Registradores de ponto flutuante XMM e vetores de ponto flutuante

Os registradores de vetor de pontos flutuantes xmm, que vão de xmm0 até xmm15, possuem 128 bits. Assim, podemos ter 4 floats, 2 doubles, 8 inteiros e assim por diante.

```
v1  dd 1.1, 2.2, 3.3, 4.4 ; 4 células float(32 bits)
v2  dd 5.5, 6.6, 7.7, 8.8
```

após as instruções:

```
movups xmm0, [v1]
movups xmm1, [v2]
```

se no gdb déssemos um comando como **`display $xmm0`** e depois **`display $xmm1`**, veríamos constantemente no gdb:

```
$xmm0 ={
        v4_float = {1.10000002, 2.20000005, 3.29999995, 4.4000001},
        v2_double = {3.6000008549541236,     921.60022034645078},
        v16_int8 = {-51, -52, -116, 63, -51, -52, 12, 64, 51, 51, 83, 64, -51, -52, -116, 64},
        v8_int16 = {-13107, 16268, -13107, 16396, 13107, 16467, -13107, 16524},
        v4_int32 = {1066192077, 1074580685, 1079194419, 1082969293},
        v2_int64 = {4615288900054469837,    4651317697086436147},
        uint128 = 85801667163669575356477574890839133389}

$xmm1 ={
        v4_float = {5.5, 6.5999999, 7.69999981, 8.80000019},
        v2_double = {19660.80082321167, 235929.65671997069},
        v16_int8 = {0, 0, -80, 64, 51, 51, -45, 64, 102, 102, -10, 64, -51, -52, 12, 65},
        v8_int16 = {0, 16560, 13107, 16595, 26214, 16630, -13107, 16652},
        v4_int32 = {1085276160, 1087583027, 1089889894, 1091357901},
        v2_int64 = {4671133533734961152, 4687346494116095590},
        uint128 = 86466281161759329983057081721229934592}
```

Embora vejamos xmm0 = [1.1, 2.2, 3.3, 4.4], devemos tratar como uma pilha, deste modo:

Índice em binário	11	10	01	00
Conteúdo	4.4	3.3	2.2	1.1

Ora, a sequência dos índices é vista como 11, 10, 01, 00, onde o primeiro índice é o topo.

Ao final, fazendo **`x/4fw &v3`** no dbg, teríamos para a variável v3 o seguinte conteúdo:

```
v3        = [30.7999992, 51.4800034,  77, 107.360008]
```

Ou conforme a tela:

```
(gdb) x/4fw &v3
0x404074:       30.7999992      51.4800034      77      107.360008
```

Uso da função C printf em pontos flutuantes

A função printf imprime a primeira célula de um double nos vetores xmm, lembrando que em um xmm cabem 2 doubles (qword) ou 4 floats (dword). No nosso exemplo, para xmm0, printf imprimiria o valor correspondente ao primeiro elemento de **v2_double** em:

```
$xmm0 ={
        v4_float = {1.10000002, 2.20000005, 3.29999995, 4.4000001},
        v2_double = {3.6000008549541236,      921.60022034645078},
        v16_int8 = {-51, -52, -116, 63, -51, -52, 12, 64, 51, 51, 83, 64, -51, -52, -116, 64},
        v8_int16 = {-13107, 16268, -13107, 16396, 13107, 16467, -13107, 16524},
        v4_int32 = {1066192077, 1074580685, 1079194419, 1082969293},
        v2_int64 = {4615288900054469837,    4651317697086436147},
        uint128 = 85801667163669575356477574890839133389}
```

Ou seja, imprimiria **3.6000008549541236.**

Exemplo, para o caso de xmm0:

```
...
fmt db "Resultado: %.2f",10,0
...

...
mov rdi, fmt ; carrega o formato
mov rax,1    ; informa que tem um registrador de ponto flutuante a
ser usado, só o xmm0
call printf
...
```

Caso queiramos imprimir os quatro valores float precisamos transformar cada float em double. Existem instruções para isso, como a **cvtss2sd** que converte **float** para **double**.
A função **printf** sempre lerá o primeiro valor **double** em **xmm**.