

UNIVERSIDADE ESTADUAL DE MATO GROSSO DO SUL – UEMS
CIÊNCIA DA COMPUTAÇÃO
PROF DR OSVALDO VARGAS JAQUES
1a Avaliação de Linguagem de Montagem

Enviar cópia digital da resposta manuscrita, em um único pdf, com o título P1NomeRGM.pdf, até as 18h30, horário MS, dia 24/06/2021, para ojacques@comp.uems.br.

1) Descreva um algoritmo que simule o modo como o computador executa programas e instruções.

```
PC = EnderecoInicial;  
ByteDeExecução = 1;  
Enquanto ByteDeExecução == 1 do  
    IR = Memoria[PC];  
    PC = PC+1;  
    DeterminaTipoInstrucao(IR, TipoDaInstrução);  
    AchaDado(TipoDaInstrução, IR, LocalizaçãoDado, DadoRequerido);  
    if DadoRequerido então  
        dado = Memoria[LocalizaçãoDado];  
    Execute(TipoDeInstrução, dado, memória, AC, PC, BitDeInstrução);
```

2) Se al for definido como 05_{16} e ah for definido como 07_{16} , qual é o conteúdo do registrador rax?
O valor de rax será 0705_{16}

3) Se o registro rax estiver definido como $81.985.529.216.486.895_{10}$ ($123456789ABCDEF_{16}$), qual é o conteúdo dos seguintes registradores em hexadecimal?

- a) al - EF
- b) ax - CDEF
- c) eax - 89ABCDEF
- d) rax - 123456789ABCDEF

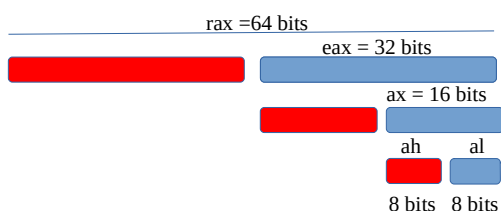
4) Na arquitetura Intel X86-64, quantos bytes podem ser armazenados em cada endereço?
Cada registrador padrão pode ler 64 bits por vez. Contudo, temos registradores maiores, alguns com 80 bits.

Cada segmento ou seção possui 64 kb = de $0x0000$ a $0xFFFF$ = 0 a 65535. Assim temos 65536 bytes / 1024 bytes = 64 kb

5) Dado o hexadecimal de 32 bits $004C4B40_{16}$, qual é o:
Vemos que existem 10 caracteres, mas para 32 bits são suficientes 8 caracteres.
Assim, desprezamos os primeiros '00'.

- a) Bit Menos Significativo (LSB) - $4C_{16} = 01001100_2$ logo o mais significativo é 0
- b) Bit Mais significativo (MSB) - $16_{16} = 00010100_2$ logo o mais significativo é 0.

5) Faça um desenho do layout do registrador acumulador.



6) Quantos bits cada um dos seguintes registradores representa:

- a) al - 8 bits
- b) rcx - 64 bits
- c) bx - 16 bits
- d) edx - 32 bits

8) Responda as seguintes questões:

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

Mov, move o conteúdo de um endereço se especificado entre colchetes, caso contrário pode carregar o endereço de um registrador ou variável, porém sem deslocamento.

Lea, carrega o endereço efetivo do segundo operando representado entre colchetes com deslocamento, sendo o deslocamento opcional.

mov rax, tabela1 ; endereço sem deslocamento

mov rax, [tabela1] ; primeiros 8 bytes de tabela1

lea rax,[tabela1] ; endereço sem deslocamento

mov rax,[tabela1 + rsi] ; erro

lea rax, [tabela1 + rsi] ; ok, endereço inicial de tabela1 + deslocamento em rsi
; em C seria &tabela1[rsi] ou &(tabela1+rsi)

b) Descreva as diferentes seções ou segmentos de um programa.

.text - onde está o código do programa

.data - onde estão os dados, variáveis, constantes, etc

.bss - dados não inicializados

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

AX - acumulador

BX - base

CX - loops, contagem

DX - auxiliar de acumulador

DI - índice de destino

SI - índice de origem (source)

IP - contador de programa

BP - ponteiro base de pilha

SP - topo de pilha

CS - seção ou segmento de código

DS - seção ou segmento de dados

ES - seção ou segmento extra

SS - seção ou segmento de pilha

sf - flag de sinal positivo ou negativo

pf - flag de paridade

of - flag de overflow

zf - flag de resultado zero

df - flag de direção

cf - flag de vai um

d) Qual registrador que aponta para a próxima instrução a ser executada?

O contador de programa ou ponteiro de instrução (IP)

9) Os números abaixo devem ser transformados em binários (b), octal(o), decimal (d) e hexadecimal (x). Todos para 16 bits com sinal

Pode-se usar o comando *echo "obase=2; ibase=10; 123" | bc*

Acima a entrada (obase) é em binário e saída (ibase) é em decimal para 123 em decimal.

Cuidado com números negativos. Melhor usar o método ortodoxo.

a) $123_{10} = 0000\ 0000\ 0111\ 1011_2 = 173_8 = 007B_{16}$

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

c) $101\ 1111_2 = 137_8 = 95_{10} = 005F_{16}$

d) $AFAD_{16} = 1010\ 1111\ 1010\ 1101_2 = 127655_8 = (-8+2).16^3 + 15 \times 16^2 + 10 \times 16^1 + 13 \times 16^0 = -20563$

10) Converta os seguintes números octais em binários de 24 bits

a) $6740_8 = 000\ 000\ 000\ 000\ 110\ 111\ 100\ 000_2$

b) $10021_8 = 000\ 000\ 000\ 001\ 000\ 000\ 010\ 001_2$

c) $4764_8 = 000\ 000\ 000\ 000\ 100\ 111\ 110\ 100_2$

d) $477_8 = 000\ 000\ 000\ 000\ 000\ 100\ 111\ 111_2$