

指令说明

浮点数转换分为：32 位浮点数和 64 位浮点数相互转换，32 位整数和 32 位浮点数相互转换，64 位整数和 64 位浮点数相互转换。

cvtss2sd 指令，表示 32 位浮点数转 64 位浮点数。比如，`cvtss2sd %xmm0, %xmm1`。

cvtsd2ss 指令，表示 64 位浮点数转 32 位浮点数。比如，`cvtsd2ss %xmm0, %xmm1`。

cvtsi2sd 指令，表示 64 位整数转 64 位浮点数。比如，`cvtsi2sd %r8, %xmm0`。

cvtsi2ss 指令，表示 32 位整数转 32 位浮点数。比如，`cvtsi2ss %r8d, %xmm0`。

cvttsd2si 指令，表示 64 位浮点数转 64 位整数，没有四舍五入。比如，`cvttsd2si %xmm0, %r8`。

cvtsd2si 指令，表示 64 位浮点数转 64 位整数，有四舍五入。比如，`cvtsd2si %xmm0, %r8`。

cvtss2si 指令，表示 32 位浮点数转 32 位整数，没有四舍五入。比如，`cvtss2si %xmm0, %r8d`。

cvtss2si 指令，表示 32 位浮点数转 32 位整数，有四舍五入。比如，`cvtss2si %xmm0, %r8d`。

用汇编代码分析

编写代码：float_cvt.s

```
.data

float32 :
    .float -111.777

float64 :
    .double 555.666

str_32_to_64 :
    .string " float32 = %.3f    float64 = %.3f \n"

str_64_to_32 :
    .string " float64 = %.3f    float32 = %.3f \n"

str_i2f_64 :
    .string " int64 = %lld    float64 = %.3f \n"

str_f2i_64 :
    .string " float64 = %.3f    int64 = %lld \n"

.text
.global main

main :
    pushq %rbp
```

```

movq %rsp, %rbp

# 浮点数, 32 位转 64 位
movss float32(%rip), %xmm0 # 浮点数, 32 位
cvtss2sd %xmm0, %xmm1      # 浮点数, 32 位转 64 位
movq $str_32_to_64, %rdi
cvtss2sd %xmm0, %xmm0      # %f 表示 64 位浮点数
callq printf

# 浮点数, 64 位转 32 位
movsd float64(%rip), %xmm0 # 浮点数, 64 位
cvtsd2ss %xmm0, %xmm1      # 浮点数, 64 位转 32 位
movq $str_64_to_32, %rdi
cvtss2sd %xmm1, %xmm1      # %f 表示 64 位浮点数
callq printf

# 整数转浮点数, 64 位
movq $77777, %r8           # 整数, 64 位
cvtsi2sd %r8, %xmm0        # 整数转浮点数, 64 位
movq $str_i2f_64, %rdi
movq %r8, %rsi
callq printf

# 浮点数转整数, 64 位。没有四舍五入
movsd float64(%rip), %xmm0 # 浮点数, 64 位
cvttsd2si %xmm0, %r8       # 浮点数转整数, 64 位
movq $str_f2i_64, %rdi
movq %r8, %rsi
callq printf

# 浮点数转整数, 64 位。四舍五入
movsd float64(%rip), %xmm0 # 浮点数, 64 位
cvtsd2si %xmm0, %r8       # 浮点数转整数, 64 位
movq $str_f2i_64, %rdi
movq %r8, %rsi
callq printf

# 整数转浮点数, 32 位
movl $-8888, %r8d          # 整数, 32 位
cvtsi2ss %r8d, %xmm0       # 整数转浮点数, 32 位
movq $str_i2f_64, %rdi
movslq %r8d, %r8           # 符号扩展, 32 位转 64 位
movq %r8, %rsi
cvtss2sd %xmm0, %xmm0      # 浮点数, 32 位转 64 位
callq printf

# 浮点数转整数, 32 位。没有四舍五入
movss float32(%rip), %xmm0 # 浮点数, 32 位
cvtts2si %xmm0, %r8d       # 浮点数转整数, 32 位

```

```

movq $str_f2i_64, %rdi
cvtss2sd %xmm0, %xmm0    # 浮点数, 32 位转 64 位
movslq %r8d, %r8         # 符号扩展, 32 位转 64 位
movq %r8, %rsi
callq printf

# 浮点数转整数, 32 位。四舍五入
movss float32(%rip), %xmm0 # 浮点数, 32 位
cvtss2si %xmm0, %r8d       # 浮点数转整数, 32 位
movq $str_f2i_64, %rdi
cvtss2sd %xmm0, %xmm0    # 浮点数, 32 位转 64 位
movslq %r8d, %r8         # 符号扩展, 32 位转 64 位
movq %r8, %rsi
callq printf

popq %rbp
retq

```

编译代码:

```
gcc float_cvt.s -o float_cvt
```

运行代码:

```

[root@local cvt]# ./float_cvt
float32 = -111.777    float64 = -111.777
float64 = 555.666    float32 = 555.666
int64 = 77777       float64 = 77777.000
float64 = 555.666    int64 = 555
float64 = 555.666    int64 = 556
int64 = -8888       float64 = -8888.000
float64 = -111.777   int64 = -111
float64 = -111.777   int64 = -112

```

分析结果:

汇编代码	结果和分析
# 浮点数, 32 位转 64 位 movss float32(%rip), %xmm0 # 浮点数, 32 位 cvtss2sd %xmm0, %xmm1 # 浮点数, 32 位转 64 位	float32 = -111.777 float64 = -111.777 cvtss2sd 指令, 表示 32 位浮点数转 64 位浮点数。
# 浮点数, 64 位转 32 位 movsd float64(%rip), %xmm0 # 浮点数, 64 位 cvtsd2ss %xmm0, %xmm1 # 浮点数, 64 位转 32 位	float64 = 555.666 float32 = 555.666 cvtsd2ss 指令, 表示 64 位浮点数转 32 位浮点数。
# 整数转浮点数, 64 位 movq \$77777, %r8 # 整数, 64 位 cvtsi2sd %r8, %xmm0 # 整数转浮点数, 64 位	int64 = 77777 float64 = 77777.000 cvtsi2sd 指令, 表示 64 位整数转 64 位浮点数。
# 浮点数转整数, 64 位。没有四舍五入 movsd float64(%rip), %xmm0 # 浮点数, 64 位 cvttsd2si %xmm0, %r8 # 浮点数转整数, 64 位	float64 = 555.666 int64 = 555 cvttsd2si 指令, 表示 64 位浮点数转 64 位整数, 没有四舍五入。

# 浮点数转整数，64 位。四舍五入 movsd float64(%rip), %xmm0 # 浮点数，64 位 cvttsd2si %xmm0, %r8 # 浮点数转整数，64 位	float64 = 555.666 int64 = 556 cvttsd2si 指令，表示 64 位浮点数转 64 位整数，有四舍五入。
# 整数转浮点数，32 位 movl \$-8888, %r8d # 整数，32 位 cvtsi2ss %r8d, %xmm0 # 整数转浮点数，32 位	int64 = -8888 float64 = -8888.000 cvtsi2ss 指令，表示 32 位整数转 32 位浮点数。
# 浮点数转整数，32 位 。没有四舍五入 movss float32(%rip), %xmm0 # 浮点数，32 位 cvtss2si %xmm0, %r8d # 浮点数转整数，32 位	float64 = -111.777 int64 = -111 cvtss2si 指令，表示 32 位浮点数转 32 位整数，没有四舍五入。
# 浮点数转整数，32 位 。四舍五入 movss float32(%rip), %xmm0 # 浮点数，32 位 cvtss2si %xmm0, %r8d # 浮点数转整数，32 位	float64 = -111.777 int64 = -112 cvtss2si 指令，表示 32 位浮点数转 32 位整数，有四舍五入。