



链滴

shell 脚本中的 #是什么意思? `aa=$((2#1 number | index))` 最终 aa 等于多少?

作者: [liabio](#)

原文链接: <https://ld246.com/article/1574866040146>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

作者简介

作者：小碗汤，一位热爱、认真写作的小伙，目前维护原创公众号：『我的小碗汤』，专注于写linux golang、docker、kubernetes等知识等提升硬实力的文章，期待你的关注。转载说明：务必注明来（注明：来源于公众号：我的小碗汤，作者：小碗汤）

最近在网上看到一个问题：

```
#!/bin/bash
number=10
index=20
aa=$(( 2#1 << number | index))
echo ${aa}
```

执行上面的shell结果是什么？

#作用是什么？

首先得弄清楚#是什么意思？网上搜了一大圈没搜到，自己做几个实验看看：

```
[root@liabio test]#
[root@liabio test]# echo $((2#000))
0
[root@liabio test]# echo $((2#001))
1
[root@liabio test]# echo $((2#010))
2
[root@liabio test]# echo $((2#011))
3
[root@liabio test]# echo $((2#012))
-bash: 2#012: value too great for base (error token is "2#012")
[root@liabio test]# echo $((2#100))
4
[root@liabio test]# echo $((2#101))
5
[root@liabio test]# echo $((2#102))
-bash: 2#102: value too great for base (error token is "2#102")
[root@liabio test]# echo $((2#110))
6
[root@liabio test]# echo $((2#111))
7
[root@liabio test]# echo $((2#1000))
8
[root@liabio test]# echo $((16#ff))
255
[root@liabio test]# echo $((16#1))
1
[root@liabio test]# echo $((16#16))
22
[root@liabio test]# echo $((16#15))
21
[root@liabio test]# echo $((16#10))
16
```

```

[root@liabio test]# echo $((16#9))
9
[root@liabio test]# echo $((16#a1))
161
[root@liabio test]# echo $((16#a))
10
[root@liabio test]# echo $((16#b))
11

```

可以看到#作用是进制转换，即a#b代表a进制的b转换为十进制。

\$[]和\$(())

它们是一样的，都是进行数学运算的。支持+ - * / %：分别为“加、减、乘、除、取模”。但是注，bash只能作整数运算，对于浮点数是当作字符串处理的。

运算符优先级

左移符号<<和按位或符号|优先级哪个高？

优先级	运算符	名称或含义	使用形式	结合方向	说明
1	[]	数组下标	数组名 [常量表达式]	左到右	
1	()	圆括号	(表达式) / 函数名 (形参表)	左到右	
1	.	成员选择 (对象)	对象 . 成员名	左到右	

```

</tr>
<tr>
  <td>1</td>
  <td>-></td>
  <td>成员选择 (指针) </td>
  <td>对象指针 -> 成员名</td>
  <td>左到右</td>
  <td></td>
</tr>
<tr>
  <td>2</td>
  <td>-</td>
  <td>负号运算符</td>
  <td>- 表达式</td>
  <td>右到左</td>
  <td>单目运算符</td>
</tr>
<tr>
  <td>2</td>
  <td>( 类型 )</td>
  <td>强制类型转换</td>
  <td>( 数据类型 ) 表达式</td>
  <td>右到左</td>
  <td></td>
</tr>
<tr>
  <td>2</td>
  <td>++</td>
  <td>自增运算符</td>
  <td>++ 变量名 / 变量名 ++</td>
  <td>右到左</td>
  <td>单目运算符</td>
</tr>
<tr>
  <td>2</td>
  <td>--</td>
  <td>自减运算符</td>
  <td>-- 变量名 / 变量名 --</td>
  <td>右到左</td>
  <td>单目运算符</td>
</tr>
<tr>
  <td>2</td>
  <td>*</td>
  <td>取值运算符</td>
  <td>* 指针变量</td>
  <td>右到左</td>
  <td>单目运算符</td>
</tr>
<tr>
  <td>2</td>
  <td>&</td>
  <td>取地址运算符</td>
  <td>& 变量名</td>

```

```

    <td>右到左</td>
    <td>单目运算符</td>
</tr>
<tr>
    <td>2</td>
    <td>!</td>
    <td>逻辑非运算符</td>
    <td>! 表达式</td>
    <td>右到左</td>
    <td>单目运算符</td>
</tr>
<tr>
    <td>2</td>
    <td>~</td>
    <td>按位取反运算符</td>
    <td>~ 表达式</td>
    <td>右到左</td>
    <td>单目运算符</td>
</tr>
<tr>
    <td>2</td>
    <td>sizeof</td>
    <td>长度运算符</td>
    <td>sizeof( 表达式 )</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>3</td>
    <td>/</td>
    <td>除</td>
    <td>表达式 / 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>3</td>
    <td>*</td>
    <td>乘</td>
    <td>表达式 * 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>3</td>
    <td>%</td>
    <td>余数 (取模) </td>
    <td>整型表达式 / 整型表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>4</td>
    <td>+</td>

```

```

    <td>加</td>
    <td>表达式 + 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>4</td>
    <td>-</td>
    <td>减</td>
    <td>表达式 - 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>5</td>
    <td><<</td>
    <td>左移</td>
    <td>变量 << 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>5</td>
    <td>>></td>
    <td>右移</td>
    <td>变量 >> 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>6</td>
    <td>></td>
    <td>大于</td>
    <td>表达式 > 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>6</td>
    <td>>=</td>
    <td>大于等于</td>
    <td>表达式 >= 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>
    <td>6</td>
    <td><<</td>
    <td>小于</td>
    <td>表达式 < 表达式</td>
    <td>左到右</td>
    <td>双目运算符</td>
</tr>
<tr>

```

```

<td>6</td>
<td><=</td>
<td>小于等于</td>
<td>表达式 <= 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>7</td>
<td>==</td>
<td>等于</td>
<td>表达式 == 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>7</td>
<td>!=</td>
<td>不等于</td>
<td>表达式 != 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>8</td>
<td>&</td>
<td>按位与</td>
<td>表达式 & 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>9</td>
<td>^</td>
<td>按位异或</td>
<td>表达式 ^ 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>10</td>
<td>|</td>
<td>按位或</td>
<td>表达式 | 表达式</td>
<td>左到右</td>
<td>双目运算符</td>
</tr>
<tr>
<td>11</td>
<td>&&</td>
<td>逻辑与</td>
<td>表达式 && 表达式</td>
<td>左到右</td>
<td>双目运算符</td>

```

```

</tr>
<tr>
  <td>12</td>
  <td>||</td>
  <td>逻辑或</td>
  <td>表达式 || 表达式</td>
  <td>左到右</td>
  <td>双目运算符</td>
</tr>
<tr>
  <td>13</td>
  <td>?:</td>
  <td>条件运算符</td>
  <td>表达式 1? 表达式 2: 表达式 3</td>
  <td>右到左</td>
  <td>三目运算符</td>
</tr>
<tr>
  <td>14</td>
  <td>=</td>
  <td>赋值运算符</td>
  <td>变量 = 表达式</td>
  <td>右到左</td>
  <td></td>
</tr>
<tr>
  <td>14</td>
  <td>/=</td>
  <td>除后赋值</td>
  <td>变量 /= 表达式</td>
  <td>右到左</td>
  <td></td>
</tr>
<tr>
  <td>14</td>
  <td>*</td>
  <td>乘后赋值</td>
  <td>变量 *= 表达式</td>
  <td>右到左</td>
  <td></td>
</tr>
<tr>
  <td>14</td>
  <td>%=</td>
  <td>取模后赋值</td>
  <td>变量 %= 表达式</td>
  <td>右到左</td>
  <td></td>
</tr>
<tr>
  <td>14</td>
  <td>+=</td>
  <td>加后赋值</td>
  <td>变量 += 表达式</td>

```



```

    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td>-=</td>
    <td>减后赋值</td>
    <td>变量 -= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td><<=</td>
    <td>左移后赋值</td>
    <td>变量 <<= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td>>>=</td>
    <td>右移后赋值</td>
    <td>变量 >>= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td>&=</td>
    <td>按位与后赋值</td>
    <td>变量 &= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td>^=</td>
    <td>按位异或后赋值</td>
    <td>变量 ^= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>14</td>
    <td>|=</td>
    <td>按位或后赋值</td>
    <td>变量 |= 表达式</td>
    <td>右到左</td>
    <td></td>
</tr>
<tr>
    <td>15</td>
    <td>,</td>

```

```
<td>逗号运算符</td>
<td>表达式, 表达式, ...</td>
<td>左到右</td>
<td>从左向右顺序运算</td>
</tr>
</table>
```

同一优先级的运算符，运算次序由结合方向所决定。

简单记就是：！ > 算术运算符 > 关系运算符 > && > || > 赋值运算符

结果

```
#!/bin/bash
number=10
index=20
aa=$(( 2#1 << number | index))
echo ${aa}
```

1左移10位相当于 $1*2^{10}=1024$

```
10000000000
00000010100
```

```
-----
10000010100
```

即10进制的 $2^2+2^4+2^{10}=1044$

```
[root@liabio test]# cat aa.sh
#!/bin/bash
```

```
number=10
index=20
```

```
aa=$((2#1 << number | index))
echo $aa
```

```
[root@liabio test]# sh aa.sh
1044
```

参考

[Linux shell 中\\${}, \\$\[\], \\$\(\), \[\], \(\), \[\[\]\]作用与区别](#)

[C语言运算符优先级表](#)

作者简介

作者：小碗汤，一位热爱、认真写作的小伙伴，目前维护原创公众号：『我的小碗汤』，专注于写linux、golang、docker、kubernetes等知识等提升硬实力的文章，期待你的关注。转载说明：务必注明来（注明：来源于公众号：我的小碗汤，作者：小碗汤）