

用GNUprofiler提高代码运行速度 上 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_\\_E7\\_94\\_A8G](https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8G)

NUprof\_c103\_144234.htm 改进应用程序的性能是一项非常耗时耗力的工作，但是究竟程序中是哪些函数消耗掉了大部分执行时间，这通常都不是非常明显的。在本文中我们将学习如何使用 gprof 为 Linux ？？平台上的用户空间和系统调用精确分析性能瓶颈。简介 各种软件对于性能的需求可能会有很大的区别，但是很多应用程序都有非常严格的性能需求，这一点并不奇怪。电影播放器就是一个很好的例子：如果一个电影播放器只能以所需要速度的 75% 来播放电影，那么它几乎就没什么用处了。其他应用程序（例如视频编码）如果是耗时非常长的操作，最好以“批处理”任务的方式运行，此时启动一个作业，让其一直运行，然后我们就可以去干别的事情了。尽管这些类型的应用程序没有这种硬性性能指标的限制，但是提高速度仍然会带来很多好处，例如可以在给定的时间内可以对更多电影进行编码，在同样的时间内可以以更高的品质进行编码。通常，除了最简单的应用程序之外，对于其他应用程序来说，性能越好，这个应用程序的用处就越大，也就会越流行。由于这个原因，性能考虑是（也应该是）很多应用程序开发人员脑袋中的第一根弦。不幸的是，很多尝试让应用程序速度更快的努力都白费了，因为开发人员通常都是对自己的软件进行一些小型的优化，而没有去研究程序在更大的范围内是如何操作的。例如，我们可能会花费大量的时间来让某个特定函数的运行速度达到原来的两倍，这一点非常不错，但是如果这个函数很少被调用（例如打

开文件)，那么将这个函数的执行时间从 200ms 减少到 100ms，对于整个软件的总体执行时间来说并不会太大的影响。有效地利用您的时间的方法是，尽量优化软件中被频繁调用的部分。例如，假设应用程序花了 50% 的时间在字符串处理函数上，如果可以对这些函数进行优化，提高 10% 的效率，那么应用程序的总体执行时间就会改进 5%。因此，如果希望能够有效地对程序进行优化，那么精确地了解时间在应用程序中是如何花费的，以及真实的输入数据，这一点非常重要。这种行为就称为代码剖析（code profiling）。本文将简要介绍 GNU 编译器工具包所提供的一种剖析工具，它的名字让人可以产生无限遐想，叫 GNU profiler（gprof）。本文主要面向那些开放源码软件开发工具的新手。gprof 来救援了

在开始介绍如何使用 gprof 之前，需要首先了解一下在整个开发周期中，剖析应该在何处进行。通常来说，编写代码应该有 3 个目标，按照重要性的次序分别如下所示：保证软件可以正确地工作。这通常是开发过程的重点。通常，如果一个软件根本连我们期望它做的事情都实现不了，那么即使它运行速度非常快，也根本没有任何意义！显然，正确性在某些情况下可能并不是至关重要的；例如，如果一个电影播放器可以正确地播放 99% 的电影文件，但是偶尔会有些显示问题，那它依然可以使用。但是通常来说，正确性要远远比速度更加重要。保证软件是可维护的。这实际上是第一个目标的一个子项。通常，如果软件编写得可维护性不好，那么即使它最开始时可以很好地工作，很快您（或其他人）在修正 bug 或添加新特性时可能也会破坏程序的正确性。让软件可以快速运行。这就是剖析的用武之地。当软件可以正确运行

之后，我们就可以开始剖析的过程来帮助它更快地运行了。假设我们现在已经有了一个可以工作的应用程序，接下来让我们来看一下如何使用 gprof 来精确测量应用程序执行过程中时间都花费到什么地方去了，这样做的目的是了解一下在什么地方进行优化效果最佳。gprof 可以对 C、C++、Pascal 和 Fortran 77 应用程序进行剖析。本文中的例子使用的是 C。

1. 耗时的应用程序示例

```
#include <stdio.h>
int a(void) { int i=0,g=0. while(i { g=i. } return g.}
int b(void) { int i=0,g=0. while(i { g=i. } return g.}
int main(int argc, char** argv){ int iterations. if(argc != 2) { printf("Usage %s \n", argv[0]). exit(-1). } else iterations = atoi(argv[1]). printf("No of iterations = %d\n", iterations). while(iterations--) { a(). b(). }}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)