

Stack vs. Heap：了解 Java 的内存分配机制【已翻译100%】

英文原文：Stack vs. Heap: Understanding Java Memory Allocation

标签：Java

0 顶

oschina 推荐于 4天前 (共 8 段, 翻译完成于 08-17) 评论 0

参与翻译 (5人): 亚林瓜子, Tocy, 我是菜鸟我骄傲, 圣洁之子, leoxu

仅中文 中英文对照 仅英文 打印此文章

知道栈和堆之间的区别吗？什么时候该用哪一个，它们提供了什么功能？ 这是一篇关于内存分配的指南。

栈和堆是与关于 Java 内存分配的两个重要概念。我们来看看这两个概念，为什么它们很重要，以及什么时候使用它们。

Java 中的栈是什么？

Java 的栈是计算机内存的一部分，其中存储由你所编写的所有函数所创建的临时变量。它用于执行线程，并且可能具有某些短生命周期的值及对其他对象的引用。它使用 LIFO（[后进先出](#)）的数据结构。



Tocy

翻译于 4天前

0人顶

顶 翻译得不错哦！

其它翻译版本(1)

这是什么意思？当一个方法被调用时，它会为该特定方法在堆栈中创建一个新的块。新块将具有所有局部变量，以及对该方法正在使用的其他对象的引用。当方法结束时，新的块将被擦除，并且可以被下一个方法使用。您在这里找到的对象只能访问该特定函数，不能超越它。

这使得跟踪堆栈非常容易，最新的保留块也是首先被释放的。为方法创建的变量直接存储在内存中，可以快速访问。

Java堆栈的内存大小通常远小于Java堆空间，因为当一个方法结束时，堆栈上创建的所有变量将永远被删除。

下面是如何在堆栈中创建对象的示例：

```
void somefunction( )
{
    / *创建Member类成员的“m”对象
    它将被放在堆栈上
    我们没有使用“new”关键字
    在函数内创建对象
    * /

    Member m;
} //一旦函数结束，对象“m”被销毁
```



亚林瓜子

翻译于 4天前

0人顶

顶 翻译得不错哦！

Java 中的堆是什么？

Java对象位于一个称为堆的区域中。它是在程序运行时创建的，并且其大小可能随程序运行而减小或增加。它可以很容易地填满，当遇到类似堆满的情况时，垃圾回收机制会触发。即在当不再使用的对象被删除后，该为新对象腾出空间的时候。

与Java栈的在编译程序时完成内存分配完成不同的是，使用堆时，内存分配是在运行程序时完成的。访问放置在堆上的变量与栈上的直接和快速访问相比有点慢。

堆比较像[全局内存池](#)。如果需要使用数据或变量的生命周期长于相关的方法或函数，方法或函数将使用堆进行内存分配。你在堆上获得的对象是可以被所有函数访问的。



Tocy

翻译于 4天前

0人顶

顶 翻译得不错哦！

此外，在堆的保留块中没有特定的顺序。你可以随时分配块，然后你可以随意释放它。您可以想像，跟踪和可分配的部分要复杂得多，但也可以分为两代或多个子区域。

这些分区被称为新生区（或苗圃）和旧空间。新生区通常用于新对象的内存分配。当新生区变得满满时，[垃圾回收](#)就会发生。短暂的或临时的物体通常使用新生区。与没有任何分区的堆相比，这样可使垃圾回收更快。

下面是如何在堆中创建对象的[示例](#)：



亚林瓜子

翻译于 4天前

0人顶

顶 翻译得不错哦！

```
void somefunction( )
{
    / *创建Member类成员的“m”对象
    这将被放在堆上
    我们使用“new”关键字
    在函数内创建对象
    * /

    Member* m = new Member( );

    / *对象“m”必须被删除
    否则会发生内存泄漏
    * /
    delete m;
}
```

堆和栈之间的相似与差异

两者都是[Java内存分配](#)的方式并且都存储在RAM中。然而，为了更容易记忆，堆用于动态内存分配，而栈用于静态分配。

它存储在哪里？在栈中分配的变量可直接从内存访问，因此这些变量可以运行得非常快。另一方面，访问堆中的对象需要更多的时间。

什么时候分配内存？在栈中，当程序被编译时分配内存。而在堆中，则是开始于程序运行时。

既然如此，如果要使用栈，则需要知道在编译之前需要多少数据和内存。栈的另一个限制是它不能处理需要大量内存的大块变量。如果您不知道运行时需要多少数据，或者如果需要大量数据的内存，则需要使用堆。



亚林瓜子

翻译于 4天前
0人顶

翻译得不错哦！

简而言之...

栈

- 栈的大小会随方法和函数增删局部变量而按需变化。
- 内存分配随后会被释放，而无需你去管理内存分配。
- 栈的大小有限制，根据你所使用的操作系统而定。
- 存储在栈中的变量一直存在，只要创建它们的函数在运行。

堆

- 内存不是自动管理的，也不是由中央处理器像栈那样严格管理。当这些块不再需要时，你需要自己释放已分配内存。
- 堆容易出现内存泄漏，即内存被分配给未使用的对象，而对其他进程则不可用。
- 堆没有大小限制。
- 与栈相比，堆中的对象访问起来慢得多，堆内存中的写操作也比较慢。



圣洁之子

翻译于 2天前
0人顶

翻译得不错哦！

栈使用起来更简单、更快捷，但它有许多限制，如果使用堆，你可以忽略那些限制。

什么时候使用栈？ 栈只能用于局部变量，使用少量的内存。好消息是内存分配和管理不会成为你的问题，对这些对象的访问非常快。它的大小确实受到限制，事实上你也不能调整栈的变量的大小。

什么时候用堆？ 如果存在需要全局访问的变量，而不是仅对创建它的方法和函数可用，则使用堆来分配内存。当你需要大量的内存，堆也是不错的，因为它对内存大小没有限制。你还可以调整堆的变量的大小。



圣洁之子

翻译于 2天前
1人顶

翻译得不错哦！

其它翻译版本(1)

其它的资源与教程

如果你想要更加详细地了解有关与堆和栈之间的差异，还有最佳的实践，那就试着访问下面这些资源和教程吧：

- [《内存空间：栈VS堆》](#)
- [《堆和栈之间有什么区别？》](#)
- [《堆与栈的内存分配》](#)



leoxu

翻译于 2天前
0人顶

翻译得不错哦！

- [《.NET中的C # 堆与栈（ 以及它们是如何分配的 ）：第一部分》](#)

堆和栈是Java中分配内存的两种方式。如果变量要可以全局访问，那堆是较好的选择，而栈适合于只需要少量内存的局部变量。了解什么时候以及如何使用堆栈和堆对于你开发出更好的Java程序而言至关重要。

知道内存分配机制在处理[内存泄漏](#)的时候是如何工作的，这一点也很有帮助。 如果你想要对它们的工具，网站，博客和其他资源有一个全盘的掌握，那就需要学习进阶的知识了，可以去下载我们的[《Java开发者综合指南》](#)。



本文中的所有译文仅用于学习和交流目的，转载请务必注明文章译者、出处、和本文链接
我们的翻译工作遵照 [CC 协议](#)，如果我们的工作有侵犯到您的权益，请及时联系我们

评论(0)



Ctrl/CMD+Enter

发表评论

暂无网友评论