Sledování systému

„Máš dozor na chodbě!“
Spongebob Squarepants
(animovaný seriál)

Jedním z nejdůležitějších úkolů správce systému je sledování systému. Jako správce musíte být schopen zjistit, co se v dané chvíli v systému děje. Může to být využívání systémových prostředků, které příkazy se provádějí, kdo je přihlášený. V této kapitole si řekneme o tom, jak se monitoruje systém, a v některých případech i o řešení problémů, které nastaly.

Při zvyšování výkonu systému je nutno uvažovat o čtyřech hlavních oblastech: základní jednotka, paměť, disky a síť. Umíte-li v systému nalézt kritické místo, ušetříte spoustu času.

Systémové prostředky

Nejdůležitější je sledování výkonu počítače. Nedostatek systémových prostředků může způsobit značné problémy. Mohou je využívat jak uživatelé, tak i služby spojené s činností systému jako elektronická pošta nebo internetové stránky. Když víte, co se v systému děje, snáze můžete rozhodovat o tom, zda je nutno posílit systém nebo jestli by bylo lepší přesunout některé služby na jiný počítač.

Příkaz top

Základním příkazem pro monitorování systému je příkaz **top**, který kontinuálně vypisuje aktuální hlášení o využívání systémových prostředků.

```
# top
 12:10:49  up 1 day,  3:47,  7 users,  load average: 0.23, 0.19, 0.10
125 processes: 105 sleeping, 2 running, 18 zombie, 0 stopped
CPU states:   5.1% user   1.1% system   0.0% nice   0.0% iowait  93.6% idle
Mem:   512716k av,  506176k used,    6540k free,       0k shrd,   21888k buff
Swap: 1044216k av,  161672k used,  882544k free                  199388k cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME CPU COMMAND
 2330 root      15   0  161M  70M  2132 S     4.9 14.0  1000m   0 X
 2605 weeksa    15   0  8240 6340  3804 S     0.3  1.2   1:12   0 kdeinit
 3413 weeksa    15   0  6668 5324  3216 R     0.3  1.0   0:20   0 kdeinit
18734 root      15   0  1192 1192   868 R     0.3  0.2   0:00   0 top
 1619 root      15   0   776  608   504 S     0.1  0.1   0:53   0 dhclient
    1 root      15   0   480  448   424 S     0.0  0.0   0:03   0 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00   0 keventd
```

```
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00   0 kapmd
    4 root      35  19     0    0     0 SWN   0.0  0.0   0:00   0 ksoftirqd_CPU0
    9 root      25   0     0    0     0 SW    0.0  0.0   0:00   0 bdflush
    5 root      15   0     0    0     0 SW    0.0  0.0   0:00   0 kswapd
   10 root      15   0     0    0     0 SW    0.0  0.0   0:00   0 kupdated
   11 root      25   0     0    0     0 SW    0.0  0.0   0:00   0 mdrecoveryd
   15 root      15   0     0    0     0 SW    0.0  0.0   0:01   0 kjournald
   81 root      25   0     0    0     0 SW    0.0  0.0   0:00   0 khubd
 1188 root      15   0     0    0     0 SW    0.0  0.0   0:00   0 kjournald
 1675 root      15   0   604  572   520 S     0.0  0.1   0:00   0 syslogd
 1679 root      15   0   428  376   372 S     0.0  0.0   0:00   0 klogd


 1813 root      25   0   752  528   524 S     0.0  0.1   0:00   0 sshd
 1828 root      25   0   704  548   544 S     0.0  0.1   0:00   0 xinetd

<zkráceno>
```

V horní části hlášení jsou informace o systémovém čase, době bezporuchového chodu, využívání základní jednotky, využívání fyzické a odkládací paměti a o počtu procesů. Pod ní je seznam procesů setříděných podle doby využívání základní jednotky počítače.

Výstup příkazu top můžete v průběhu činnosti modifikovat. Když stisknete klávesu **i** (jako idle), program přestane vypisovat nečinné procesy. Opětovným stisknutím **i** výpis obnovíte. Stisknutím klávesy **M** setřídíte procesy podle toho, jak využívají paměť, klávesou **S** podle délky běhu procesu a klávesou **P** se vrátíte k původnímu třídění podle doby využívání základní jednotky počítače.

Kromě volby způsobu zobrazení můžete procesy příkazem **top** také modifikovat. Pomocí **u** můžete prohlížet procesy, které náleží určitému uživateli, pomocí **k** můžete procesy rušit a pomocí **r** můžete měnit parametry procesů.

Podrobnější informace o procesech naleznete v souborovém systému `/proc`, který obsahuje řadu podadresářů s číselnými jmény. Vztahují se k ID běžících procesů a naleznete v nich soubory s informacemi o těchto procesech.

TYTO SOUBORY NESMÍTE V ŽÁDNÉM PŘÍPADĚ ZMĚNIT – MOHLO BY DOJÍT K POŠKOZENÍ SYSTÉMU!

Příkaz iostat

Příkaz **iostat** vypisuje průběžné zatížení základní jednotky a informace o V/V. Je to vynikající příkaz pro sledování využití disků.

```
# iostat
Linux 2.4.20-24.9 (myhost)       12/23/2003

avg-cpu:  %user   %nice    %sys   %idle
          62.09    0.32    2.97   34.62

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev3-0            2.22        15.20        47.16    1546846    4799520
```

V jádru 2.4 a 2.6 vypisuje tento příkaz hlavní a vedlejší číslo zařízení. V tomto případě je to `/dev/hda`. V příkazu **iostat** zadejte volbu **-x**.

```
# iostat -x
Linux 2.4.20-24.9 (myhost)       12/23/2003

avg-cpu:  %user   %nice    %sys   %idle
          62.01    0.32    2.97   34.71

Device:  rrqm/s wrqm/s r/s  w/s rsec/s wsec/s rkB/s wkB/s avgrq-sz avgqu-sz
await svctm %util
/dev/hdc   0.00   0.00 .00 0.00   0.00   0.00  0.00  0.00     0.00     2.35
0.00  0.00 14.71
/dev/hda   1.13   4.50 .81 1.39  15.18  47.14  7.59 23.57    28.24     1.99
63.76 70.48 15.56
/dev/hda1  1.08   3.98 .73 1.27  14.49  42.05  7.25 21.02    28.22     0.44
21.82  4.97  1.00
/dev/hda2  0.00   0.51 .07 0.12   0.55   5.07  0.27  2.54    30.35     0.97
52.67 61.73  2.99
/dev/hda3  0.05   0.01 .02 0.00   0.14   0.02  0.07  0.01     8.51     0.00
12.55  2.95  0.01
```

Význam jednotlivých sloupců je popsán v manuálových stránkách **iostat**.

Příkaz ps

Seznam běžících procesů. Příkaz má mnoho různých voleb.

Příkazem obvykle vypisujeme všechny běžící procesy. Použijte volbu **ps -ef**. (Celkový výpis je velmi dlouhý, je tedy uvedena jenom část.)

```
UID         PID  PPID  C STIME TTY          TIME CMD
root          1     0  0 Dec22 ?        00:00:03 init
root          2     1  0 Dec22 ?        00:00:00 [keventd]
root          3     1  0 Dec22 ?        00:00:00 [kapmd]
root          4     1  0 Dec22 ?        00:00:00 [ksoftirqd_CPU0]
root          9     1  0 Dec22 ?        00:00:00 [bdflush]
root          5     1  0 Dec22 ?        00:00:00 [kswapd]
root          6     1  0 Dec22 ?        00:00:00 [kscand/DMA]
root          7     1  0 Dec22 ?        00:01:28 [kscand/Normal]
root          8     1  0 Dec22 ?        00:00:00 [kscand/HighMem]
root         10     1  0 Dec22 ?        00:00:00 [kupdated]
root         11     1  0 Dec22 ?        00:00:00 [mdrecoveryd]
root         15     1  0 Dec22 ?        00:00:01 [kjournald]
root         81     1  0 Dec22 ?        00:00:00 [khubd]
root       1188     1  0 Dec22 ?        00:00:00 [kjournald]
root       1675     1  0 Dec22 ?        00:00:00 syslogd -m 0
root       1679     1  0 Dec22 ?        00:00:00 klogd -x
rpc        1707     1  0 Dec22 ?        00:00:00 portmap
root       1813     1  0 Dec22 ?        00:00:00 /usr/sbin/sshd
ntp        1847     1  0 Dec22 ?        00:00:00 ntpd -U ntp
root       1930     1  0 Dec22 ?        00:00:00 rpc.rquotad
root       1934     1  0 Dec22 ?        00:00:00 [nfsd]
root       1942     1  0 Dec22 ?        00:00:00 [lockd]
root       1943     1  0 Dec22 ?        00:00:00 [rpciod]
root       1949     1  0 Dec22 ?        00:00:00 rpc.mountd
root       1961     1  0 Dec22 ?        00:00:00 /usr/sbin/vsftpd
/etc/vsftpd/vsftpd.conf
```

```
root      2057     1  0 Dec22 ?        00:00:00 /usr/bin/spamd -d -c -a

bin       2076     1  0 Dec22 ?        00:00:00 /usr/sbin/cannaserver -syslog -u bin
root      2087     1  0 Dec22 ?        00:00:00 crond
<zkráceno>
```

V prvním sloupci je uveden vlastník procesu, v druhém sloupci je ID procesu. Ve třetím sloupci je ID rodiče, což je proces, který daný proces vytvořil nebo spustil. Ve čtvrtém sloupci je využití základní jednotky v procentech. Pátý sloupec obsahuje dobu spuštění procesu, případně i datum, běží-li proces dlouho. V šestém sloupci je tty, které náleží k danému procesu, pokud existuje. Sedmý sloupec obsahuje souhrnný čas využití základní jednotky (celkový čas základní jednotky spotřebovaný daným procesem). V sedmém sloupci je příkaz samotný.

S využitím těchto informací je zřejmé, co se v systému děje. Bezprizorní procesy a procesy, které způsobují problémy, můžete ukončit.

Příkaz vmstat

Příkaz **vmstat** poskytuje hlášení, které obsahuje statistiku systémových procesů, paměti, odkládání, V/V a základní jednotky. Statistika se vytváří z dat od posledního zadání tohoto příkazu do přítomnosti. Nebyl-li příkaz ještě použit, berou se data od spuštění systému.

```
# vmstat
   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 181604  17000  26296 201120    0    2     8    24  149     9 61  3 36
```

Následující popis polí je z manuálových stránek **vmstat:**

Procs

r: Počet procesů čekajících na spuštění.

b: Počet procesů v nepřerušitelném spánku.

w: Počet odložených běžících procesů. Linux nikdy neodkládá procesy bezdůvodně.

Memory

swpd: Velikost použité virtuální paměti (kB).
free: Velikost nevyužité paměti (kB).
buff: Velikost paměti využívané pro buffery (kB).
cache: Velikost paměti využívané pro cache (kB).

Swap

si: Velikost paměti přenášené z disku (kB/s).
so: Velikost paměti přenášené na disk (kB/s).

IO

bi: Bloky posílané na blokové zařízení (blocks/s).
bo: Bloky čtené z blokového zařízení (blocks/s).

System

in: Počet přerušení za vteřinu včetně přerušení od hodin.
cs: Počet souvisejících přepnutí za vteřinu.

CPU
: Doba využití základní jednotky v procentech.
us: uživatel
sy: systém
id: běh naprázdno

Příkaz lsof

Příkaz **lsof** vypíše seznam všech právě používaných souborů. Vzhledem k tomu, že Linux považuje za soubor všechno, může být seznam velmi dlouhý. Při diagnostických problémech je však tento příkaz velmi užitečný. Za příklad může posloužit situace, kdy chcete odpojit souborový systém, avšak nelze to provést kvůli tomu, že jej někdo používá. Pomocí tohoto příkazu a pomocí příkazu **grep** zjistíte, kdo tento soubor používá.

Anebo předpokládejte, že chcete vidět všechny soubory používané určitým procesem. V takovém případě stačí zadat příkaz **lsof -p -processid-**.

Kde najdete další nástroje

Více se o řádkových nástrojích dočtete v referenční příručce, kterou sepsal Chris Karakas a najdete ji na adrese GNU/Linux Command-Line Tools Summary (*http://www.karakas-online.de/gnu-linux-tools-summary/*). Je vhodným zdrojem pro vyhledání dalších nástrojů a návodů k jejich používání.

Souborový systém

Neustále se dočítáme, jak je diskový prostor levný, avšak převážná část uživatelů s tím nemůže souhlasit. Většina z nás se neustále potýká s nedostatkem místa na disku. Potřebovali bychom sledovat a řídit jeho využívání.

Příkaz df

Příkaz **df** je nejjednodušším nástrojem, kterým můžeme sledovat využití disku. Zadáte pouze **df** a vypíše se vám využití všech připojených disků v blocích o velikosti 1 kB.

```
user@server:~> df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda3              5242904    759692   4483212  15% /
tmpfs                   127876         8    127868   1% /dev/shm
/dev/hda1               127351     33047     87729  28% /boot
/dev/hda9             10485816     33508  10452308   1% /home
/dev/hda8              5242904    932468   4310436  18% /srv
/dev/hda7              3145816     32964   3112852   2% /tmp
/dev/hda5              5160416    474336   4423928  10% /usr
/dev/hda6              3145816    412132   2733684  14% /var
```

Pomocí volby **-h** dostanete výstup v „čitelnější“ podobě. Velikosti budou uvedeny v kilobajtech, megabajtech nebo gigabajtech v závislosti na velikosti souborového systému. Velikost bloku zadáte volbou **-B**.

Kromě využití prostoru na disku si pomocí volby **-i** můžete také zobrazit počet použitých a volných čísel inod.

```
user@server:~> df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/hda3                  0       0       0    -  /
tmpfs                  31969       5   31964    1% /dev/shm
/dev/hda1              32912      47   32865    1% /boot
/dev/hda9                  0       0       0    -  /home
/dev/hda8                  0       0       0    -  /srv
/dev/hda7                  0       0       0    -  /tmp
/dev/hda5             656640   26651  629989    5% /usr
/dev/hda6                  0       0       0    -  /var
```

Příkaz du

Teď už víte, kolik místa máte na souborových systémech zabráno, jak ale zjistíte, kde konkrétně data jsou? Například pomocí příkazu **du**. Pokud neuvedete jméno souboru, příkaz **du** bude fungovat rekurzivně. Například:

```
user@server:~> du file.txt
1300    file.txt
```

Nebo jako u příkazu **df** můžete použít volbu **-h** a výstup se vypíše v čitelnější podobě.

```
user@server:~> du -h file.txt
1.3M     file.txt
```

Pokud neuvedete jméno souboru, příkaz **du** bude fungovat rekurzivně.

```
user@server:~> du -h /usr/local
4.0K    /usr/local/games
16K     /usr/local/include/nessus/net
180K    /usr/local/include/nessus
208K    /usr/local/include
62M     /usr/local/lib/nessus/plugins/.desc
97M     /usr/local/lib/nessus/plugins
164K    /usr/local/lib/nessus/plugins_factory
<zkráceno>
```

Chcete-li vypsat pouze souhrnné číslo pro daný adresář, provedete to pomocí volby **-s**.

```
user@server:~> du -hs /usr/local 210M    /usr/local
```

Quotas

Informace o diskových kvótách naleznete v The Quota HOWTO na adrese *http://www.tldp.org/HOWTO/Quota.html*.

Monitorování uživatelů

To, že jste paranoik, neznamená, že vás nedostanou…
Neznámý autor

Čas od času vás začne zajímat, co provádějí uživatelé v systému. Zjistíte třeba, že se nadměrně využívá vyrovnávací paměť RAM nebo základní jednotka. Určitě se budete chtít podívat, kdo je v systému, co provozuje a jaké k tomu využívá prostředky.

Příkaz who

Nejjednodušší způsob zjištění, kdo je v systému, je zadat **who** nebo **w**. Tímto příkazem si vypíšete, kdo je přihlášen do systému a na kterém portu, resp. terminálu.

```
user@server:~>  who
bjones   pts/0        May 23 09:33
wally    pts/3        May 20 11:35
aweeks   pts/1        May 22 11:03
aweeks   pts/2        May 23 15:04
```

Ještě jednou příkaz ps!

V předchozí kapitole jsme viděli, že uživatel aweeks je přihlášený jak na pts/1, tak i na pts/2, avšak co když chceme vědět, co na nich dělá? Zadáme **ps -u aweeks** a dostaneme výstup:

```
user@server:~> ps -u aweeks

20876 pts/1    00:00:00 bash
20904 pts/2    00:00:00 bash
20951 pts/2    00:00:00 ssh
21012 pts/1    00:00:00 ps
```

Vidíme, že tento uživatel provozuje **ps** a **ssh**.

Příkaz w

Ještě jednodušší než příkazy **who** a **ps -u** je příkaz **w**, který vypíše, nejen kdo je přihlášen do systému, ale také jaký příkaz provádí.

```
user@server:~> w
aweeks   :0       09:32   ?xdm?  30:09   0.02s -:0
aweeks   pts/0    09:33    5:49m  0.00s  0.82s kdeinit: kded
aweeks   pts/2    09:35    8.00s  0.55s  0.36s vi sag-0.9.sgml
aweeks   pts/1    15:03   59.00s  0.03s  0.03s /bin/bash
```

Vidíme, že mám spuštěný program **kde**, pracuji na tomto dokumentu :-) a jsem přihlášený ještě na jednom terminálu, který běží naprázdno s vypsaným promptem bash.