

指令说明

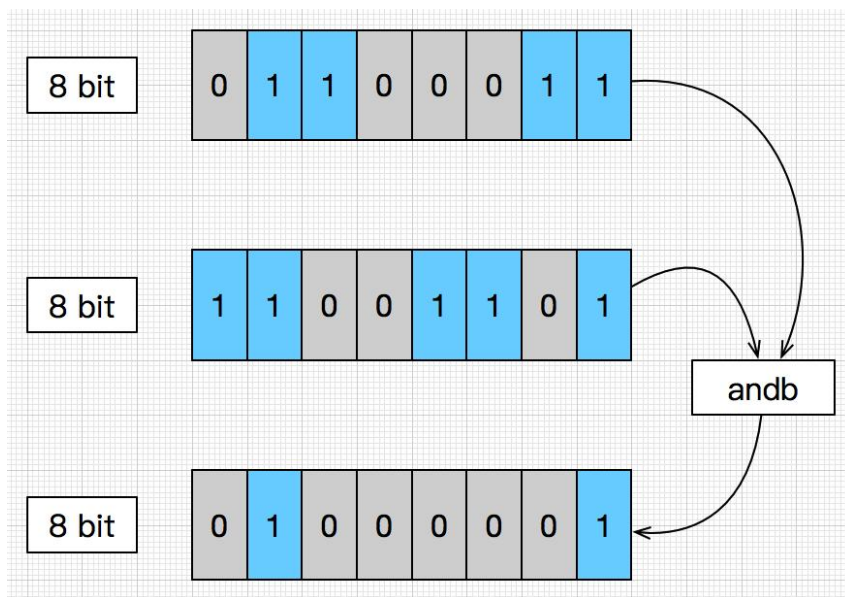
and 指令表示位与。

and 指令分为 8 位 andb、16 位 andw、32 位 andl、64 位 andq。

and 指令可以操作立即数、寄存器、内存。

语法格式 `and bb, aa` 表示 `aa = aa & bb`。

对应的位执行位与操作。如果 2 个位都为 1，则结果为 1，否则结果为 0。



用汇编代码分析

编写代码： `and_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 $0x1111111111111111, %rbx # 64 位, 寄存器
andq %rbx, %rax                # 64 位, 位与
movq $str64, %rdi
movq %rax, %rsi
callq printf

# 32 位。操作寄存器
movl $0xA2B2C2D2, %ecx # 32 位, 寄存器
movl $0xA2FFFF00, %edx # 32 位, 寄存器
andl %edx, %ecx        # 32 位, 位与
movq $str32, %rdi
movl %ecx, %esi
callq printf

# 32 位。操作栈内存
movl $0xFFFFFFFF, -8(%rbp) # 32 位, 栈
andl $0x33000007, -8(%rbp) # 32 位, 位与
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
andw $0xFF00, %bx      # 16 位, 位与
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
andb $0xF0, %bl        # 8 位, 位与
movq $str32, %rdi
movl %ebx, %esi
callq printf

addq $64, %rsp
popq %rbp
retq

```

编译代码:

```
gcc and_bit.s -o and_bit
```

运行代码:

```
[root@local binary]# ./and_bit
```

```

int64  0X1010101010101010
int32  0XA2B2C200
int32  0X33000007
int32  0XAAAACCO0
int32  0XAAAAAACO

```

分析结果：

汇编代码	结果和分析
# 64 位。操作寄存器 movq \$0x1010101010101010, %rax # 64 位, 寄存器 movq \$0x1111111111111111, %rbx # 64 位, 寄存器 andq %rbx, %rax # 64 位, 位与	<pre>int64 0X1010101010101010</pre> <p>操作 64 位寄存器 rbx、rax。 0x1010101010101010 & 0x1111111111111111 = 0X1010101010101010</p>
# 32 位。操作寄存器 movl \$0xA2B2C2D2, %ecx # 32 位, 寄存器 movl \$0xA2FFFF00, %edx # 32 位, 寄存器 andl %edx, %ecx # 32 位, 位与	<pre>int32 0XA2B2C200</pre> <p>操作 32 位寄存器 edx、ecx。 0xA2B2C2D2 & 0xA2FFFF00 = 0XA2B2C200 A2 与 A2 执行位与等于 A2。 B2C2 与 FFFF 执行位与等于 B2C2。 D2 与 00 执行位与等于 00。</p>
# 32 位。操作栈内存 movl \$0xFFFFFFFF, -8(%rbp) # 32 位, 栈 andl \$0x33000007, -8(%rbp) # 32 位, 位与	<pre>int32 0X33000007</pre> <p>操作栈内存。 0xFFFFFFFF & 0x33000007 = 0X33000007</p>
# 16 位。操作寄存器 movl \$0xAAAAAAAA, %ebx # 32 位, 占位 movw \$0xCCCC, %bx # 16 位, 寄存器 andw \$0xFF00, %bx # 16 位, 位与	<pre>int32 0XAAAACCO0</pre> <p>操作 16 位寄存器 bx。 高 16 位, 没有变化, 都为 AAAA。 低 16 位, 发生变化, 从 CCCC 变成 CC00。 CCCC & FF00 = CC00</p>
# 8 位。操作寄存器 movl \$0xAAAAAAAA, %ebx # 32 位, 占位 movb \$0xCC, %bl # 8 位, 寄存器 andb \$0xF0, %bl # 8 位, 位与	<pre>int32 0XAAAAAACO</pre> <p>操作 8 位寄存器 bl。 高 24 位, 没有变化, 都为 AAAAAA。 低 8 位, 发生变化, 从 CC 变成 C0。 CC & F0 = C0</p>