

Android 中的 Enum 到底占多少内存？该如何用？

28 Apr 2016

听说过一些论调，Enum 不该用啊，占用了很大的 dex 文件，占用很多内存。而到底确切占用了多少内存，没说。本文分析了枚举所占用的精确的内存大小，方便大家权衡选择，希望对大家有帮助。

关于 Enum 的使用

Enum 需要占用较大的内存，如果对内存敏感，请尽量少使用 Enum，换用做静态常量。

文档提到：

Enums often require more than twice as much memory as static constants. You should strictly avoid using enums on Android.

关于具体要占用多少内存呢？说得比较模糊。

内存占用对比

我在《[Dalvik 中的对象大小](#)》一文中，介绍过如何衡量对象的大小，这个文章非常详细，建议大家看看，现举例说明。

```
public enum MonthEnum {  
    // 4 bytes  
    JANUARY, // -> 87 bytes  
    // 4 bytes  
    FEBRUARY // -> 88 bytes
```

```

// 生成的数组 24 + 4 + 4
// MonthEnum[] values
}

public class MonthConst {
    // 4 bytes
    public static final int JANUARY = 1;
    // 4 bytes
    public static final int FEBRUARY = 2;
}

public class UseMonth {

    // 4 bytes
    private int mMonth = MonthConst.JANUARY;

    // 4 bytes
    private MonthEnum mMonthEnum = MonthEnum.JANUARY;
}

```

我们不考虑 `MonthEnum` 和 `MonthConst` 他们对于 dex 大小的影响，这个没什么意义，几十个 `Enum` 占用的大小，也不及一张图片。

我们要对比的是 `UseMonth` 中这两种写法所占用的内存大小在 Dalvik 虚拟机下的区别。

在 `UseMonth` 中，他们一个是 `int` 类型，一个是对象引用，都是 4 字节，没有区别。

我们对比的大小，指的是对象本身的大小加上对象成员指向的其他对象大小，即 shadow heap + maintain heap。

- `MonthEnum`

对于一个 `MonthEnum`，`JANUARY` 和 `FEBRUARY` 是两个指向 `MonthEnum` 实例的引用。他们分别占用 4 个字节。

他们指向的实例对象还要占用额外的内存。

我们看看 `enum` 的定义：

```

class Enum {
    private final String name;
    private final int ordinal;
}

```

作为 `Enum` 成员变量 `name`（对象引用）和 `ordinal`（int）他们各占用 4 个字节，该对象实例占用： $12 + 4 + 4 = 20$ bytes，对齐之后是 24 字节。

但是，`name` 是字符串，空字符串对象本身就是 32 字节，加上其中的字符数组最少也会占据 24 个字节，对字符串加字符数组最少会占据 56 个字节。故一个 `Enum` 实例，最少 80 个字节。

`MonthEnum.JANUARY`，含有 7 个字符，87 个字节；`MonthEnum.FEBRUARY`，8 个字符，88 个字节。

枚举编译完之后会有一个 `values()` 数组，两个对象引用的数组占用： $24 + 4 + 4 = 32$ bytes。

总计是： $4 + 4 + 87 + 88 + 32$

- `MonthConst`

`JANUARY` 和 `FEBRUARY` 各占 4 个字节。共计 8 个字节。

总计是： $4 + 4$

上面我们对比了只具有两个枚举值的枚举和常量，如果数量更多的话，枚举的命名更长的话，这个差距会更大。

文档所说的两倍

所以实际占用的内存，并非 [文档](#) 所说的两倍左右。

假设有 n 个枚举值，仅仅考虑枚举类，静态占用的内存， n 个引用 + n 个数组 + 24 空数组长度： $8n + 24$ 。

而对于 n 个值的常量，则有 $4n$ 字节。当 n 很大时，这样的关系是两倍，但是枚举引用所指向的内存（retained heap）没有考虑进来。

该用不该用？

[文档](#) 提到：

You should strictly avoid using enums on Android.

枚举有其其他的特性，如果你需要这些特性，比如：非连续数值的判断，重载等时，可以用。

另外，内存用量也并非那么地可怕，枚举带来的编码的便捷，代码可读性的提升也是很大的利好。

看到这里，你应该了解了所有的细节了，是否该用，各位自己权衡。

更多的讨论，可以看这里：[该不该用枚举](#)。

如何更好地使用常量

如果应用确实对内存用量敏感，或者你就是追求极致，可用常量来代替枚举。

常量一般会 and Bit Mask 结合起来用，这样可以极致地减少了内存使用，同时使代码有较好的可读性。

下一篇文章会提到。

欢迎关注我的 [新浪微博](#)，有问题随时交流。

欢迎关注我的 [GitHub](#)，了解我最新关注的项目。