

IA32 : Modos de Endereçamento

Guião: G-4

Apresentação

Este guião tem em vista introduzir o jogo de instruções do IA32 e assistir na utilização dos diferentes modos de acesso a dados.

Questão 1 (cálculo de endereços)

Considere os valores apresentados nos dois quadros abaixo:

Endereço	Valor
0x100	0xFF
0x104	0xAB
0x108	0x13
0x10C	0x11

Registo	Valor
%eax	0x100
%ecx	0x1
%edx	0x3

- a) Complete os campos vazios do quadro que seque

Operando	Valor
%eax	
0x104	
\$0x108	
(%eax)	
4(%eax)	
9(%eax,%edx)	
260(%ecx,%edx)	
0xFC(,%ecx,4)	
(%eax,%edx,4)	

- b) Para cada uma das instruções, à esquerda, preencha os campos à direita, que referem respectivamente o destino (registo/endereço), o valor calculado e o valor dos bits de condição (zero, sinal, transporte, excesso)

Instrução	Destino	Valor	Z	S	C	O
addl %ecx, (%eax)						
subl %edx, 4(%eax)						
imull \$16, (%eax,%edx,4)						
incl 8(%eax)						
decl %ecx						
addl \$0x7fffffff, %edx						
addl \$-1, %eax						
andl \$-4, (%eax)						
cmpl %eax, %edx						
test 0x08(%eax), %edx						
subl %edx, %eax						

Questão 2 (Cálculo de expressões):

Abaixo os registos `%eax` e `%ecx` contêm respectivamente os valores `x` e `y`. Para cada uma das instruções à esquerda escreva a expressão que representa o valor do registo `%edx`, após a execução.

Instrução	Valor
<code>leal 6(%eax), %edx</code>	
<code>leal (%eax,%ecx), %edx</code>	
<code>leal (%eax,%ecx,4), %edx</code>	
<code>leal 7(%eax,%eax,8), %edx</code>	
<code>leal 0xA(,%ecx,4), %edx</code>	
<code>leal 9(%eax,%ecx,2), %edx</code>	
<code>leal 0xFFFFFFFDE(%eax,%ecx,4), %edx</code>	

Questão 3 (Movimentação de bits):

Use o editor de texto mais conveniente para escrever os dois módulos de funções C, abaixo.

1. Crie o código executável com os dois módulos e corra-o a partir do `gdb`.
 - Quais os endereços de memória correspondentes a todas as variáveis envolvidas?
 - Execute passo-a-passo a função `movBits`, registando para cada operação, no código C, os valores dos registos e das variáveis envolvidos nos cálculos.
 - Altere livremente o código das funções originais para explorar outras operações lógicas/aritméticas disponíveis no IA32.

main.c	imprime.c
<pre> int x=128, n=31; unsigned u; short s; char c; int movBits(void); main() { movBits(); return 0; } </pre>	<pre> extern int x, n; extern unsigned u; extern short s; extern char c; int movBits() { int t=-1; x <<= 3; u=x; s=x; c=x; u <<= 3; x=t; x >>= n; u=x; s=x; c=x; u >>= n;} </pre>