

指令说明

左移指令，包括逻辑左移指令、算术左移指令。

shl 指令表示逻辑左移指令。

shl 指令分为 8 位 shlb、16 位 shlw、32 位 shll、64 位 shlq。

shl 指令可以操作寄存器、内存。

sal 指令表示算术左移指令。

sal 指令分为 8 位 salb、16 位 salw、32 位 sall、64 位 salq。

sal 指令可以操作寄存器、内存。

逻辑左移指令，右侧补 0。

算术左移指令，右侧补 0。

用汇编代码分析

编写代码：left.s

```
.data

num32:
    .long 0x0

num64 :
    .quad 0x0

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

str_num64 :
    .string "  int64 = %#llx  \n"

str_num32_cmp :    # 对比
    .string "  int32  shl = %#X   sal = %#X \n"

.text
.global main

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

```

# 32 位。操作寄存器
movl $0x88888888, %r8d
shll $8 , %r8d          # 逻辑左移
movq $str_num32 , %rdi
movl %r8d , %esi
callq printf

# 32 位。操作栈内存
movl $0x77777777, -4(%rbp)
shll $8 , -4(%rbp)      # 逻辑左移
movq $str_num32 , %rdi
movl -4(%rbp) , %esi
callq printf

# 32 位。操作数据段内存
movl $0x66666666, num32(%rip)
shll $8 , num32(%rip)   # 逻辑左移
movq $str_num32 , %rdi
movl num32(%rip) , %esi
callq printf

# 64 位。移动 32 位。操作寄存器
movq $0x3333333355555555, %rcx
shll $8 , %ecx          # 逻辑左移。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)
shll $8 , num64(%rip)   # 逻辑左移。
movq $str_num64 , %rdi
movq num64(%rip), %rsi
callq printf

# 32 位。操作寄存器。右侧低位为 1
movl $0x11111111, %ebx
shll $8 , %ebx          # 逻辑左移
movl $0x11111111, %r8d
sall $8 , %r8d          # 算术左移
movq $str_num32_cmp , %rdi
movl %ebx , %esi
movl %r8d , %edx
callq printf

# 64 位。操作寄存器
movq $0x2222222233333333, %rcx
shlq $16 , %rcx         # 逻辑左移。

```

```
movq $str_num64 , %rdi
movq %rcx , %rsi
callq printf

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

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

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

编译代码:

```
gcc left.s -o left
```

运行代码:

```
[root@local shift]# ./left
int32 = 0X88888800
int32 = 0X77777700
int32 = 0X66666600
int64 = 0X55555500
int64 = 0X3333333355555500
int32  shl = 0X11111100    sal = 0X11111100
int64 = 0X2222333333330000
int32 = 0X55667880
int32 = 0X55667780
```

分析结果:

汇编代码	结果和分析
# 32 位。操作寄存器 movl \$0x88888888, %r8d shll \$8 , %r8d # 逻辑左移	int32 = 0X88888800 shll 指令，逻辑左移。 操作寄存器 r8d。 0x88888888 逻辑左移 8 位，结果为 0X88888800。
# 32 位。操作栈内存 movl \$0x77777777, -4(%rbp) shll \$8 , -4(%rbp) # 逻辑左移	int32 = 0X77777700 shll 指令，逻辑左移。 操作栈内存-4(%rbp)。 0x77777777 逻辑左移 8 位，结果为 0X77777700。

# 32 位。操作数据段内存 movl \$0x66666666, num32(%rip) shll \$8 , num32(%rip) # 逻辑左移	int32 = 0X66666600 shll 指令，逻辑左移。 操作数据段内存 num32。 0x66666666 逻辑左移 8 位，结果为 0X66666600。
# 64 位。移动 32 位。操作寄存器 movq \$0x3333333555555555, %rcx shll \$8 , %ecx # 逻辑左移。高位清 0	int64 = 0X55555500 shll 指令，逻辑左移。 操作寄存器 ecx。 shll 操作通用寄存器，触发高位清 0。 0x3333333555555555 的低 32 位逻辑左移 8 位，结果为 0X55555500。
# 64 位。移动 32 位。操作数据段 movq \$0x3333333555555555, %rcx movq %rcx , num64(%rip) shll \$8 , num64(%rip) # 逻辑左移。	int64 = 0X3333333555555500 shll 指令，逻辑左移。 操作数据段 num64。操作低 32 位。 0x3333333555555555 的低 32 位逻辑左移 8 位，结果为 0X3333333555555500。 高 32 位，没有变化，都为 33333333。 低 32 位，发生变化，从 55555555 变为 55555500。
# 32 位。操作寄存器。右侧低位为 1 movl \$0x11111111, %ebx shll \$8 , %ebx # 逻辑左移 movl \$0x11111111, %r8d sall \$8 , %r8d # 算术左移	int32 shl = 0X11111100 sal = 0X11111100 shll 指令，逻辑左移。 sall 指令，算术左移。 0x11111111 逻辑左移 8 位、算术左移 8 位，结果相同，都为 0X11111100。
# 64 位。操作寄存器 movq \$0x2222222333333333, %rcx shlq \$16 , %rcx # 逻辑左移。	int64 = 0X2222333333330000 shlq 指令，逻辑左移。操作 64 位。 0x2222222333333333 逻辑左移 16 位，结果为 0X2222333333330000。
# 16 位。操作寄存器 movl \$0x55667788, %ebx shlw \$4, %bx # 逻辑左移。	int32 = 0X55667780 shlw 指令，逻辑左移。 0x55667788 的低 16 位逻辑左移 4 位，结果为 0X55667780。 高 16 位，没有变化，都为 5566。 低 16 位，发生变化，从 7788 变为 7780。
# 8 位。操作寄存器 movl \$0x55667788, %ebx shlb \$4, %bl # 逻辑左移。	int32 = 0X55667780 shlb 指令，逻辑左移。 0x55667788 的低 8 位逻辑左移 4 位，结果为 0X55667780。 高 24 位，没有变化，都为 556677。 低 8 位，发生变化，从 88 变为 80。

64 位 0x3333333555555555，shll 逻辑左移 8 位，操作寄存器、操作内存，结果不同。

操作寄存器，结果为 0X55555500。

操作内存，结果为 0X3333333555555500。

shll 是 32 位指令，操作通用寄存器，触发 64 位寄存器高位清零。

左移指令，操作对应的位数。

比如，32 位 0x55667788，写到 32 位寄存器 ebx。

`shlb $4, %b1` 操作低 8 位，结果为 0X55667780。

高 24 位，没有变化，都为 556677。

低 8 位，发生变化，从 88 变为 80。