**1 free命令输出解释**

　　下面是free的运行结果，一共有4行，为了方便说明对其加上了列号，这样可以把free的输出看成一个二维数组FO(Free Output)，例如：

* FO[2][1] = 24677460
* FO[3][2] = 10321516

          1          2          3          4          5          6

1        **total**       **used**       **free**     **shared**    **buffers**     **cached**

2 **Mem:** 24677460   23276064   1401396          0     870540   12084008  
3 **-/+ buffers/cache:**   10321516   14355944  
4 **Swap:**    25151484     224188   24927296

　　free的输出一共有四行，**第四行**为交换区的信息，分别是交换的总量（total），使用量（used）和有多少空闲的交换区（free）。

　　free输出地**第二行和第三行**是比较让人迷惑的，这两行都是说明内存使用情况的。第一列是总量（total），第二列是使用量（used），第三列是可用量（free）。

**第二行的输出是从操作系统（OS）来看**的，也就是说，从OS的角度来看，计算机上一共有:

* 24677460KB（缺省时free的单位为KB）物理内存，即FO[2][1]；
* 在这些物理内存中有23276064KB（即FO[2][2]）被使用了；
* 还用1401396KB（即FO[2][3]）是可用的；

这里得到第一个**等式**：

* FO[2][1] = FO[2][2] + FO[2][3]

FO[2][4]表示被几个进程共享的内存的，现在已经deprecated，其值总是0（当然在一些系统上也可能不是0，主要取决于free命令是怎么实现的）。

FO[2][5]表示被OS buffer住的内存，FO[2][6]表示被OS cache的内存。在有些时候buffer和cache这两个词经常混用，不过在一些比较低层的软件里是要区分这两个词的，看老外的洋文:

* *A buffer is something that has yet to be "written" to disk.*
* *A cache is something that has been "read" from the disk and stored for later use.*

也就是说buffer是用于存放要输出到disk（块设备）的数据的，而cache是存放从disk上读出的数据。这二者是为了提高IO性能的，并由OS管理。

Linux和其他成熟的操作系统（例如windows），为了提高IO read的性能，总是要多cache一些数据，这也就是为什么FO[2][6]（cached memory）比较大，而FO[2][3]比较小的原因。我们可以做一个**简单的测试**:

1. 释放掉被系统cache占用的数据；

echo 3>/proc/sys/vm/drop\_caches

1. 读一个大文件，并记录时间；
2. 关闭该文件；
3. 重读这个大文件，并记录时间；

第二次读应该比第一次快很多，原来我做过一个BerkeleyDB的读操作，大概要读5G的文件，几千万条记录，在我的环境上，第二次读比第一次大概可以快9倍左右。

**free输出的第三行是从一个应用程序的角度**看系统内存的使用情况。

* 对于FO[3][2]，即-buffers/cache，表示一个应用程序认为系统被用掉多少内存；
* 对于FO[3][3]，即+buffers/cache，表示一个应用程序认为系统还有多少内存；

因为被系统cache和buffer占用的内存可以被快速回收，所以通常FO[3][3]比FO[2][3]会大很多。

这里还用两个等式：

* FO[3][2] = FO[2][2] - FO[2][5] - FO[2][6]
* FO[3][3] = FO[2][3] + FO[2][5] + FO[2][6]

这二者都不难理解。

free命令由procps.\*.rpm提供（在Redhat系列的OS上）。

free命令的所有输出值都是从/proc/meminfo中读出的。

在系统上可能有meminfo(2)这个函数，它就是为了解析/proc/meminfo的，procps这个包自己实现了meminfo()这个函数。可以下载一个procps的tar包看看具体实现。