

指令说明

divsd、divss 指令表示浮点数除法。

divsd 指令为 64 位。divss 指令为 32 位。

divsd、divss 指令可以操作寄存器、内存。

语法格式 `divsd cc, dd` 表示 `dd / cc = dd`。

语法格式 `divss cc, dd` 表示 `dd / cc = dd`。

简单的除法

编写代码： `div.s`

```
.data

float64 :
    .double 888.66

float32 :
    .float 333.3

tmp32 :
    .float -3

str_64 :
    .string " float64 = %.5f  \n"

.text
.global main

main :
    pushq %rbp
    movq %rsp, %rbp

    # 除法。64 位
    movsd float64(%rip), %xmm0 # 64 位，浮点数，xmm0
    movq $-20, %r9             # 64 位，整数，r9
    cvtsi2sd %r9, %xmm7         # 64 位，整数转浮点数
    divsd %xmm7, %xmm0          # xmm0 = xmm0 / xmm7
    movq $str_64, %rdi
    callq printf

    # 除法。32 位
```

```
movss float32(%rip), %xmm0 # 32 位, 浮点数, xmm0
divss tmp32(%rip), %xmm0    # xmm0 = xmm0 / tmp32
movq $str_64, %rdi
cvtss2sd %xmm0, %xmm0      # 32 位浮点数转 64 位浮点数
callq printf

popq %rbp
retq
```

编译代码:

```
gcc div.s -o div
```

运行代码:

```
[root@local float]# ./div
float64 = -44.43300
float64 = -111.10000
```

分析结果:

汇编代码	结果和分析
<div># 除法。64 位</div> <div>movsd float64(%rip), %xmm0 # 64 位, 浮点数, xmm0</div> <div>movq \$-20, %r9 # 64 位, 整数, r9</div> <div>cvtss2sd %r9, %xmm7 # 64 位, 整数转浮点数</div> <div>divsd %xmm7, %xmm0 # xmm0 = xmm0 / xmm7</div>	<div>float64 = -44.43300</div> <div>把 64 位浮点数变量 float64 的值 888.66 写到 xmm0 。</div> <div>把 64 位整数-20 写到 r9, 再转为浮点数写到 xmm7 。</div> <div>浮点数除法, 寄存器 xmm0 除以寄存器 xmm7, 结果写到 xmm0。</div> <div>888.66 / (-20) = -44.43300</div>
<div># 除法。32 位</div> <div>movss float32(%rip), %xmm0 # 32 位, 浮点数, xmm0</div> <div>divss tmp32(%rip), %xmm0 # xmm0 = xmm0 / tmp32</div>	<div>float64 = -111.10000</div> <div>把 32 位浮点数变量 float32 的值 333.3 写到 xmm0 。</div> <div>浮点数除法, 寄存器 xmm0 除以内存变量 tmp32, 结果写到 xmm0。</div> <div>333.3 / (-3) = -111.10000</div>