

指令说明

addsd、addss 指令表示浮点数加法。

addsd 指令为 64 位。addss 指令为 32 位。

addsd、addss 指令可以操作寄存器、内存。

语法格式 `addsd ee, ff` 表示 `ff = ff + ee`。

语法格式 `addss ee, ff` 表示 `ff = ff + ee`。

简单的加法

编写代码： `add.s`

```
.data

float64 :
    .double 11111.222

float32 :
    .float 55555.555

tmp32 :
    .float -333.33

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

.text
.global main

main :
    pushq %rbp
    movq %rsp, %rbp

    # 加法。64 位
    movsd float64(%rip), %xmm0 # 64 位，浮点数，xmm0
    movq $33, %r9               # 64 位，整数，r9
    cvtsi2sd %r9, %xmm7         # 64 位，整数转浮点数
    addsd %xmm7, %xmm0          # xmm0 = xmm0 + xmm7
    movq $str_64, %rdi
    callq printf

    # 加法。32 位
```

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

运行代码:

```
[root@local float]# ./add
float64 = 11144.22200
float64 = 55222.22656
```

分析结果:

汇编代码	结果和分析
<div># 加法。64 位</div> <div>movsd float64(%rip), %xmm0 # 64 位, 浮点数, xmm0</div> <div>movq \$33, %r9 # 64 位, 整数, r9</div> <div>cvtsi2sd %r9, %xmm7 # 64 位, 整数转浮点数</div> <div>addsd %xmm7, %xmm0 # xmm0 = xmm0 + xmm7</div>	<div>float64 = 11144.22200</div> <div>把变量 float64 的值 11111.222 写到 xmm0。</div> <div>把整数 33 写到 r9, 再转为浮点数写到 xmm7。</div> <div>浮点数加法, 2 个浮点数寄存器相加。</div> <div>结果写到 xmm0。</div> <div>11111.222 + 33 = 11144.22200</div>
<div># 加法。32 位</div> <div>movss float32(%rip), %xmm0 # 32 位, 浮点数, xmm0</div> <div>addss tmp32(%rip), %xmm0 # xmm0 = xmm0 + tmp32</div>	<div>float64 = 55222.22656</div> <div>把变量 float32 的值 55555.555 写到 xmm0。</div> <div>浮点数加法, 浮点数寄存器加上内存变量。</div> <div>结果写到 xmm0。</div> <div>55555.555 + (-333.33) = 55222.22656</div>

浮点数加法, 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, 然后使用浮点数加法。比如, `addss 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, 最后使用浮点数加法。比如, `movq $33, %r9` 和 `cvtsi2sd %r9, %xmm7`。

浮点数操作, 可能有精度问题。

上例 32 位浮点数加法 $55555.555 + (-333.33) = 55222.22656$, 准确结果应该为 55222.225000。

写 2 个方法, 作为对比。

```
void func_add(float aa, float bb)
{
    float cc = aa + bb;
    int dd = (int)(cc * 10000);
    printf("add float = %f %d \n", cc, dd);
}

void func_add2(double aa, double bb)
```

```
{  
    double cc = aa + bb;  
    printf("add double = %f \n", cc);  
}
```

分别调用 2 个方法。

```
func_add(55555.555, -333.33);  
// add float = 55222.226562 552222272  
  
func_add2(55555.555, -333.33);  
// add double = 55222.225000
```

入参相同，结果不相同。

func_add 的结果为 55222.226562。

func_add2 的结果为 55222.225000。