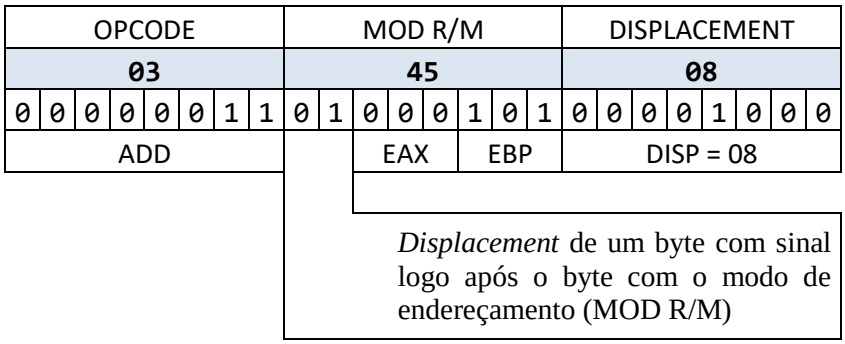
LISTA DE INTRODUÇÃO Á ARQUITETURA DE COMPUTADORES

Prof. *Jorge Habib Hanna El Khouri*
Aluno: _____ Turma: 2º UNIOESTE

1. Qual o formato das instruções da arquitetura Intel 80386 $32bits$? Descreva cada uma das partes.
2. Qual o tamanho em bytes da instrução mais curta? Cite algumas.
3. Qual o maior tamanho em bytes possível para uma instrução?
4. Quais os códigos binários de cada uma das seguintes instruções? Usar o item a como exemplo de como resolver os demais itens.

```
a. add eax, [ebp+8]
```

Solução: O código binário para a instrução $add\ eax, [ebp + 8]$ obtido é `03 45 08`. A interpretação para estes códigos é:




```
b.  mov ax, bx
c.  mov eax, ebx
d.  mov eax, dword ptr cs:[ebx+8*ecx+0x01020304]
e.  mov eax, dword ptr ss:[ebx+8*ecx+0x01020304]
f.  mov eax, dword ptr ds:[ebx+8*ecx+0x01020304]
g.  mov eax, dword ptr fs:[ebx+8*ecx+0x01020304]
h.  mov ebp, esp
i.  push ebp
j.  movzx eax, dl
k.  movsx edx, ah
l.  add eax, [ebx+8*edi+0x11223344]
m.  imul eax, dword ptr _x, 0xaabbccdd
n.  imul eax, dword ptr [ebx+8*ecx+0x01020304], 0xaabbccdd
o.  pop ebp
p.  push eax
q.  jnc 1f
r.  jmp 1f
s.  jmp _f1
t.  call _f1
u.  ret
v.  ret 16
w.  idiv ecx
x.  nop
y.  shl dword ptr _x, cl
z.  sub _x, eax
aa. cmp _x, eax
```

// considerar `1f` como um $label$
// considerar `f1` como uma $function$
// considerar $int\ x$;

5. Quais as instruções correspondentes a cada um dos seguintes códigos binários? Detalhar a solução de modo semelhante ao item a do exercício anterior.

   a. `90`
   b. `21 D8`
   c. `03 03`
   d. `13 84 FB 40 30 20 10`

6. Em relação à *chamada de função*, pede-se detalhar (completar a lista abaixo) todos os elementos que compõem a semântica desta abstração, e como são implementadas na linguagem C, compilador gcc para $Intel\ 32bits$.

   i. Reservar espaço para Parâmetros — $sub\ esp, \#params$
   ii. Copiar argumentos para a pilha — $mov\ [esp + disp], param$
   iii. *Completar os demais passos...*

7. Comente as diferenças entre os seguintes padrões para a semântica da chamada de função: $stdcall$, $cdecl$, $fastcall$ e $pascal$.

8. Elaborar um programa em C com a seguinte sequência de chamada de funções:

   $main(\,)$ **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;

   O programa pode dispor de outras funções para leitura de dados e apresentação de resultados, a fim de certificar o bom funcionamento da lógica. Depois de verificada a correção do programa, pede-se deixar apenas as funções diretamente envolvidas com a ordenação.

   Pede-se:

   a. Obter o $assembly$ do $main$, $ordenar$ e $shiftmaior$ em sintaxe $Intel\ 32bits$ gerado pelo compilador para cada uma das diretivas $\_stdcall$, $\_cdecl$, $\_fastcall$ (aplicáveis somente ao $ordenar$ e $shiftmaior$).

   b. Identificar todas as instruções responsáveis pela montagem dos $RA's$ das chamadas de $ordenar$ e $shitmaior$, apenas para o modo $cdecl$.

   c. Implementar a sua lógica em $assembly$ para as funções $ordenar$ e $shiftmaior$.

Considere o seguinte programa com uma função em C que recebe um parâmetro do tipo $struct$ e retorna como resultado um valor também do tipo $struct$. Pede-se gerar o $assembly$ para este código, e responder o que está sendo solicitado:

```
typedef struct {
    int a; int b; int c; int d;
} STRUCT;


STRUCT f(STRUCT r) {
STRUCT local;
  local = r;
  local.a++; local.b++;
  local.c++; local.d++;
  return local;
}

STRUCT global, result;

int main(int argc, char *argv[]) {
  result = f (global);
  return 0;
}
```

a. Explicar como foi realizada a passagem do parâmetro;
b. Explicar como foi realizado o retorno do resultado;

Pede-se agora adicionar um array de char na $struct$, como segue:

```
typedef struct {
    int a; int b; int c; int d;
    char s[100];
} STRUCT;
```

Para esta nova condição, pede-se:

a. Explicar como foi realizada a passagem do parâmetro;
b. Explicar como foi realizado o retorno do resultado;