[内核中的 /proc/meminfo](http://blog.csdn.net/jackjones_008/article/details/41944973)

标签： [linux](http://www.csdn.net/tag/linux)[内存](http://www.csdn.net/tag/%e5%86%85%e5%ad%98)

2014-12-15 16:24 686人阅读 [评论](http://blog.csdn.net/jackjones_008/article/details/41944973#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/jackjones_008/article/details/41944973#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Linux驱动（24） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

**来自 http://leave001.blog.163.com/blog/static/1626912932010111101124590/**

**红字部分为自己找的**

**https://www.centos.org/docs/5/html/5.1/Deployment\_Guide/s2-proc-meminfo.html**

1. **Active = Active(anon) + Active(file) Inactive = Inactive(anon) + Inactive(file)**
2. **Active(file), Inactive(file) has file back-end which means its original file is in disk but to use it faster it was loaded on RAM.**
3. **Active(file) + Inactive(file) + Shmem = Cached + Buffer + SwapCached**

先看Documentation/filesystem/proc.txt中对meminfo的介绍：

meminfo:

Provides information about distribution and utilization of memory.  This

varies by architecture and compile options.  The following is from a

16GB PIII, which has highmem enabled.  You may not have all of these fields.

> cat /proc/meminfo

MemTotal:     16344972 kB

MemFree:      13634064 kB

Buffers:          3656 kB

Cached:        1195708 kB

SwapCached:          0 kB

Active:         891636 kB

Inactive:      1077224 kB

HighTotal:    15597528 kB

HighFree:     13629632 kB

LowTotal:       747444 kB

LowFree:          4432 kB

SwapTotal:           0 kB

SwapFree:            0 kB

Dirty:             968 kB

Writeback:           0 kB

AnonPages:      861800 kB

Mapped:         280372 kB

Slab:           284364 kB

SReclaimable:   159856 kB

SUnreclaim:     124508 kB

PageTables:      24448 kB

NFS\_Unstable:        0 kB

Bounce:              0 kB

WritebackTmp:        0 kB

CommitLimit:   7669796 kB

Committed\_AS:   100056 kB

VmallocTotal:   112216 kB

VmallocUsed:       428 kB

VmallocChunk:   111088 kB

    MemTotal: Total usable ram (i.e. physical ram minus a few reserved

              bits and the kernel binary code)

     MemFree: The sum of LowFree+HighFree

     Buffers: Relatively temporary storage for raw disk blocks

              shouldn't get tremendously large (20MB or so)

      Cached: in-memory cache for files read from the disk (the

              pagecache).  Doesn't include SwapCached

  SwapCached: Memory that once was swapped out, is swapped back in but

              still also is in the swapfile (if memory is needed it

              doesn't need to be swapped out AGAIN because it is already

              in the swapfile. This saves I/O)

      Active: Memory that has been used more recently and usually not

              reclaimed unless absolutely necessary.

    Inactive: Memory which has been less recently used.  It is more

              eligible to be reclaimed for other purposes

   HighTotal:

    HighFree: Highmem is all memory above ~860MB of physical memory

              Highmem areas are for use by userspace programs, or

              for the pagecache.  The kernel must use tricks to access

              this memory, making it slower to access than lowmem.

    LowTotal:

     LowFree: Lowmem is memory which can be used for everything that

              highmem can be used for, but it is also available for the

              kernel's use for its own data structures.  Among many

              other things, it is where everything from the Slab is

              allocated.  Bad things happen when you're out of lowmem.

   SwapTotal: total amount of swap space available

    SwapFree: Memory which has been evicted from RAM, and is temporarily

              on the disk

       Dirty: Memory which is waiting to get written back to the disk

   Writeback: Memory which is actively being written back to the disk

   AnonPages: Non-file backed pages mapped into userspace page tables

      Mapped: files which have been mmaped, such as libraries

        Slab: in-kernel data structures cache

SReclaimable: Part of Slab, that might be reclaimed, such as caches

  SUnreclaim: Part of Slab, that cannot be reclaimed on memory pressure

  PageTables: amount of memory dedicated to the lowest level of page

              tables.

NFS\_Unstable: NFS pages sent to the server, but not yet committed to stable

     storage

      Bounce: Memory used for block device "bounce buffers"

WritebackTmp: Memory used by FUSE for temporary writeback buffers

 CommitLimit: Based on the overcommit ratio ('vm.overcommit\_ratio'),

              this is the total amount of  memory currently available to

              be allocated on the system. This limit is only adhered to

              if strict overcommit accounting is enabled (mode 2 in

              'vm.overcommit\_memory').

              The CommitLimit is calculated with the following formula:

              CommitLimit = ('vm.overcommit\_ratio' \* Physical RAM) + Swap

              For example, on a system with 1G of physical RAM and 7G

              of swap with a `vm.overcommit\_ratio` of 30 it would

              yield a CommitLimit of 7.3G.

              For more details, see the memory overcommit documentation

              in vm/overcommit-accounting.

Committed\_AS: The amount of memory presently allocated on the system.

              The committed memory is a sum of all of the memory which

              has been allocated by processes, even if it has not been

              "used" by them as of yet. A process which malloc()'s 1G

              of memory, but only touches 300M of it will only show up

              as using 300M of memory even if it has the address space

              allocated for the entire 1G. This 1G is memory which has

              been "committed" to by the VM and can be used at any time

              by the allocating application. With strict overcommit

              enabled on the system (mode 2 in 'vm.overcommit\_memory'),

              allocations which would exceed the CommitLimit (detailed

              above) will not be permitted. This is useful if one needs

              to guarantee that processes will not fail due to lack of

              memory once that memory has been successfully allocated.

VmallocTotal: total size of vmalloc memory area

 VmallocUsed: amount of vmalloc area which is used

VmallocChunk: largest contiguous block of vmalloc area which is free

根据<http://www.win.tue.nl/~aeb/linux/lk/lk-9.html>上的介绍，我这里使用了两个程序来测试meminfo的变化情况

程序1：

#include <stdio.h>

#include <stdlib.h>

int main (void) {

        int n = 0;

        while (n<160) {

                if (malloc(1<<20) == NULL) {

                        printf("malloc failure after %d MiB\n", n);

                        return 0;

                }

                printf ("got %d MiB\n", ++n);

        }

        while (1)

         sleep(1);

}

gcc 1.c -o 1

生成的可执行文件为1。

程序2：

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main (void) {

        int n = 0;

        char \*p;

        while (n<160) {

                if ((p = malloc(1<<20)) == NULL) {

                        printf("malloc failure after %d MiB\n", n);

                        return 0;

                }

                memset (p, 0, (1<<20));

                printf ("got %d MiB\n", ++n);

        }

        while (1)

         sleep(1);

}

gcc 2.c -o 2

生成的可执行文件为2。1和2的差别在于，1用malloc分配了内存，但并没有使用它，而2使用了。

实验步骤为：

1. 保存meminfo为beforeX.txt（X为1或者2）

2. 执行X

3. 再次保存meminfo为afterX.txt

4. 用ps检查VSS和RSS

实验得出了四个文件：

before1.txt

MemTotal:         254152 kB

MemFree:          176588 kB

Buffers:               0 kB

Cached:             8600 kB

SwapCached:            0 kB

Active:             4320 kB

Inactive:           6100 kB

Active(anon):       1732 kB

Inactive(anon):      120 kB

Active(file):       2588 kB

Inactive(file):     5980 kB

Unevictable:           0 kB

Mlocked:               0 kB

SwapTotal:             0 kB

SwapFree:              0 kB

Dirty:                 0 kB

Writeback:             0 kB

AnonPages:          1848 kB

Mapped:             1196 kB

Shmem:                32 kB

Slab:               2780 kB

SReclaimable:        740 kB

SUnreclaim:         2040 kB

KernelStack:         360 kB

PageTables:          132 kB

NFS\_Unstable:          0 kB

Bounce:                0 kB

WritebackTmp:          0 kB

CommitLimit:      127076 kB

Committed\_AS:       4296 kB

VmallocTotal:     745472 kB

VmallocUsed:       35000 kB

VmallocChunk:     700412 kB

after1.txt:

MemTotal:         254152 kB

MemFree:          175844 kB

Buffers:               0 kB

Cached:             8600 kB

SwapCached:            0 kB

Active:             4984 kB

Inactive:           6100 kB

Active(anon):       2396 kB

Inactive(anon):      120 kB

Active(file):       2588 kB

Inactive(file):     5980 kB

Unevictable:           0 kB

Mlocked:               0 kB

SwapTotal:             0 kB

SwapFree:              0 kB

Dirty:                 0 kB

Writeback:             0 kB

AnonPages:          2512 kB

Mapped:             1348 kB

Shmem:                32 kB

Slab:               2784 kB

SReclaimable:        740 kB

SUnreclaim:         2044 kB

KernelStack:         368 kB

PageTables:          460 kB

NFS\_Unstable:          0 kB

Bounce:                0 kB

WritebackTmp:          0 kB

CommitLimit:      127076 kB

Committed\_AS:     169056 kB

VmallocTotal:     745472 kB

VmallocUsed:       35000 kB

VmallocChunk:     700412 kB

before2.txt:

MemTotal:         254152 kB

MemFree:          181508 kB

Buffers:               0 kB

Cached:             3792 kB

SwapCached:            0 kB

Active:             2920 kB

Inactive:           2680 kB

Active(anon):       1712 kB

Inactive(anon):      128 kB

Active(file):       1208 kB

Inactive(file):     2552 kB

Unevictable:           0 kB

Mlocked:               0 kB

SwapTotal:             0 kB

SwapFree:              0 kB

Dirty:                 0 kB

Writeback:             0 kB

AnonPages:          1836 kB

Mapped:             1180 kB

Shmem:                32 kB

Slab:               2728 kB

SReclaimable:        688 kB

SUnreclaim:         2040 kB

KernelStack:         360 kB

PageTables:          132 kB

NFS\_Unstable:          0 kB

Bounce:                0 kB

WritebackTmp:          0 kB

CommitLimit:      127076 kB

Committed\_AS:       4296 kB

VmallocTotal:     745472 kB

VmallocUsed:       35000 kB

VmallocChunk:     700412 kB

after2.txt:

MemTotal:         254152 kB

MemFree:           16784 kB

Buffers:               0 kB

Cached:             3792 kB

SwapCached:            0 kB

Active:           167320 kB

Inactive:           2784 kB

Active(anon):     166224 kB

Inactive(anon):      120 kB

Active(file):       1096 kB

Inactive(file):     2664 kB

Unevictable:           0 kB

Mlocked:               0 kB

SwapTotal:             0 kB

SwapFree:              0 kB

Dirty:                 0 kB

Writeback:             0 kB

AnonPages:        166340 kB

Mapped:             1324 kB

Shmem:                32 kB

Slab:               2780 kB

SReclaimable:        728 kB

SUnreclaim:         2052 kB

KernelStack:         368 kB

PageTables:          464 kB

NFS\_Unstable:          0 kB

Bounce:                0 kB

WritebackTmp:          0 kB

CommitLimit:      127076 kB

Committed\_AS:     169056 kB

VmallocTotal:     745472 kB

VmallocUsed:       35000 kB

VmallocChunk:     700412 kB

另外，执行1和2时PS情况分别为：

  PID CPU% S  #THR     VSS     RSS UID      Name

  987   0% S     1 165208K    816K root     ./1

  991   0% S     1 165208K 164648K root     ./2

根据以上实验可以得出：

1. 应用层malloc分配的内存统计在AnonPages中

2. 分配的内存没有使用时，不会占用物理内存。这可以从ps中看出来：VSS为分配的总数，RSS为实际占用的物理内存

3. Active = Active(anon) + Active(file)，Inactive = Inactive(anon) + Inactive(file)