5 commands to check memory usage on Linux

Memory Usage

On linux, there are commands for almost everything, because the gui might not be always available. When working on servers only shell access is available and everything has to be done from these commands. So today we shall be checking the commands that can be used to check memory usage on a linux system. Memory include RAM and swap.

It is often important to check memory usage and memory used per process on servers so that resources do not fall short and users are able to access the server. For example a website. If you are running a webserver, then the server must have enough memory to serve the visitors to the site. If not, the site would become very slow or even go down when there is a traffic spike, simply because memory would fall short. Its just like what happens on your desktop PC.

free command

The free command is the most simple and easy to use command to check memory usage on linux. Here is a quick example

```
$ free -m
             total       used       free     shared    buffers     cached
Mem:          7976       6459       1517          0        865       2248
-/+ buffers/cache:       3344       4631
Swap:         1951          0       1951
```

The m option displays all data in MBs. The total os 7976 MB is the total amount of RAM installed on the system, that is 8GB. The used column shows the amount of RAM that has been used by linux, in this case around 6.4 GB. The output is pretty self explanatory. The catch over here is the cached and buffers column. The second line tells that 4.6 GB is free. This is the free memory in first line added with the buffers and cached amount of memory.

Linux has the habit of caching lots of things for faster performance, so that memory can be freed and used if needed.
The last line is the swap memory, which in this case is lying entirely free.

/proc/meminfo

The next way to check memory usage is to read the /proc/meminfo file. Know that the /proc file system does not contain real files. They are rather virtual files that contain dynamic information about the kernel and the system.

```
$ cat /proc/meminfo
MemTotal:        8167848 kB
MemFree:         1409696 kB
Buffers:          961452 kB
Cached:          2347236 kB
SwapCached:            0 kB
Active:          3124752 kB
Inactive:        2781308 kB
Active(anon):    2603376 kB
Inactive(anon):   309056 kB
Active(file):     521376 kB
Inactive(file):  2472252 kB
Unevictable:        5864 kB
Mlocked:            5880 kB
SwapTotal:       1998844 kB
SwapFree:        1998844 kB
Dirty:              7180 kB
Writeback:             0 kB
AnonPages:       2603272 kB
Mapped:           788380 kB
```

```
Shmem:            311596 kB
Slab:             200468 kB
SReclaimable:     151760 kB
SUnreclaim:        48708 kB
KernelStack:        6488 kB
PageTables:        78592 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     6082768 kB
Committed_AS:    9397536 kB
VmallocTotal:   34359738367 kB
VmallocUsed:      420204 kB
VmallocChunk:   34359311104 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:       62464 kB
DirectMap2M:     8316928 kB
```

Check the values of MemTotal, MemFree, Buffers, Cached, SwapTotal, SwapFree.

They indicate same values of memory usage as the free command.

vmstat

The vmstat command with the s option, lays out the memory usage statistics much like the proc command. Here is an example

```
$ vmstat -s
      8167848 K total memory
      7449376 K used memory
      3423872 K active memory
      3140312 K inactive memory
       718472 K free memory
      1154464 K buffer memory
      2422876 K swap cache
      1998844 K total swap
            0 K used swap
      1998844 K free swap
       392650 non-nice user cpu ticks
         8073 nice user cpu ticks
        83959 system cpu ticks
     10448341 idle cpu ticks
        91904 IO-wait cpu ticks
            0 IRQ cpu ticks
         2189 softirq cpu ticks
            0 stolen cpu ticks
      2042603 pages paged in
      2614057 pages paged out
            0 pages swapped in
            0 pages swapped out
     42301605 interrupts
     94581566 CPU context switches
   1382755972 boot time
         8567 forks
$
```

The top few lines indicate total memory, free memory etc and so on.

top command

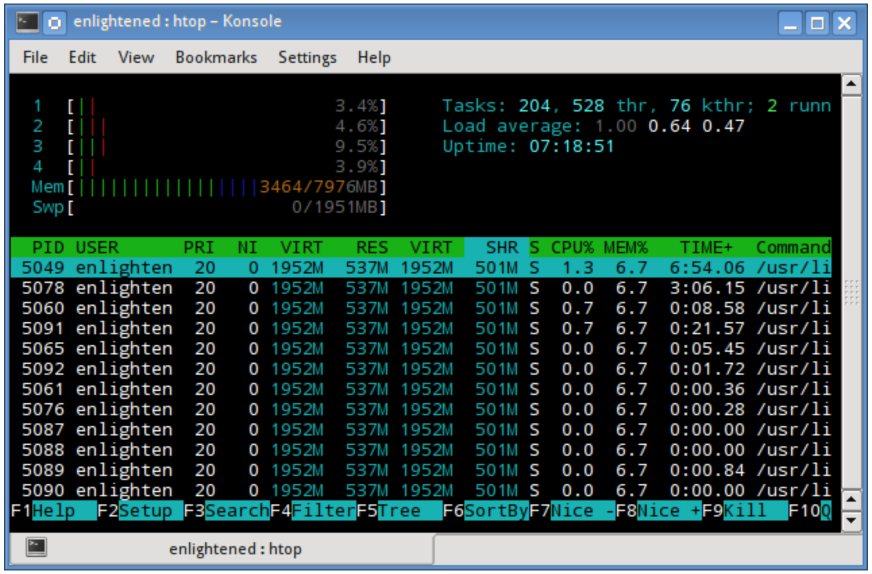
The top command is generally used to check memory and cpu usage per process. However it also reports total memory usage and can be used to monitor the total RAM usage. The header on output has the required information. Here is a sample output

```
top - 15:20:30 up  6:57,  5 users,  load average: 0.64, 0.44, 0.33
Tasks: 265 total,   1 running, 263 sleeping,   0 stopped,   1 zombie
%Cpu(s):  7.8 us,  2.4 sy,  0.0 ni, 88.9 id,  0.9 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   8167848 total,  6642360 used,  1525488 free,  1026876 buffers
KiB Swap:  1998844 total,        0 used,  1998844 free,  2138148 cached
  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2986 enlighte  20   0  584m  42m  26m S  14.3  0.5   0:44.27 yakuake
 1305 root      20   0  448m  68m  39m S   5.0  0.9   3:33.98 Xorg
 7701 enlighte  20   0  424m  17m  10m S   4.0  0.2   0:00.12 kio_thumbnail
```

Check the KiB Mem and KiB Swap lines on the header. They indicate total, used and free amounts of the memory. The buffer and cache information is present here too, like the free command.

htop

Similar to the top command, the htop command also shows memory usage along with various other details.



```
  1  [||                        3.4%]     Tasks: 204, 528 thr, 76 kthr; 2 runn
  2  [|||                       4.6%]     Load average: 1.00 0.64 0.47
  3  [|||                       9.5%]     Uptime: 07:18:51
  4  [||                        3.9%]
  Mem[|||||||||||||||||||3464/7976MB]
  Swp[                     0/1951MB]

  PID USER      PRI  NI  VIRT   RES  VIRT   SHR S CPU% MEM%   TIME+  Command
 5049 enlighten  20   0 1952M  537M 1952M  501M S  1.3  6.7  6:54.06 /usr/li
 5078 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  3:06.15 /usr/li
 5060 enlighten  20   0 1952M  537M 1952M  501M S  0.7  6.7  0:08.58 /usr/li
 5091 enlighten  20   0 1952M  537M 1952M  501M S  0.7  6.7  0:21.57 /usr/li
 5065 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:05.45 /usr/li
 5092 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:01.72 /usr/li
 5061 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.36 /usr/li
 5076 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.28 /usr/li
 5087 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.00 /usr/li
 5088 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.00 /usr/li
 5089 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.84 /usr/li
 5090 enlighten  20   0 1952M  537M 1952M  501M S  0.0  6.7  0:00.00 /usr/li
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Q
```

The header on top shows cpu usage along with RAM and swap usage with the corresponding figures.

RAM Information

To find out hardware information about the installed RAM, use the demidecode command. It reports lots of information about the installed RAM memory.

```
$ sudo dmidecode -t 17
# dmidecode 2.11
SMBIOS 2.4 present.
Handle 0x0015, DMI type 17, 27 bytes
Memory Device
        Array Handle: 0x0014
        Error Information Handle: Not Provided
        Total Width: 64 bits
        Data Width: 64 bits
        Size: 2048 MB
        Form Factor: DIMM
        Set: None
        Locator: J1MY
        Bank Locator: CHAN A DIMM 0
        Type: DDR2
        Type Detail: Synchronous
        Speed: 667 MHz
        Manufacturer: 0xFF00000000000000
        Serial Number: 0xFFFFFFFF
        Asset Tag: Unknown
        Part Number: 0x524D32474235383443412D36344643FFFFFF
```

Provided information includes the size (2048MB), type (DDR2) , speed(667 Mhz) etc.

Summary

All the above mentioned commands work from the terminal and do not have a gui. When working on a desktop with a gui, it is much easier to use a GUI tool with graphical output. The most common tools are gnome-system-monitor on gnome and

ksysguard on KDE. Both provide resource usage information about cpu, ram, swap and network bandwidth in a graphical and easy to understand visual output.

Last Updated On : 26th October 2013
Read more at Binary Tides