

## 指令说明

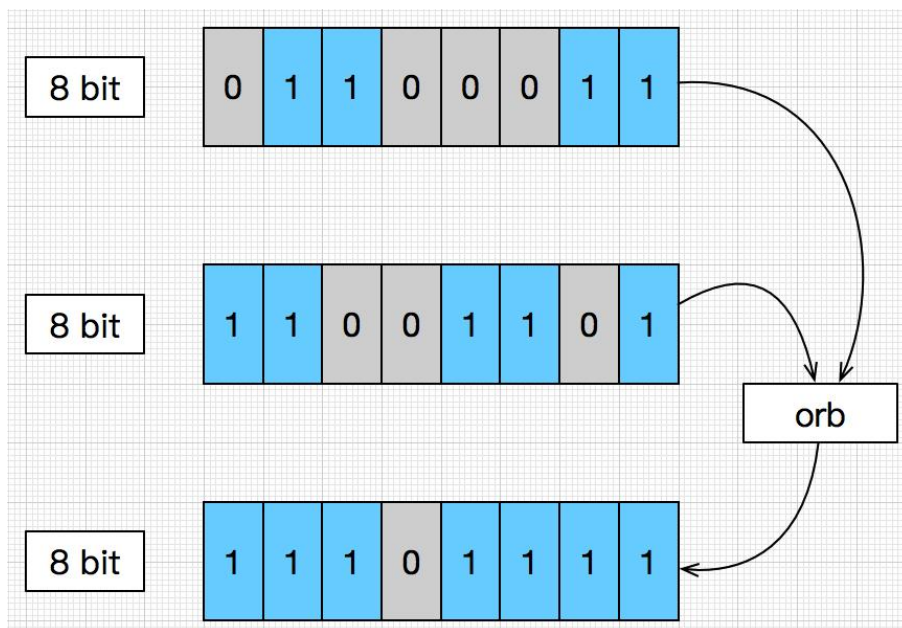
or 指令表示位或。

or 指令分为 8 位 orb、16 位 orw、32 位 orl、64 位 orq。

or 指令可以操作立即数、寄存器、内存。

语法格式 `or bb, aa` 表示 `aa = aa | bb`。

对应的位执行位或操作。如果 2 个位有 1 个为 1 或都为 1，则结果为 1，否则结果为 0。



## 用汇编代码分析

编写代码：or\_bit.s

```
.data

str64 :
    .string " int64  %#llx \n"

str32 :
    .string " int32  %#X \n"

.text

.global main

main :
    pushq %rbp
    movq %rsp, %rbp
```

```

subq $64, %rsp

# 64 位。操作寄存器
movq $0x1010101010101010, %rax # 64 位, 寄存器
movq $0x0101010101010101, %rbx # 64 位, 寄存器
orq  %rbx, %rax                # 64 位, 位或
movq $str64, %rdi
movq %rax, %rsi
callq printf

# 32 位。操作寄存器
movl $0x0000AAAA, %ecx # 32 位, 寄存器
movl $0xBBBB0000, %edx # 32 位, 寄存器
orl  %edx, %ecx        # 32 位, 位或
movq $str32, %rdi
movl %ecx, %esi
callq printf

# 32 位。操作栈内存
movl $0x0000AAAA, -8(%rbp) # 32 位, 栈
orl  $0xBBBB0000, -8(%rbp) # 32 位, 位或
movq $str32, %rdi
movl -8(%rbp), %esi
callq printf

# 16 位。操作寄存器
movl $0xAAAAAAAA, %ebx # 32 位, 占位
movw $0xCCCC, %bx      # 16 位, 寄存器
orw  $0xFF00, %bx       # 16 位, 位或
movq $str32, %rdi
movl %ebx, %esi
callq printf

# 8 位。操作寄存器
movl $0xAAAAAAAA, %ebx # 32 位, 占位
movb $0xCC, %bl        # 8 位, 寄存器
orb  $0xF0, %bl        # 8 位, 位或
movq $str32, %rdi
movl %ebx, %esi
callq printf

addq $64, %rsp
popq %rbp
retq

```

编译代码:

```
gcc or_bit.s -o or_bit
```

运行代码:

```
[root@local binary]# ./or_bit
int64  0X1111111111111111
int32  0XBBBBAAAA
int32  0XBBBBAAAA
int32  0XAAAAFFCC
int32  0XAAAAAFC
```

分析结果：

汇编代码	结果和分析
<div># 64 位。操作寄存器</div> <div>movq \$0x1010101010101010, %rax # 64 位，寄存器</div> <div>movq \$0x0101010101010101, %rbx # 64 位，寄存器</div> <div>orq %rbx, %rax # 64 位，位或</div>	<div>int64  0X1111111111111111</div> <div>操作 64 位寄存器 rbx、rax。</div> <div>0x1010101010101010   0x0101010101010101 = 0X1111111111111111</div>
<div># 32 位。操作寄存器</div> <div>movl \$0x0000AAAA, %ecx # 32 位，寄存器</div> <div>movl \$0xBBBB0000, %edx # 32 位，寄存器</div> <div>orl %edx, %ecx # 32 位，位或</div>	<div>int32  0XBBBBAAAA</div> <div>操作 32 位寄存器 edx、ecx。</div> <div>0x0000AAAA   0xBBBB0000 = 0XBBBBAAAA</div> <div>0000 与 BBBB 执行位或，等于 BBBB。</div> <div>AAAA 与 0000 执行位或，等于 AAAA。</div>
<div># 32 位。操作栈内存</div> <div>movl \$0x0000AAAA, -8(%rbp) # 32 位，栈</div> <div>orl \$0xBBBB0000, -8(%rbp) # 32 位，位或</div>	<div>int32  0XBBBBAAAA</div> <div>操作栈内存。</div> <div>0x0000AAAA   0xBBBB0000 = 0XBBBBAAAA</div>
<div># 16 位。操作寄存器</div> <div>movl \$0xAAAAAAAA, %ebx # 32 位，占位</div> <div>movw \$0xCCCC, %bx # 16 位，寄存器</div> <div>orw \$0xFF00, %bx # 16 位，位或</div>	<div>int32  0XAAAAFFCC</div> <div>操作 16 位寄存器 bx。</div> <div>高 16 位，没有变化，都为 AAAA。</div> <div>低 16 位，发生变化，从 CCCC 变成 FFCC。</div> <div>CCCC   FF00 = FFCC</div>
<div># 8 位。操作寄存器</div> <div>movl \$0xAAAAAAAA, %ebx # 32 位，占位</div> <div>movb \$0xCC, %bl # 8 位，寄存器</div> <div>orb \$0xF0, %bl # 8 位，位或</div>	<div>int32  0XAAAAAFC</div> <div>操作 8 位寄存器 bl。</div> <div>高 24 位，没有变化，都为 AAAAAA。</div> <div>低 8 位，发生变化，从 CC 变成 FC。</div> <div>CC   F0 = FC</div>