

### Lista de instruções Assembly

| Instrução   | Significado  |
|---|--|
| MOV a, b  | Carregar Endereço de b em a  |
| MOV a, [b]  | Carregar Valor de b em a   |
| MOV a, [a + 4]  | Carregar Valor de b + 4 bytes (Apontamento, deslocamento relativo) em a  |
| MOV b, 3  | Atribui 3 a b  |
| ADD b, 4  | Soma 4 ao valor de b   |
| SUB EAX, 4  | subtrai 4 de EAX   |
| MOV ECX, 3<br>label:<br>** outras instruções**<br>LOOP label                                | Estrutura for(i = 3; i > 0; i ----), o registrador ECX é o registrador de contagem (automaticamente)   |
| CMP EAX, 0<br>Je labelIF ;vai pro if<br>; else  | Estrutura if (i == 0) em<br>Para outras condicionais troque Je pelo que está na tabela <a href="#">Condicionais</a><br>OBS: Para fazer várias comparações faça vários CMP na sequência             |
| CMP a,b   | if a < b --> registradores "ZF = 0" e "CF = 1"<br>if a > b --> registradores "ZF = 0" and "CF = 0"<br>if a = b --> registradores "ZF = 1" and "CF = 0"   |
| PUSH a  | Coloca a na pilha  |
| POP a   | Remove último valor da pilha e move para a   |
| CALL label  | Chama um procedimento, empilha o endereço de retorno   |
| RET   | Retorna do CALL  |
| mov edx, 0<br>mov eax, 5<br>mov ecx, 2 ; divisor<br>idiv ecx                                | Exemplo de divisão inteira, onde edx conterá o resto (do div)<br>eax é o numerador e ecx é o denominador. O resultado será colocado em eax após div.   |
| imul ecx, 3   | Atribui a ecx 3 vezes o seu valor  |
| imul edx, eax, 5  | Atribui a edx 5 vezes o valor contido em eax   |
| JMP label   | Pula para o label especificado   |
| %include "io.inc"<br><br>...<br><br>PRINT_UDEC 4, EAX<br>PRINT_UDEC 4, EAX<br>PRINT_CHAR '' | Imprimindo resultados... há a opção de print_char ao qual imprime caracter. print_udesc e print_dec imprimem número decimal sem sinal e com sinal, respectivamente (4 indica inteiros de 4 bytes). |