(%rip), %rsi
261      movb    $0, %al
262      callq   _printf
263
264  ## TAC_PRINTINT
265      leaq    printIntStr(%rip), %rdi
266      movl    _INTnumber(%rip), %esi
267      movb    $0, %al
268      callq   _printf
269
270  ## TAC LABEL
271  .myLabe_l2:
272
273  ## TAC RETURN
274      movl    _INT0(%rip), %eax
275
276  ## TAC_ENDFUN
277      popq    %rbp
278      retq
279
```


Assembly gerado - Data Section

```
280 # DATA SECTION
281     .section    __DATA,__data
282
283     _INT0:      .long    0
284     _INT1:      .long    1
285     _INT3:      .long    3
286     _INT4:      .long    4
287     _INT5:      .long    5
288     _INT6:      .long    6
289     _INT7:      .long    7
290     _INT8:      .long    8
291     _INT9:      .long    9
292     _FLOAT5.5:  .float    5.5
293     _FLOAT2.:   .float    2.
294     _BOOLtrue:  .byte     1
295     _CHAR98:    .long     98
296     _INT10:     .long     10
297     _INT100:    .long     100
298     _FLOAT9.:   .float    9.
299     _FLOAT2.5:  .float    2.5
300     _CHAR120:   .long     120
301     _CHAR97:    .long     97
302     _FLOAT2.0:  .float    2.0
303     _INT199:    .long     199
```

```
304     .globl _CHARa
305     _CHARa:     .long     97
306     .globl _BOOLb
307     _BOOLb:     .long     1
308     .globl _CHARc
309     _CHARc:     .long     120
310     .globl _INTd
311     _INTd:      .long     0
312     .globl _INTnumber
313     _INTnumber: .long     199
314     .globl _INTi
315     _INTi:      .long     1
316     .globl _INTv
317     _INTv:
318         .long     'a'
319         .long     1
320         .long     'b'
321         .long     3
322         .long     4
323         .long     5
324         .long     6
325         .long     7
326         .long     8
327         .long     9
328     .comm _INTmatrix,400,4
329     .globl _FLOATf
330     _FLOATf:    .float     2.5
331     .globl _BOOLcondicao
332     _BOOLcondicao: .long     1
333
334     .comm _INTmYWeeirT_emp0,4,2
335     .comm _INTmYWeeirT_emp1,4,2
336     .comm _FLOATmYWeeirT_emp2,4,2
337     .comm _FLOATmYWeeirT_emp3,4,2
338     .comm _BOOLmYWeeirT_emp4,1,0
339     .comm _BOOLmYWeeirT_emp5,1,0
340     .comm _BOOLmYWeeirT_emp6,1,0
341     .comm _INTmYWeeirT_emp7,4,2
342     .comm _INTmYWeeirT_emp8,4,2
```

Saída

```
etapa6 — -ba
Compilation succefull with no semantic errors!
File has 64 lines
[(base) Matheus-2:etapa6 MatheusLima$ gcc out.s
[(base) Matheus-2:etapa6 MatheusLima$ ./a.out
digite o valor de d:
9
i = a + i -> 98
d = d - 5 -> 4
f = 2.0 * 9. -> 18.000000
f = 5.5 / 2. -> 2.750000
b = ~b -> 0
i = 0 -> 0
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
number = v[8] -> 8
```

Caso incompleto - Funções

```
58     number = func(i, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14);
59
60     return 0;
61 }
62
63 int func(int a1, int b1, int c1, int d1, int e1, int f1, int g1, int h1, int i1, int j1, int k1, int l1, int m1, int n1){
64     //
65 }
66
```

```
273     ## TAC_CALL
274     pushq    %rbx
275     pushq    %r14
276     pushq    %r15
277     pushq    %r12
278     pushq    %r13
279     subq     $104, %rsp
280     movl     _INT14(%rip), %edi
281     movl     _INT13(%rip), %esi
282     movl     _INT12(%rip), %edx
283     movl     _INT11(%rip), %ecx
284     movl     _INT10(%rip), %r8d
285     movl     _INT9(%rip), %r9d
286     movl     _INT8(%rip), %r13d
287     movl     _INT7(%rip), %r12d
288     movl     _INT6(%rip), %r15d
289     movl     _INT5(%rip), %r14d
290     movl     _INT4(%rip), %ebx
291     movl     _INT3(%rip), %r11d
292     movl     _INT2(%rip), %r10d
293     movl     _INT1(%rip), %eax
294     call     _func
295     movl     %eax, _INTmYWeeirT_emp9(%rip)
296
```

```
---
297     ## TAC_COPY
298     xorl     %eax, %eax
299     movq     _INTnumber@GOTPCREL(%rip), %rcx
300     movl     _INTmYWeeirT_emp9(%rip), %edx
301     movl     %edx, (%rcx)
302
303     ## TAC_RETURN
304     movl     _INT0(%rip), %eax
305
306     ## TAC_ENDFUN
307     popq     %rbp
308     retq
309
310     ## TAC_BEGINFUN
311     .globl   _func
312     _func:
313     pushq    %rbp
314     movq     %rsp, %rbp
315
316     ## TAC_ENDFUN
317     popq     %rbp
318     retq
```