作者：代兴　邮箱：503268771@qq.com　Vx 公众号：东方架构师

指令说明

右移指令，包括逻辑右移指令、算术右移指令。

shr 指令表示逻辑右移指令。
shr 指令分为 8 位 shrb、16 位 shrw、32 位 shrl、64 位 shrq。
shr 指令可以操作寄存器、内存。

sar 指令表示算术右移指令。
sar 指令分为 8 位 sarb、16 位 sarw、32 位 sarl、64 位 sarq。
sar 指令可以操作寄存器、内存。

逻辑右移指令，左侧补 0。
算术右移指令，左侧补符号位。

用汇编代码分析

编写代码：right.s

```
.data

num32:
    .long 0x0

num64 :
    .quad 0x0

str_num32 :
    .string "  int32 = %#19X   \n"

str_num64 :
    .string "  int64 = %#1911X   \n"

str_num32_cmp :
    .string "  int32   shr = %#10X   sar = %#10X \n"

.text
.global main

main :
    pushq %rbp
    movq %rsp, %rbp
    subq $64 , %rsp
```

```
# 32 位。操作寄存器
movl $0x88888888, %r8d
shrl $8 , %r8d          # 逻辑右移
movq $str_num32 , %rdi
movl %r8d , %esi
callq printf

# 32 位。操作栈
movl $0x77777777, -4(%rbp)
shrl $8 , -4(%rbp)      # 逻辑右移
movq $str_num32 , %rdi
movl -4(%rbp)  , %esi
callq printf

# 32 位。操作数据段
movl $0x66666666, num32(%rip)
shrl $8 , num32(%rip)   # 逻辑右移
movq $str_num32 , %rdi
movl num32(%rip) , %esi
callq printf

# 64 位。移动 32 位。操作寄存器
movq $0x3333333355555555, %rcx
shrl $8 , %ecx          # 逻辑右移。高位清 0
movq $str_num64 , %rdi
movq %rcx , %rsi
callq printf

# 64 位。移动 32 位。操作数据段
movq $0x3333333355555555, %rcx
movq %rcx , num64(%rip)
shrl $8 , num64(%rip)   # 逻辑右移。
movq $str_num64 , %rdi
movq num64(%rip) , %rsi
callq printf

# 32 位。操作寄存器。左侧高位为 0
movl $0x33333333, %ebx
shrl $8 , %ebx          # 逻辑右移。左侧补 0
movl $0x33333333, %r8d
sarl $8 , %r8d          # 算术右移。左侧补符号位
movq $str_num32_cmp , %rdi
movl %ebx , %esi
movl %r8d , %edx
callq printf

# 32 位。操作寄存器。左侧高位为 1
movl $0x88888888, %ebx
shrl $8 , %ebx          # 逻辑右移。左侧补 0
```

```
    movl $0x88888888, %r8d
    sarl $8 , %r8d          # 算术右移。左侧补符号位
    movq $str_num32_cmp , %rdi
    movl %ebx , %esi
    movl %r8d , %edx
    callq printf

    # 16 位。操作寄存器
    movl $0x55667788, %ebx
    shrw $4, %bx            # 逻辑右移。
    movq $str_num32, %rdi
    movl %ebx, %esi
    callq printf

    # 8 位。操作寄存器
    movl $0x55667788, %ebx
    shrb $4, %bl            # 逻辑右移。
    movq $str_num32, %rdi
    movl %ebx, %esi
    callq printf

    addq $64 , %rsp
    popq %rbp
    retq
```

编译代码：

```
gcc right.s -o right
```

运行代码：

```
[root@local shift]# ./right
  int32 =            0X888888
  int32 =            0X777777
  int32 =            0X666666
  int64 =            0X555555
  int64 =  0X3333333300555555
  int32   shr =   0X333333   sar =   0X333333
  int32   shr =   0X888888   sar = 0XFF888888
  int32 =          0X55660778
  int32 =          0X55667708
```

分析结果：

汇编代码	结果和分析
# 32 位。操作寄存器 movl $0x88888888, %r8d shrl $8 , %r8d # 逻辑右移	int32 = 0X888888 shrl 指令，逻辑右移。 操作寄存器 r8d。 0x88888888 逻辑右移 8 位，结果为 0X888888。
# 32 位。操作栈 movl $0x77777777, -4(%rbp)	int32 = 0X777777

shrl $8 , -4(%rbp)　　# 逻辑右移	shrl 指令，逻辑右移。 操作栈内存-4(%rbp)。 0x77777777 逻辑右移 8 位，结果为 0X777777。
# 32 位。操作数据段 movl $0x66666666, num32(%rip) shrl $8 , num32(%rip)　# 逻辑右移	int32 =　　0X666666 shrl 指令，逻辑右移。 操作数据段内存 num32。 0x66666666 逻辑右移 8 位，结果为 0X666666。
# 64 位。移动 32 位。操作寄存器 movq $0x3333333355555555, %rcx shrl $8 , %ecx　　# 逻辑右移。高位清 0	int64 =　　0X555555 shrl 指令，逻辑右移。 操作寄存器 ecx。 shrl 操作通用寄存器，触发高位清 0。 0x3333333355555555 的低 32 位逻辑右移 8 位，结果为 0X555555。
# 64 位。移动 32 位。操作数据段 movq $0x3333333355555555, %rcx movq %rcx , num64(%rip) shrl $8 , num64(%rip)　# 逻辑右移。	int64 =　0X3333333300555555 shrl 指令，逻辑右移。 操作数据段 num64。操作低 32 位。 0x3333333355555555 的低 32 位逻辑右移 8 位，结果为 0X3333333300555555。 高 32 位，没有变化，都为 33333333。 低 32 位，发生变化，从 55555555 变为 00555555。

# 32 位。操作寄存器。左侧高位为 0 	
movl $0x33333333, %ebx
shrl $8 , %ebx　　# 逻辑右移。左侧补 0
movl $0x33333333, %r8d
sarl $8 , %r8d　　# 算术右移。左侧补符号位
	int32　shr =　0X333333　sar =　0X333333
	
	shrl 指令，逻辑右移。
	sarl 指令，算术右移。
	左侧高位为 0。
	0x33333333 逻辑右移 8 位、算术右移 8 位，结果相同，都为 0X333333。

# 32 位。操作寄存器。左侧高位为 1 	
movl $0x88888888, %ebx
shrl $8 , %ebx　　# 逻辑右移。左侧补 0
movl $0x88888888, %r8d
sarl $8 , %r8d　　# 算术右移。左侧补符号位
	int32　shr =　0X888888　sar = 0XFF888888
	
	shrl 指令，逻辑右移。
	sarl 指令，算术右移。
	左侧高位为 1。
	逻辑右移。左侧补 0。结果为 0X888888。
	算术右移。左侧补符号位。结果为 0XFF888888。

# 16 位。操作寄存器 movl $0x55667788, %ebx shrw $4, %bx　　# 逻辑右移。	int32 =　　0X55660778 shrw 指令，逻辑右移。 0x55667788 的低 16 位逻辑右移 4 位，结果为 0X55660778。 高 16 位，没有变化，都为 5566。 低 16 位，发生变化，从 7788 变为 0778。
# 8 位。操作寄存器 movl $0x55667788, %ebx shrb $4, %bl　　# 逻辑右移。	int32 =　　0X55667708 shrb 指令，逻辑右移。 0x55667788 的低 8 位逻辑右移 4 位，结果为 0X55667708。 高 24 位，没有变化，都为 556677。 低 8 位，发生变化，从 88 变为 08。

64 位 0x3333333355555555，shrl 逻辑右移 8 位，操作寄存器、操作内存，结果不同。
操作寄存器，结果为 0X555555。操作内存，结果为 0X3333333300555555。
shrl 是 32 位指令，操作通用寄存器，触发 64 位寄存器高位清零。

右移指令，操作对应的位数。
比如，32 位 0x55667788，写到 32 位寄存器 ebx。 `shrb $4, %bl` 操作低 8 位，结果为 0X55667708。
高 24 位，没有变化，都为 556677。低 8 位，发生变化，从 88 变为 08。