

LISTA DE INTRODUÇÃO Á ARQUITETURA DE COMPUTADORES

1. Codificar em *assembly* as funcionalidades indicadas abaixo:

main() **chama** *ordenar (v, n)* **chama** *shiftmaior (v, a, b)*, onde:

ordena (v, n) ordena um vetor *v* de inteiros composto por *n* elementos;

shiftmaior (v, a, b) desloca o elemento de maior valor entre os índices *a* e *b* do vetor *v* para a posição *b*;

Enfatizar o uso de instruções para manipulação de strings. Construir os RA's de *ordena* e *shiftmaior*.

<pre>#include <stdio.h> #include <stdlib.h> void shiftmaior (int *v, int a, int b) { asm (".intel_syntax noprefix \n\att_syntax prefix"); } void ordena (int *v, int n) { asm (".intel_syntax noprefix \n\ ... call _shiftmaior \n\att_syntax prefix"); } void showvet (int *v, int n) { int i; for (i = 0; i < n; i++) printf ("%d = %d\n", i, v[i]); }</pre>	<pre>int v[10] = {10, 9, 8, 7, 6, 1, 2, 3, 4, 5}, n = 10; int main(int argc, char *argv[]) { asm (".intel_syntax noprefix \n\ sub esp, 8 \n\ mov eax, _n \n\ mov [esp+4], eax \n\ mov eax, OFFSET _v \n\ mov [esp], eax \n\ call _ordena \n\ add esp, 8 \n\ .att_syntax prefix"); showvet (v, n); return 0; }</pre>
---	---

2. Considere o seguinte programa escrito em *C* com suas respectivas funções.

<pre>#include <stdio.h> #include <stdlib.h> int strlen (char *s) { const char *sc; for (sc = s; *sc ; sc++) /* nothing */; return sc - s; } char *strend (char *s) { return s + strlen (s); } char *strcpy (char *dest, const char *src) { char *tmp = dest; while (*dest++ = *src++) /* nothing */; return tmp; } char *strcat (char *dest, const char *src) { strcpy (strend (dest), src); return dest; } char *strchr (char *s, int c) {</pre>	<pre>int contachar (char *s, int c) { int t; for (t = 0; *s; s++) t+= (*s == (char) c); return t; } int ispalin (char *s) { char *p = s; while (*p) p++; // p = strend(s); p--; for (;s < p; s++, p--) if (*s != *p) return 0; return 1; } char sa[100], sb[100], *pa, *pb, *pc, *pd; int ta, ca, cb;</pre>
--	--

<pre> for (; *s; s++) if (*s == (char) c) return s; return NULL; } char *strrchr (char *s, int c) { char *p; for (p = strend(s)-1; p >= s; p--) if (*p == (char) c) return p; return NULL; } </pre>	<pre> int main(int argc, char *argv[]) { scanf ("%^[^\n]", sb); pa = strend (sa); pb = strcpy (sa, sb); ta = strlen (sa); pc = strchr (sa, 'x'); pd = strrchr (sa, 'x'); ca = contachar (sa, 'x'); ca = ispalin(sa); return 0; } </pre>
--	--

Implementar em assembly todas as funções indicadas em **negrito** utilizando instruções para manipulação de strings (scasb, lodsb, stosb, rep, repne, repe).

MOV REG/MEM, REG/MEM/IMM MOVSX REG, REG/MEM MOVZX REG, REG/MEM XCHG REG/MEM, REG/MEM ADD REG/MEM, REG/MEM/IMM ADC REG/MEM, REG/MEM/IMM INC REG/MEM SUB REG/MEM, REG/MEM/IMM SBB REG/MEM, REG/MEM/IMM DEC REG/MEM LEA REG, MEM CBW CWD CWDE CDQ MUL REG/MEM IMUL REG/MEM IMUL REG, REG/MEM/IMM IMUL REG, REG/MEM, IMM DIV REG/MEM IDIV REG/MEM CMP REG/MEM, REG/MEM/IMM JMP LABEL J{C, O, S, P, Z} JN{C, O, S, P, Z} JP{E, O} LABEL J{E, A, B, G, L} JN{E, A, B, G, L} LABEL JN{A, B, G, L} E LABEL JECXZ LABEL	AND TEST REG/MEM, REG/MEM/IMM OR REG/MEM, REG/MEM/IMM XOR REG/MEM, REG/MEM/IMM NOT REG/MEM NEG REG/MEM SHL – SAL REG/MEM, IMM/CL SHR – SAR REG/MEM, IMM/CL ROL – RCL REG/MEM, IMM/CL ROR – RCR REG/MEM, IMM/CL SHLD REG/MEM, REG, IMM/CL SHRD REG/MEM, REG, IMM/CL BSF REG, REG/MEM BSR REG, REG/MEM BT BTS BTR BTC REG/MEM, REG/IMM SetCC REG₈/MEM₈ LOOP LABEL LOOPE/LOOPZ LABEL LOOPNE/LOOPNZ LABEL REP STRING REPE/REPZ STRING REPNE/REPNZ STRING PUSH REG/MEM/IMM POP REG/MEM/IMM IN {AL, AX, EAX}, IMM/DX OUT IMM/DX, {AL, AX, EAX} INT IMM RET IMM STD STI CLD CLI
<i>Deve ser dada atenção à combinação MEM-MEM e ao tamanho dos operandos.</i>	
LODS{B, W, D} {AL, AX, EAX} = [ESI] ESI = ESI ± {1, 2, 4} STOS{B, W, D} [EDI] = {AL, AX, EAX} EDI = EDI ± {1, 2, 4} MOVS{B, W, D} LODS{B, W, D} STOS{B, W, D} Não alteram FLAGS	SCAS{B, W, D} CMP {AL, AX, EAX}, [EDI] {AL, AX, EAX} – [EDI] EDI = EDI ± {1, 2, 4} CMPS{B, W, D} CMP [ESI], [EDI] [ESI] – [EDI] {EDI, ESI} = {EDI, ESI} ± {1, 2, 4} Alteram FLAGS

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	I D	V I P	V I F	A C	V M	R F	0	N T	I O P L	O F	D F	I F	T F	S F	Z F	0	A F	0	P F	1	C F