Att få information om sitt Linux-system

Följande kommando kan användas för att få information om Linuxsystemet: *vmstat, ps, uptime och top.*

vmstat

vmstat – virtual memory statistics, tillhandahåller information om virtuellt minne, disk access och hur CPU används. Resultatet från första körningen ger ett genomsnitt från uppstart av maskinen. För att få information om eventuella trögheter i system specificera ett intervall.

För cpu-statistik, specificera ett intervall på 2 sekunder, för statistik om hårddiskar specificera intervall på 60 sekunder.vmstat-kommandot räknar *inte* sig själv som en process.

Kommando

```
[root@ /root]# vmstat
```

Producerar följande:

```
procs                  memory    swap        io    system         cpu
 r b w  swpd  free  buff cache  si  so   bi   bo   in   cs  us  sy  id
 0 0 0     0 17280  6660  9356   0   0    0    0  101    3   0   0 100
```

Fält

Procs

r = jobb som ligger och väntar på att köras. Är siffran hög tyder detta på någon form av problem som kan härledas till CPU.

b = indikerar jobb som ligger och väntar på någon resurs. Resurs i detta fallet kan vara hårddisk etc. Om siffran är hög tyder detta på något problem med I/O.

w = antalet jobb som har körts senaste 20 sekunderna och blivit utbytt. Är fältet något annat än 0 kan detta tyda på för litet minne.

Memory

swpd = hur mycket virtuellt minne som används.

free = hur mycket minne som är ledigt (idle).

Växlar

vmstat -n

```
[root@ /root]# vmstat -n
```

Skriv ut rubrik en gång. Producerar följande resultat:

```
procs                  memory    swap        io    system         cpu
 r b w   swpd  free  buff cache  si  so   bi   bo   in   cs  us  sy  id
 0 0 0      0 15992  7684  9476   0   0    0    0  100    3   0   0 100
 0 0 0      0 15992  7684  9476   0   0    0    0  101    3   0   1  99
 0 0 0      0 15992  7684  9476   0   0    0    2  106    4   0   1  99

etc.
```

vmstat [delay [count]]

```
[root@ /root]# vmstat –n 5 6
```

Skapar statistik var femte sekund för sex mätningar. Producerar följande resultat:

```
procs                  memory    swap        io    system         cpu
 r b w   swpd  free  buff cache  si  so   bi   bo   in   cs  us  sy  id
 0 0 0      0 15992  7684  9476   0   0    0    0  100    3   0   0 100
 0 0 0      0 15992  7684  9476   0   0    0    0  101    3   0   1  99
 0 0 0      0 15992  7684  9476   0   0    0    2  106    4   0   1  99
 0 0 0      0 15992  7684  9476   0   0    0    0  102    3   0   1  99
 0 0 0      0 15992  7684  9476   0   0    0    0  102    4   0   1  99
 0 0 0      0 15992  7684  9476   0   0    0    0  101    3   0   1  99
```

vmstat –V

```
[root@ /root]# vmstat –V
```

Rapportera vilken version som används. Ger följande resultat:

```
procps version 1.2.7
```

ps

ps kommandot med växlarna aux och ef ger dig värdefull information om de processer som körs på ditt system.

Kontrollera information om:

- Hur mycket av cpu processen tar, titta i %CPU kolumnen.
- Hur mycket minne processen tar, titta i %MEM kolumnen.
- Status för pågående process, titta i STAT kolumnen.

Identifiera de processer som är de högsta användarna av CPU och/eller minne. Notera att minnes informationen som ges releteras till fysiskt minne, inkluderar inte minne som används av kernel eller instruktions segment för varje process.

Undersök STAT-kolumnen, om en process är i ”RW-state” kan systemet ha problem med minnet, då främst för lite minne.

Kommando

```
[root@ /root]# ps
```

Producerar följande:

```
PID        TTY STAT TIME         COMMAND
  356      1 S    0:00          /bin/login -- root
  357      2 S    0:00          /sbin/mingetty tty2
  358      3 S    0:00          /sbin/mingetty tty3
  359      4 S    0:00          /sbin/mingetty tty4
  360      5 S    0:00          /sbin/mingetty tty5
  361      6 S    0:00          /sbin/mingetty tty6
  364      1 S    0:00          -bash
 1063      1 R    0:00          ps
```

Växlar

ps l

```
[root@ /root]# ps l
```

Producerar följande resultat:

```
FLAGS   UID   PID  PPID PRI  NI   SIZE   RSS WCHAN       STA TTY TIME COMMAND
100100    0   356     1   0   0   1496   780 wait4        S    1  0:00 /bin/logi
100100    0   357     1   0   0    740   316 read_chan    S    2  0:00 /sbin/min
100100    0   358     1   0   0    740   316 read_chan    S    3  0:00 /sbin/min
100100    0   359     1   0   0    740   316 read_chan    S    4  0:00 /sbin/min
100100    0   360     1   0   0    740   316 read_chan    S    5  0:00 /sbin/min
100100    0   361     1   0   0    740   316 read_chan    S    6  0:00 /sbin/min
   100    0   364   356  17   0   1220   804 wait4        S    1  0:00 -bash
100000    0  1068   364  10   0    960   496              R    1  0:00 ps l
```

ps aux

```
[root@ /root]# ps aux
```

Producerar följande resultat:

```
USER       PID %CPU %MEM  SIZE   RSS TTY STAT START   TIME COMMAND
bin        239  0.0  0.8   764   340  ?  S   Aug 16   0:00 portmap
daemon     217  0.0  1.0   792   412  ?  S   Aug 16   0:00 /usr/sbin/atd
nobody     317  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     318  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     319  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     320  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     321  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     322  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     323  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     324  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     325  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
nobody     326  0.0  2.2  1520   876  ?  S   Aug 16   0:00 httpd
root         1  0.0  1.0   776   412  ?  S   Aug 16   0:03 init [3]
root         2  0.0  0.0     0     0  ?  SW  Aug 16   0:00 (kflushd)
root         3  0.0  0.0     0     0  ?  SW< Aug 16   0:00 (kswapd)
root       360  0.0  0.8   740   316   5 S   Aug 16   0:00 /sbin/mingetty tty5
root       361  0.0  0.8   740   316   6 S   Aug 16   0:00 /sbin/mingetty tty6
root       363  0.0  0.6   732   240  ?  S   Aug 16   0:00 update (bdflush)
root       364  0.0  2.0  1220   804   1 S   Aug 16   0:00 -bash
root      1066  0.0  1.2   864   496   1 R    11:58   0:00 ps aux
```

uptime

uptime kommandot ger dig värdefull information om:

- Hur länge systemet har varit uppe.
- Antalet användare i ditt system.
- Hur belastningen har varit senaste minuten, fem minuterna eller femton minuterna.

Kernel håller räkning på en genomsnitt av aktiva jobb för den senaste minuten, fem och femton minuterna.

Kommando

```
[root@ /root]# uptime
```

Producerar följande:

```
12:26pm  up 4 days, 17:08,  1 user,  load average: 0.00, 0.00, 0.00
```

top

top är ett public domain program som visar och uppdaterar information om de 15 ”största” processerna på systemet.

Kommando

```
[root@ /root]# top
```

Producerar följande:

```
12:29pm  up 4 days, 17:12,  1 user,  load average: 0.00, 0.00, 0.00
38 processes: 37 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  5.5% user,  1.8% system,  0.0% nice, 92.7% idle
Mem:   38916K av,  26772K used,  12144K free,  18128K shrd,   9924K buff
Swap: 122888K av,      0K used, 122888K free  10680K cached
```

```
PID USER     PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM   TIME COMMAND
 1108 root      10   0   724  724   560 R       0  7.3  1.8   0:00 top
    1 root       0   0   412  412   344 S       0  0.0  1.0   0:03 init
    2 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kflushd
    3 root     -12 -12     0    0     0 SW<     0  0.0  0.0   0:00 kswapd
    4 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 md_thread
    5 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 md_thread
  364 root      10   0   804  804   628 S       0  0.0  2.0   0:00 bash
  356 root       0   0   780  780   568 S       0  0.0  2.0   0:00 login
  357 root       0   0   316  316   260 S       0  0.0  0.8   0:00 mingetty
   36 root       0   0   360  360   308 S       0  0.0  0.9   0:00 kerneld
  197 root       0   0   460  460   380 S       0  0.0  1.1   0:00 syslogd
  206 root       0   0   544  544   332 S       0  0.0  1.3   0:00 klogd
  217 daemon     0   0   412  412   336 S       0  0.0  1.0   0:00 atd
  228 root       0   0   484  484   404 S       0  0.0  1.2   0:00 crond
  239 bin        0   0   340  340   276 S       0  0.0  0.8   0:00 portmap
  250 root       0   0   772  772   448 S       0  0.0  1.9   0:00 snmpd
  262 root       0   0   420  420   344 S       0  0.0  1.0   0:00 inetd
  273 root       0   0   644  644   512 S       0  0.0  1.6   0:00 named
```