函数内联的优点如下：

* **消除调用和返回以及参数传递的开销**：避免了函数调用时保存和恢复寄存器、传递参数等操作带来的时间和资源消耗。
* **改善代码缓存**：由于代码变得连续，更有利于 CPU 的指令缓存，能提高缓存命中率，减少从内存读取指令的时间。
* **减少代码体积（单调用情况）**：如果对被内联的函数只有一次调用，内联后代码总体积可能会变小。
* **开启其他优化可能**：如下面所解释的，函数内联可以为编译器进行其他优化提供更多的机会。

函数内联的缺点是，如果对被内联的函数有多次调用，并且该函数体较大，代码体积会增大。编译器更倾向于内联那些代码量小或者只在一两个地方被调用的函数。

9优化内存访问

9.3一起使用的函数应该存储在一起

9.4一起使用的变量应该存储在一起

如果一起使用的数据在内存中彼此相邻存储，缓存的工作效率将达到最高。变量和对象最好在使用它们的函数中声明。这样的变量和对象将存储在栈上，而栈极有可能位于一级缓存中。不同类型的变量存储方式在第 25 页有详细解释。如果可能的话，应避免使用全局变量和静态变量，并避免进行动态内存分配（使用 new 和 delete）。

面向对象编程是一种将数据组织在一起的有效方式。类的数据成员（也称为属性）总是存储在该类的对象中。父类和派生类的数据成员会存储在派生类的对象中（详见第 52 页）

9.5数据对齐

文本

描述已自动生成

文本, 信件

描述已自动生成

**自制容器类**

一个高效的容器类应该尽可能少地分配内存块。理想情况下，所有数据都应该存储在一个连续的内存块中。只有在编译时无法确定合理的最大大小时，才需要进行动态内存分配。容器所包含的对象类型可以方便地作为模板参数指定。记得定义一个析构函数来释放所有分配的内存。

使用静态链接而非动态链接的优点如下：

* 静态链接仅包含应用程序实际需要的库部分，而动态链接即使只需要库代码的一小部分，也会将整个库（或至少是其中很大一部分）加载到内存中。
* 使用静态链接时，所有代码都包含在单个可执行文件中。动态链接则需要在程序启动时加载多个文件。
* 调用动态库中的函数比调用静态链接库中的函数花费的时间更长，因为它需要通过导入表中的指针进行额外的跳转，并且可能还需要在过程链接表（PLT）中进行查找。
* 当代码分布在多个动态库中时，内存空间会变得更加碎片化。动态库会加载到能被内存页大小（4096）整除的整数内存地址处。这会导致所有动态库竞争相同的缓存行，从而降低代码缓存和数据缓存的效率。
* 由于需要位置无关代码，动态库在某些系统中的效率较低，详见下文。
* 如果使用动态链接，安装一个使用同一动态库较新版本的第二个应用程序可能会改变第一个应用程序的行为，但如果使用静态链接则不会出现这种情况。

动态链接的优点如下：

* 多个同时运行的应用程序可以共享同一个动态库，而无需将库的多个实例加载到内存中。这在同时运行多个进程的服务器上非常有用。实际上，只有代码段和只读数据段可以共享。任何可写数据段每个进程都需要一个实例。
* 动态库可以更新到新版本，而无需更新调用它的程序。
* 动态库可以被不支持静态链接的编程语言调用。
* 动态库可用于制作插件，为现有程序添加功能。