

# 位掩码(BitMask)的介绍与使用



Leewins (/u/9ec4a59d2700) + 关注

0.5 2018.05.06 23:28 字数 1185 阅读 3309 评论 0 喜欢 7

(/u/9ec4a59d2700)

## 一、前言

位运算在我们实际开发中用得很少，主要原因还是它对于我们而言不好读、不好懂、也不好计算，如果不经常实践，很容易就生疏了。但实际上，位运算是一种很好的运算思想，它的优点自然是计算快，代码更少。

## 二、基本知识介绍

- 二进制：

二进制是由1和0两个数字组成的，它可以表示两种状态，即开和关。所有输入电脑的任何信息最终都要转化为二进制。目前通用的是ASCII码。最基本的单位为bit。

- 位运算：

程序中的所有数在计算机内存中都是以二进制的形式储存的。位运算说穿了，就是直接对整数在内存中的二进制位进行操作。比如，and运算本来是一个逻辑运算符，但整数与整数之间也可以进行and运算。举个例子，6的二进制是110，11的二进制是1011，那么6 and 11的结果就是2，它是二进制对应位进行逻辑运算的结果（0表示False，1表示True，空位都当0处理）。

## 三、问题引用

- 老鼠试毒

有1000瓶水，其中有一瓶有毒，小白鼠只要尝一点带毒的水24小时后就会死亡，问至少需要多少只小白鼠才能在24小时内鉴别出哪瓶水有毒？



老鼠试毒.png



这里，位掩码的使用就可以巧妙的解决此问题。

我们先将问题简化一下：假设只有8瓶水，其中1瓶有毒。

(/apps/redi  
utm\_sourc  
banner-lic

0	0	0	1	--	1号水
0	0	1	0	--	2号水
0	0	1	1	--	3号水
0	1	0	0	--	4号水
0	1	0	1	--	5号水
0	1	1	0	--	6号水
0	1	1	1	--	7号水
1	0	0	0	--	8号水

8杯水分别编号.png

将该矩阵转置，得：

0	0	0	0	0	0	0	1	--	第一杯
0	0	0	1	1	1	1	0	--	第二杯
0	1	1	0	0	1	1	0	--	第三杯
1	0	1	0	1	0	1	0	--	第四杯
1	2	3	4	5	6	7	8	--	号水

水杯矩阵转置.png

依上述场景，取4只容器，转置后的矩阵数列配组合溶液：

取数位上为1的水，放入相应的容器，即：

第一杯：只包含8号水

第二杯：包含4、5、6、7号水

第三杯：包含2、3、6、7号水

第四杯：包含1、3、5、7号水

取4只老鼠，编号1、2、3、4，分别喝下第一杯...第四杯水，

4只老鼠的生死状态依次记为 w x y z， (w,x,y,z = {0,1})

死亡记作1，非死亡记作0

将二进制数列wxyz转为十进制，则得到有毒水的号码。

假设6号水有毒，那么往回推算，不难看出，第2、3只老鼠会死亡，得到的wxyz的数列就是0110，转十进制后就是6。



将1000瓶依次编号：1，2，3，4，...，1000；且都记作二进制；

那我们要用多少位来表示呢？

总数是1000， $2^9=512$ ， $2^{10}=1024$ ，于是至少要10位才够表示，

也就是：0000000001，0000000010，0000000011，...，1111101000；

道理同上。

(/apps/redi  
utm\_sourc  
banner-clic

## 四、结合实际问题

我们已经见识了二进制的厉害之处了，接下来我们结合代码来看看，在iOS开发中的应用（其实在任何开发中都一样）

- 在实际开发中，我们常常遇到权限的判断的问题，比如说，不同的用户对系统有不同的操作权限，有的用户可能有多种权限，我们最常规的办法就是每一个权限定义一个BOOL值。

假设，某系统有4种权限，那么，就有了：

```
@interface BM_User : NSObject

@property (nonatomic, assign) BOOL permission1;

@property (nonatomic, assign) BOOL permission2;

@property (nonatomic, assign) BOOL permission3;

@property (nonatomic, assign) BOOL permission4;

@end
```

那用户A同时拥有permission1、permission2、permission4怎么表示呢？

```
BM_User *userA = [[BM_User alloc] init];
userA.permission1 = YES;
userA.permission2 = YES;
userA.permission4 = YES;
```

这样的操作大家见多了吧？那我们来看看另一种写法：

```
@interface BM_User : NSObject

@property (nonatomic, assign) OptionPermission permission;

@end
```

有人就要问了，OptionPermission是什么鬼？来，继续。。。



```
/**
 权限枚举

- 1: permission1, 二进制第1位, 0表示否, 1表示是
- 2: permission2, 二进制第2位, 0表示否, 1表示是
- 4: permission3, 二进制第3位, 0表示否, 1表示是
- 8: permission4, 二进制第4位, 0表示否, 1表示是
*/
typedef NS_OPTIONS(NSUInteger, OptionPermission) {
    permission1 = 1 << 0, //0001, 1
    permission2 = 1 << 1, //0010, 2
    permission3 = 1 << 2, //0100, 4
    permission4 = 1 << 3, //1000, 8
};
```

(/apps/redi  
utm\_sourc  
banner-clc

那用户A同时拥有permission1、permission2、permission4怎么表示呢?

```
BM_User *userA = [[BM_User alloc] init];
userA.permission = permission1 | permission2 | permission4;
```

是不是神清气爽?

现在我们就具体化4种权限, 并给出基础位掩码的表达及运算:

```
#ifndef BM_Head_h
#define BM_Head_h

/**
 权限枚举

- 1: 是否允许查询, 二进制第1位, 0表示否, 1表示是
- 2: 是否允许新增, 二进制第2位, 0表示否, 1表示是
- 4: 是否允许修改, 二进制第3位, 0表示否, 1表示是
- 8: 是否允许删除, 二进制第4位, 0表示否, 1表示是
*/
typedef NS_OPTIONS(NSUInteger, OptionPermission) {
    ALLOW_SELECT = 1 << 0, //0001, 1
    ALLOW_INSERT = 1 << 1, //0010, 2
    ALLOW_UPDATE = 1 << 2, //0100, 4
    ALLOW_DELETE = 1 << 3, //1000, 8
};

#endif /* BM_Head_h */
```



```
#import "BM_Permission.h"
#import "BM_Head.h"

@interface BM_Permission ()

/** 存储目前的权限状态 */
@property (nonatomic, assign) OptionPermission flag;

@end

@implementation BM_Permission

/** 重新设置权限 */
- (void)setPermission:(OptionPermission)permission {
    self.flag = permission;
}

/** 添加一项或多项权限 */
- (void)enable:(OptionPermission)permission {
    self.flag |= permission;
}

/** 删除一项或多项权限 */
- (void)disable:(OptionPermission)permission {
    self.flag &= ~permission;
}

/** 是否拥有某些权限 */
- (BOOL)siAllow:(OptionPermission)permission {
    return (self.flag & permission) == permission;
}

/** 是否禁用了某些权限 */
- (BOOL)isNotAllow:(OptionPermission)permission {
    return (self.flag & permission) == 0;
}

/** 是否仅仅拥有某些权限 */
- (BOOL)isOnlyAllow:(OptionPermission)permission {
    return self.flag == permission;
}
```

(/apps/redi  
utm\_sourc  
banner-clic

## 五、写在最后

- 大家还可以自行搜索一下NS\_OPTIONS与NS\_ENUM的区别，他们都是用来定义枚举的，但其用法是有很大不同。
- 博主我最近一直在考虑优化代码，正在开发的项目中就有很多权限判断的问题，我也在寻找各种各样更好的写法。
- 也希望大家重视代码的表达，因此更加优化自己的代码。

小礼物走一走，来简书关注我

赞赏支持

 iOS (/nb/18270823)

举报文章 © 著作权归作者所有



Leewins (/u/9ec4a59d2700) ♂



写了 14650 字，被 23 人关注，获得了 34 个喜欢

+ 关注

(/u/9ec4a59d2700)

道可道，非常道；名可名，非常名。

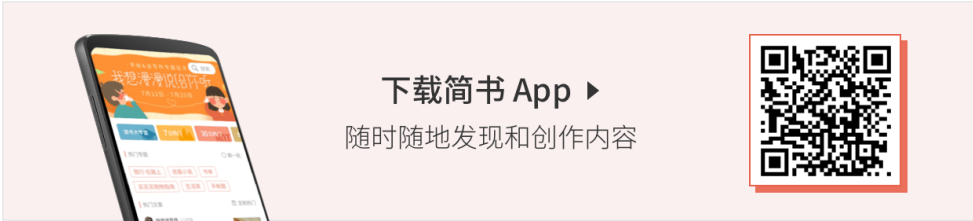
喜欢

7



更多分享

(/apps/redi  
utm\_sourc  
banner-clic



(/apps/redirect?utm\_source=note-bottom-click)



登录后发表评论 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-comr

评论

智慧如你，不想发表一点想法 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-nocomments-text)咩~

被以下专题收入，发现更多相似内容



program... (/c/40f2c0150821?utm\_source=desktop&utm\_medium=notes-included-collection)



计算机科普 (/c/a6a1c4525c20?utm\_source=desktop&utm\_medium=notes-included-collection)

JavaScript程序 (/p/22c4737e8272?utm\_campaign=maleskine&utm\_co...

第2章 基本语法 2.1 概述 基本句法和变量 语句 JavaScript程序的执行单位为行（line），也就是一行一行地执行。一般情况下，每一行就是一个语句。语句（statement）是为了完成某种任务而进行的操作，比如下...

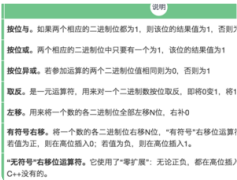


泰迪学长\_ (/u/e0d5c1d81e8b?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio




(/p/11f331d97ec2?



(/apps/redi  
utm\_sourc  
banner-cli

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendatio  
**【小知识大道理】被忽视的位运算 (/p/11f331d97ec2?utm\_campaign=mal...**


Bitwise Operation导语众所周知计算机是基于二进制01进行运算的，理所当然地，位运算相对于各种算术运算更加贴合计算机的二进制语义，运算效率会更快。这样计算机是舒服了，人类读起来就太生涩了，所以...

 曲高和寡\_健 (/u/614a0923878c?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

**Java面试宝典Beta5.0 (/p/fb7d48083e5e?utm\_campaign=maleskine&ut...**


pdf下载地址：Java面试宝典 第一章内容介绍 20 第二章JavaSE基础 21 一、Java面向对象 21 1. 面向对象都有哪些特性以及你对这些特性的理解 21 2. 访问权限修饰符public、private、protected, 以及不写（默认）...

 王震阳 (/u/773a782d9d83?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

**C#位运算讲解与示例1 (/p/92be848fecde?utm\_campaign=maleskine&ut...**

今天在项目中遇到按位或组合权限串的问题: 首先每一个权限数都是2的N次方数 如: k1=2; //添加 k2=4; //删除 k3=8; //修改 ... 如此定义功能权限数，当需要组合权限时，就需要对各个所拥有的权限数按位或了。如...

 某人在 (/u/4673233b8b76?


utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

(/p/8116c03460c9?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendatio  
**阳光正好的周末 (/p/8116c03460c9?utm\_campaign=maleskine&utm\_con...**

2017-12-03 晴 早上起来就看见了暖暖的阳光，一个温暖的周末。无拘无束的度过了一天的闲暇时光，午后约一俩好友，渭河边逛逛。阳光正好，与君初相识。好好学习，天天向上！

 于勇 (/u/118dc85e1d08?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

(/p/07efb88eccc0?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendatio  
**剑毒梅香（4） (/p/07efb88eccc0?utm\_campaign=maleskine&utm\_cont...**

城破残墙冷夜深，漠然置母归黄泉 秦修在送完花轻烟后，极快的转身回了内院，果然见杜锦官已经在等待了，面色阴沉，透着很深的戾气。连忙躬身上前道：“城主，他走了。”“嗯。”杜锦官淡淡的应了一声，似乎...


 沅抒 (/u/a7f0320fdd3a?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio



天下最聪明的投资人 (/p/544816f91559?utm\_campaign=maleskine&utm\_...


大学四年坐了无数次火车，什么都坐过，觉得自己和那些娇气的女生不一样，什么都可以忍受，直到今晚，突然好想哭，内心的恐惧害怕，感觉特别真实，想找人诉说却不知该怎么说出口。原来自己没有想象的那...

 FeliciaYolo (/u/c57d6ae11e38?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio  
banner-clic (/apps/redi

git命令行用法---分支 (/p/a20aac266e7e?utm\_campaign=maleskine&utm...


qq群：225451602 1 查看远程分支 git branch -r 2、查找远程分支 git branch -r | grep 分支名 3、 查看本地分支 git branch 4、git branch -a查看本地和远程所有分支 查找所有分支git branch ...

 RunningTeemo (/u/864e7fe73d7c?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

海底捞的模式，即使它自己也一定能完全复制 (/p/9f8b9c7a8a30?utm\_cam...

说起海底捞，那个好像有着传说一样的地方：第一家开在美国的中国的火锅店。而目前，看到的评论，国外的海底捞发展并不成功。 而最近因为公司服务规则修改的问题焦头烂额的时候朋友推荐了海底捞的书以供...

 扶不起的斗姐 (/u/2676d30401b6?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendatio

