

REDUX-V: Algoritmo Base

Nico I. G. Ramos GRR20210574

Setembro 2024

1 Assembly

```
; int i;  
; int a = 3;  
; int b = 42;  
; int *c = 10;  
;  
; for (i = 0; i < a; i++) {  
;   c[i] = b + i;  
; }
```

__setup:

```
0x00  xor r0, r0      ; r0 = 0  
0x01  addi 3          ; r0 = 3  
0x02  mov r3, r0      ; r3 = 0  
                      ; r3 := i + 1  
0x03  xor r0, r0      ; r0 = r0  
0x04  addi 4          ; r0 = 4  
0x05  mov r1, r0      ; r1 = r0  
                      ; r1 := 4  
0x06  xor r0, r0      ; r0 = 0  
0x07  addi 12         ; r0 = 1111 1100  
0x08  slr r0, r1      ; r0 = r0 << r1  
                      ; r0 := 1111 1100 << 4 = 1100 0000  
0x09  srr r0, r1      ; r0 = r0 >> r1  
                      ; r0 := 1100 0000 >> 4 = 0000 1100  
0x0A  mov r2, r0      ; r2 = r0  
                      ; r2 := 12
```

__loop:

```
0x0B  xor r0, r0      ; r0 = 0  
0x0C  addi 1          ; r0 = 0000 0001  
0x0D  slr r0, r0      ; r0 = 0000 0010  
0x0E  slr r0, r0      ; r0 = 0010 0000 = 0x20  
0x0F  addi 2          ; r0 = 0x22  
                      ; r0 := __loop_end  
0x0F  mov r1, r0      ; r1 = r0  
                      ; r1 := __loop_end  
0x10  brzr r3, r1     ; PC = r3 == 0 ? r1 : PC + 1  
                      ; PC = i == 0 ? __loop_end : PC + 1
```

```

0x11  mov r0, r3      ; r0 = r3
0x12  addi -1         ; r0 -= 1
0x13  mov r3, r0      ; r3 = r0
                        ; r3 := i—
0x14  xor r0, r0      ; r0 = 0
0x15  addi 2          ; r0 = 2
0x16  mov r1, r0      ; r1 = r0 = 0000 0010
0x17  slr r0, r1      ; r0 = 0000 1000
0x18  add r0, r1      ; r0 = 0000 1010
0x19  slr r0, r1      ; r0 = 0010 1000
0x1A  add r0, r1      ; r0 = 0010 1010
                        ; r0 := 42
0x1B  add r0, r3      ; r0 = r0 + r3
                        ; r0 := b + i
0x1C  mov r1, r0      ; r1 = r0
                        ; r1 := b + i
0x1D  st r1, r2       ; M[r2] = r1
                        ; M[c] := b + 1
0x1E  mov r0, r2      ; r0 = r2
0x1F  addi -1         ; r0 -= 1
0x20  mov r2, r0      ; r2 = r0
                        ; r2 := c - 1
0x21  ji -7           ; PC = 0010 0001 + 1111 1001 = 0001 1010 = 0x0B
                        ; PC := __loop
__loop_end:
0x22

```

2 Código de Máquina

```

10110000
01000011
01111100
10110000
01000100
01110100
10110000
01001010
11100001
11110001
01111000
10110000
01000001
11100000
11100000
01000010
01110100
00001101
01110011
01001111

```

01111100
10110000
01000010
01110100
11100001
11000001
11100001
11000001
11000011
01110100
00110110
01110010
01001111
01111000
00011001