

Prova LM + Nome: Gabriel Fagundes Monteiro + RGM: 43560

Q01: Descreva um algoritmo que simule o modo como o computador executa programas e instruções

917
==

// Passar para a execução de instruções

PC = Endereço Inicial // contador de Programa (RIP)

Byte de Execução = True

while (Byte De Execução):

R = Memória[PC]

PC = PC + 1

Determina Tipo Instrução (IR, Tipo De Instrução)

Acha Dado (Tipo De Instrução, IR, Localização Dado, Dado Requerido)

If (Dado Requerido)

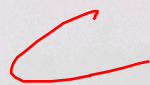
dado = Memória [Localização Dado]

Executa (Tipo De Instrução, dado, memória, AC, PC, Bit De Instrução)

C

Q.02: Se a for definido como 05_{16} e b for definido como $0F_{16}$, qual o conteúdo do registrador rax ?

O conteúdo será $0F05_{16}$



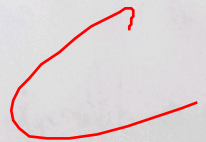
Q03: Se o registro rax estiver definido como 81.985.529.216.486.895, (123456789ABCDEF), qual é o conteúdo dos seguintes registros em hexadecimal?

alal EF

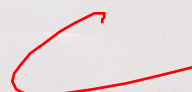
blax CDEF

cleax 89ABCDEF

dlrax 0123456789ABCDEF



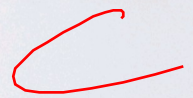
Q04: Na arquitetura Intel X86-84, quantos bytes podem ser armazenados em cada endereço?
No máximo, cada registrador alcança de 1 byte ou 8 bits a 8 bytes ou 64 bits por endereço.
Depende do tipo da variável ou registro



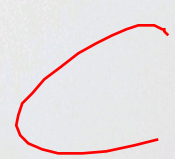
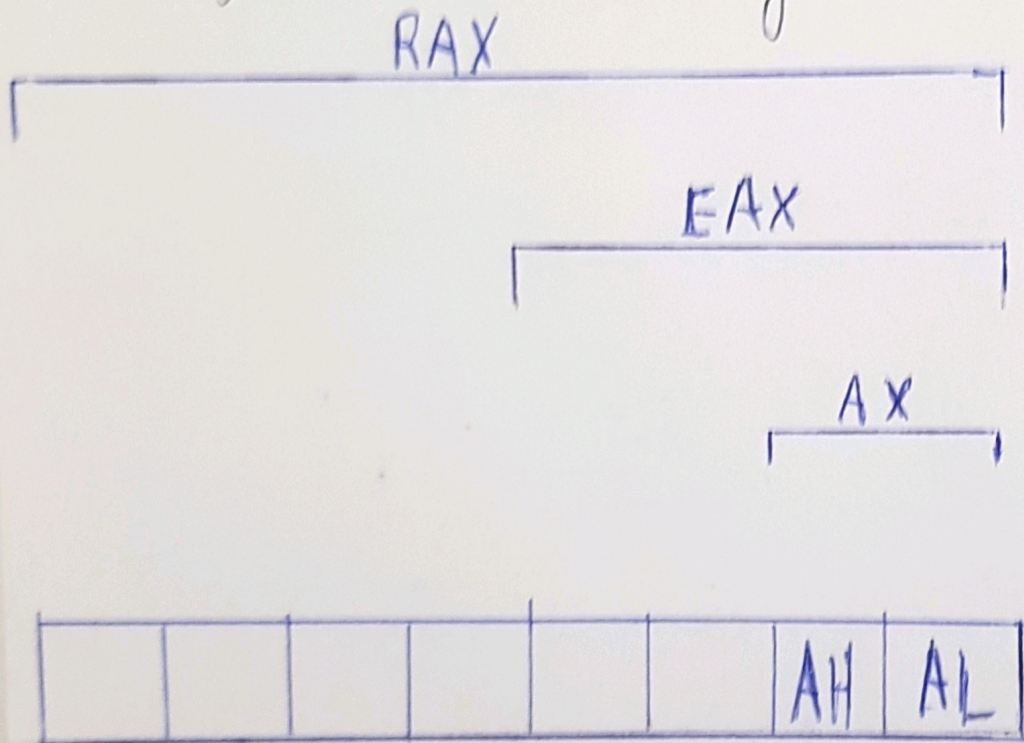
Q051 Dado o hexadecimal de 32 bits 004C4B4016, qual é o:

a) Bit Menos Significativo (LSB)

b) Bit Mais Significativo (MSB)



Q05: Faça um desenho do layout do registro acumulador



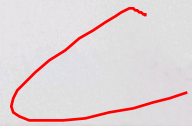
Q07: Quantos bits cada um dos seguintes registradores representa:

a) al 8

b) rcx 64

c) bx 16

d) edx 32



Q08: Responda as seguintes questões

a) Qual a diferença entre a instrução mov e a instrução lea? Dê exemplos

MOV R, Var carrega o endereço de Var em R ou o conteúdo de Var no caso de MOV R, [Var]

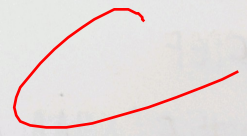
LEA - Carrega o endereço efetivo + deslocamento, se o deslocamento for 0, não é necessário adicionar. Assim, LEA R, [Var] faz o mesmo que MOV R, Var. Contudo, a instrução MOV R, Var + deslocamento não é aceita. Para isso temos que usar LEA R, [Var + deslocamento]

b) Descreva as diferentes seções ou segmentos de um programa

section.text: seção de código, onde o programa é feito

section.data: seção onde se declaram as variáveis inicializadas

section.bss: seção de variáveis não inicializadas



c) Descreva os principais registradores da arquitetura intel, incluindo os registradores de flags

- RAX : Registrador conhecido como acumulador, geralmente os resultados vão para RAX
- RBX : Registrador que pode ser usado índice de base
- RCX : Registrador geralmente usado para contagem
- RDX : Registrador que serve de auxiliar em algumas operações de multiplicação/divisão
- RIP : Registrador contador de programa, guarda o endereço da linha de instrução
- RSI : Registrador de Índice e Origem
- RDI : Registrador de Índice e Destino
- RSP : Registrador que aponta para o topo da pilha de memória
- RBP : Registrador de base da pilha

OF : Flag overflow flag
DF : Flag de manipulação de dados
IF : Flag de interrupção
TF : Flag de busca (trap)
ZF : Flag para resultado zero
PF : Flag de resultado par
CF : Flag do carry

d) Qual registro que aponta para a próxima instrução a ser executada
RIP

Q9: Os números abaixo devem ser transformados em binário (b), octal (o), decimal (d) e hexa decimal (x). Todos para 16 bits com sinal

a) 123_{10} : $0000\ 0000\ 0111\ 1011_2 = 000153_8 = 007B_{16}$

b) 277_8 : $0000\ 0000\ 1011\ 1111_2 = 191_{10} = 00BF_{16}$

c) 1011111_2 : $137_8 = 95_{10} = 005F_{16}$

d) $AFAD_{16}$: $1010\ 1111\ 1010\ 1101_2 = 127655_8 = -20563_{10}$

0,7
Q

Q10: Converta os seguintes números octais em binários de 24 bits

a) 6740_8 : 000 000 000 000 110 111 100 000

b) 10021_8 : 000 000 000 001 000 000 010 001

c) 4764_8 : 000 000 000 000 100 111 110 100

d) 477_8 : 000 000 000 000 000 100 111 111