

指令说明

subsd、subss 指令表示浮点数减法。

subsd 指令为 64 位。subss 指令为 32 位。

subsd、subss 指令可以操作寄存器、内存。

语法格式 `subsd ee, ff` 表示 `ff = ff - ee`。

语法格式 `subss ee, ff` 表示 `ff = ff - ee`。

简单的减法

编写代码： `sub.s`

```
.data

float64 :
    .double 88888.888

float32 :
    .float 55555.55

tmp32 :
    .float 77777.77

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

.text
.global main

main :
    pushq %rbp
    movq %rsp, %rbp

    # 减法。64 位
    movsd float64(%rip), %xmm0 # 64 位，浮点数，xmm0
    movq $333, %r9              # 64 位，整数，r9
    cvtsi2sd %r9, %xmm7         # 64 位，整数转浮点数
    subsd %xmm7, %xmm0          # xmm0 = xmm0 - xmm7
    movq $str_64, %rdi
    callq printf

    # 减法。32 位
```

```
movss float32(%rip), %xmm0 # 32 位, 浮点数, xmm0
subss 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 sub.s -o sub
```

运行代码:

```
[root@local float]# ./sub
float64 = 88555.88800
float64 = -22222.22266
```

分析结果:

汇编代码	结果和分析
<div># 减法. 64 位</div> <div>movsd float64(%rip), %xmm0 # 64 位, 浮点数, xmm0</div> <div>movq \$333, %r9 # 64 位, 整数, r9</div> <div>cvtssi2sd %r9, %xmm7 # 64 位, 整数转浮点数</div> <div>subsd %xmm7, %xmm0 # xmm0 = xmm0 - xmm7</div>	<div>float64 = 88555.88800</div> <div>把 64 位浮点数变量 float64 的值 88888.888 写到 xmm0。</div> <div>把 64 位整数 333 写到 r9, 再转为浮点数写到 xmm7。</div> <div>浮点数减法, 寄存器 xmm0 减去寄存器 xmm7, 结果写到 xmm0。</div> <div>88888.888 - 333 = 88555.88800</div>
<div># 减法. 32 位</div> <div>movss float32(%rip), %xmm0 # 32 位, 浮点数, xmm0</div> <div>subss tmp32(%rip), %xmm0 # xmm0 = xmm0 - tmp32</div>	<div>float64 = -22222.22266</div> <div>把 32 位浮点数变量 float32 的值 55555.55 写到 xmm0。</div> <div>浮点数减法, 寄存器 xmm0 减去内存变量 tmp32, 结果写到 xmm0。</div> <div>55555.555 - 77777.77 = -22222.22266</div>

浮点数减法, 不能直接操作立即数。

可以把浮点数型的立即数, 写到浮点数变量, 然后使用浮点数减法。比如, `subss tmp32(%rip), %xmm0`。

可以把整数型的立即数, 写到整数寄存器, 然后把整数转为浮点数, 最后使用浮点数减法。比如, `movq $333, %r9` 和 `cvtssi2sd %r9, %xmm7`。