



# 嵌入式Linux和Android培训

家

## 使用 procrank 测量嵌入式 Linux 上的内存使用情况

星期五, 08/14/2015 - 17:46 — csimmonds

程序员会问的一个基本问题（或者至少他们应该问）是：我的程序使用了多少内存？这可能是一个简单的问题，但对于像 Linux 这样的虚拟内存操作系统，答案就相当复杂了。top 和 ps 给出的数字实际上并不相符。Procrank是Android 平台开发人员常用的工具，可以提供更准确的答案，但没有理由不能在其他基于 Linux 的操作系统中更广泛地使用它，它对嵌入式 Linux 尤其有用。

## 代码

您可以从 Github 获取代码：[https://github.com/csimmonds/procrank\\_linux.git](https://github.com/csimmonds/procrank_linux.git)

README 中有关于构建和使用的说明

## 理论

进程内存使用情况的两个最常见指标是虚拟集大小 Vss 和驻留集大小 Rss：您将在 ps 和 top 中看到这些数字。

**Vss**，也称为 VIRT 和 VSZ，是进程已映射的虚拟内存总量，无论它是否已提交到物理内存

**Rss**，也称为 RES 和 RSS，是映射的物理内存量

Vss 显然是高估了，因为应用程序通常会分配它们从不使用的内存。Rss 是一种更好的衡量标准，但仍然高估了，因为它没有考虑进程之间共享的内存页面。例如，内存中只有一个 C 库副本，因为它在与其链接的所有进程之间共享，但 Rss 多次考虑了它。

几年前，Matt Mackall 研究了这个问题并添加了两个新的指标，分别称为唯一集大小 (Uss) 和比例集大小 (Pss)，并且修改了内核以公开计算它们所需的信息，您可以在 /proc/[PID]/smaps 中找到这些信息。

**Uss**是唯一集大小，即进程私有的内存量，不与任何其他进程共享

**Pss**是比例集大小，即与其他进程共享的内存量除以共享每个页面的进程数

为了稍微简化一下，下图显示了三个进程，每个进程的页面都映射到其虚拟地址空间中。这些页面被标记为 A、B 或 C 类型，其中：

A = 映射到 RAM 物理页面的私有内存。这将包括正在使用的堆栈和堆部分

B = 已映射并由一个或多个其他进程共享的共享内存，例如共享库中的代码

C = 已分配但从未触及的内存

对于每个过程来说：

$$Vss = A + B + C$$

$$Rss = A + B$$

$$Uss=A$$

$$Pss = A + B/n, \text{ 其中 } n \text{ 是共享的进程数}$$



计算这三个过程的 Pss 得出：

$$Pss(1) = 2 + 3/3 + 2/2 = 4$$

$$Pss(2) = 2 + 3/3 + 2/2 = 4$$

$$Pss(3) = 2 + 3/3 = 3$$

$$\text{Sum}(Pss) = 11 = \text{使用中的页面总数}$$

如您所见，Pss 可以准确衡量进程使用的内存，同时考虑到进程之间的共享。所有进程使用的内存总量是它们的 Pss 之和。

Uss 也很有用，因为它显示了唯一的页面。您可以将其视为如果您分叉该进程以创建副本，则在内存中需要付出的代价。

有一个现成的程序可以显示 Uss 和 Pss，名为 **smem**，由 Matt Mackall 开发。它的唯一问题是需要 Python 运行时环境，而嵌入式 Linux 设备上通常没有这种环境。Android 开发人员遇到了这个问题，他们使用 Android BIONIC C 库用 C 编写了 **procrank** 作为命令行工具。我采用了该代码，做了一些小改动，并添加了 Makefile，以便它可以在大多数 GNU/Linux 环境中编译，包括用于嵌入式使用的交叉编译。

---

版权所有 © 1999-2023 2net Limited。保留所有权利