

国庆策划03 | 揭秘代码优化操作和栈保护机制

2022-10-07 LMOS 来自北京

天下无鱼
<https://shikey.com/>

《计算机基础实战课》

[课程介绍 >](#)



讲述：陈晨

时长 05:24 大小 4.95M



你好，我是 LMOS。

今天是国庆假期策划的第三期。我们来公布第一期主观题的答案。希望你先尝试自己梳理思路，尝试回答问题以后，再来查看参考答案。

第一题

在前面课程里，我们一起揭秘了 C 语言编译器的“搬砖”日常，搞清楚了 C 语言会如何处理各种类型变量、各种运算符、流程控制以及由它们组成的函数，并把这些内容加以转换，对应到机器指令。你知道在这个转换过程中，C 编译器为了提高程序的执行性能，会有哪些额外的操作吗？试试概括一下这些操作？

第一题参考答案

存在额外的操作，概括来说是对代码进行优化操作。

为了提高程序的执行性能，C 语言编译器在经过语义分析的阶段之后，会生成平台无关的中间代码，然后经历三次不同级别的代码优化。

这里首先要经历中间代码级的代码优化；而后，编译器把中间代码优化的结果作为输入，生成机器相关的目标代码；之后还会再经过一次目标代码级别的代码优化，这个优化策略和具体机器的硬件结构高度相关，且不通用。

完成了整个优化过程后，就会产生最终运行机器平台上的目标代码了。一般通用的优化代码操作具体包括四个方面，我们挨个来看看。

第一类操作是**删除多余运算**。编译器分析中间代码的时候，可能会发现一些计算操作属于重复计算。因为有些计算并没有让结果发生变化，它们是多余的，完全可以删除。

第二类是**代码外提操作**，一般用在优化循环代码，可以减少循环中代码的总数。它的原理是这样的：如果循环中的计算结果不改变某个代码段，我们就把这段代码外提，放在循环外。这种变换把计算结果不受循环执行次数影响的表达式，提到了循环的前面，使之只在循环外计算一次。

第三类是**强度削弱操作**。强度削弱的本质是把强度大的运算换算成强度小的运算。举例来说，把加法换成乘法运算强度会更小。比如循环过程，每循环一次，变量的值增加 1，又不与循环相关，每次总是增加相同的数据。因此，可以把循环中计该值的加法运算变换成在循环前进行一次乘法运算。

最后一类操作是**合并已知量和复写传播**。有时很多运算结果都是编码时已知的，所以在代码编译时就可以计算出它们的值，我们把这种变换称为合并已知量。

还有多个变量之间的互相引用，比如变量 A 被变量 B 引用，而变量 B 又被变量 C 引用，如果 A 与 C 之间没有能够改变 B 的代码，就直接让 C 引用 A，这种变换称为复写传播。

第二题

在 [🔗堆与栈的区别和应用](#)这节课中，我们知道了堆与栈区别。同时，我们也清楚了 C 语言的函数的局部变量和返回地址都保存在栈中，如果有人对这栈中数据破坏就会导致安全隐患，例如改写返回地址，使之指向别的恶意程序。那问题来了，请问我们有什么栈保护机制么，可以用你的语言描述一下么？

第二题参考答案

栈保护机制有很多，我给你分享比较典型的几种。

首先是由编译器在编译程序时，稍微做个检查，看看是否存在栈内缓冲区溢出的错误。程序代码中采用大量的字符串或者内存操作的函数，比较适合做这样的检查。通过给 `gcc` 加上 `-D_FORTIFY_SOURCE=1` 或者 `2` 时，在编译或者代码运行时，通过判断数组大小来替换 `strcpy`、`memcpy`、`memset` 等函数名，将它们替换成编译器中带有检查代码的函数，从而防止缓冲区溢出。

通过操作系统对页表的 `NX` 位进行设置，这种方法也很常见。`NX` 即 `No-eXecute`，意思是不可执行。带 `NX` 位的页表所指向的内存中的数据是不可执行的，当程序溢出成功转入恶意代码时，程序会尝试在数据页面上执行指令，此时 `CPU` 就会抛出异常，不去执行恶意代码，主要防止恶意代码在数据区溢出。


还有一种简称为 `ASLR` 的方法，即地址空间分布随机化。内存空间地址随机化机制可以将进程的 `mmap` 基地址、`heap` 基地址、栈基地址、共享库基地址随机化。这样能有效阻止攻击者在堆、栈上运行恶意代码。

最后还有栈溢出保护 `canary`，这是一种由编译器支持的技术。在 `Linux` 中将 `cookie` 信息称为 `canary`。攻击者在覆盖返回地址的时候往往也会将 `cookie` 信息给覆盖掉，导致栈保护检查失败。

而 `canary` 技术的大致思路是这样的，当启用栈溢出保护后，编译器会插入相关代码，在函数开始执行的时候就会向栈里写入 `cookie` 信息。当函数真正返回的时候，就会通过编译器插入的代码来验证 `cookie` 信息是否合法。如果不合法程序就会停止运行，这样就能阻止恶意攻击代码的执行。

通过这两道题目，我们又补充了代码优化和栈保护机制的知识。接下来，我们继续回到课程主线的学习，期待你把精神状态拉满，之后学有所成！

分享给需要的人，Ta购买本课程，你将得 20 元

 生成海报并分享

上一篇 国庆策划02 | 来自课代表的学习锦囊

下一篇 31 | 外设通信：IO Cache与IO调度

精选留言 (2)

💬 写留言



peter

2022-10-08 来自北京

请教一个问题：

Q1： 随机地址是物理地址还是虚拟地址？

文中提到的**ALSR**方法，会把地址空间进行随机化分布，请问被随机化分布的地址是物理地址还是虚拟地址？



LockedX

2022-10-08 来自广东

彭老师可以开一个编译器的课程，编译器可太有意思了^_^

