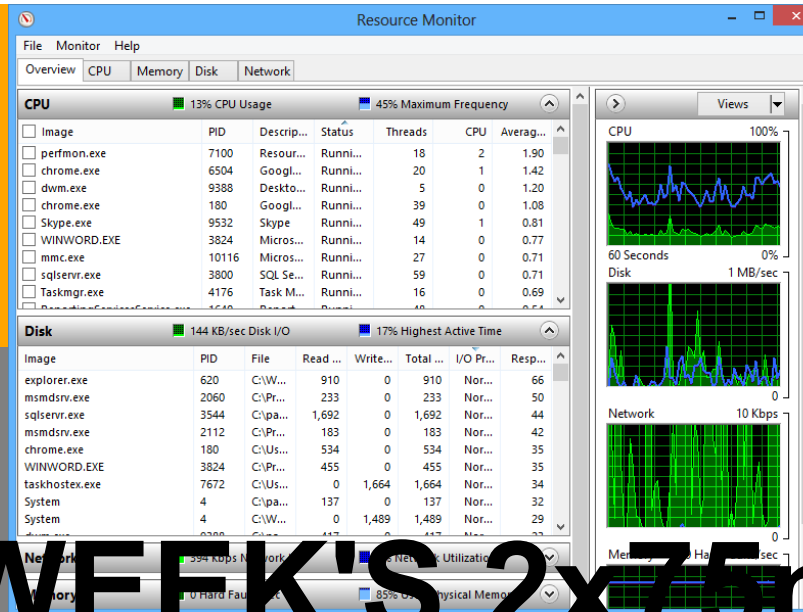


INTRO TO OPERATING SYSTEMS

MONITORING/LOGGING and
PERFORMANCE/CONFIG



THIS WEEK'S 2x75min 4p-5:15p

<https://solutioncenter.apexsql.com/operating-system-os-performance-monitoring/>

- ◆ Booting into Linux (10m)
- ◆ Byobu F9 Toggles (10m)

- ◆ Tour of Linux (40m)
 - ◆ Top, VM, and Threads Demo (40m)
 - ◆ Some Logs (20m)
 - ◆ Discussion of Labs I and II (Carryover) (30m)
-
- ◆ Booting into Linux (10m)

```

[1560389.379584] qbr311fdca5-15: port 1(qvb311fdca5-15) entered forwarding state
[1560389.379596] qbr311fdca5-15: port 1(qvb311fdca5-15) entered forwarding state
[1560389.781022] device tap311fdca5-15 entered promiscuous mode
[1560389.793115] qbr311fdca5-15: port 2(tap311fdca5-15) entered forwarding state
[1560389.793123] qbr311fdca5-15: port 2(tap311fdca5-15) entered forwarding state
[1560394.868530] kvm: zapping shadow pages for mmio generation wraparound
[1560495.686661] sdy: sdy1
[1560495.689054] qbr311fdca5-15: port 2(tap311fdca5-15) entered disabled state
[1560495.689948] device tap311fdca5-15 left promiscuous mode
[1560495.689983] qbr311fdca5-15: port 2(tap311fdca5-15) entered disabled state
[1560496.018536] qbr311fdca5-15: port 1(qvb311fdca5-15) entered disabled state
[1560496.998145] sd 11:0:0:0: [sdy] Synchronizing SCSI cache
[1594642.460622] httpd[33042]: segfault at 8 ip 00007f1ef7166c50 sp 00007f1eee417db0 error 4 in libpython2.7.so.1.0[7f1ef706c000+178000]
[1594642.714637] httpd[26850]: segfault at 8 ip 00007f1ef7166c50 sp 00007f1eee417db0 error 4 in libpython2.7.so.1.0[7f1ef706c000+178000]
[1608828.924794] rx_data returned 0, expecting 48.
[1608828.925904] iSCSI Login negotiation failed.
[1787664.206542] rx_data returned 0, expecting 48.
[1787664.207809] iSCSI Login negotiation failed.
[1800235.169914] rx_data returned 0, expecting 48.
[1800235.171012] iSCSI Login negotiation failed.
[1854895.974079] httpd[63827]: segfault at 8 ip 00007f1ef7166c50 sp 00007f1eee417db0 error 4 in libpython2.7.so.1.0[7f1ef706c000+178000]
[1854896.296884] httpd[57146]: segfault at 8 ip 00007f1ef7166c50 sp 00007f1eee417db0 error 4 in libpython2.7.so.1.0[7f1ef706c000+178000]
[2051175.120536] atal.00: hard resetting link
[2051175.425161] atal.01: hard resetting link
[2051176.431584] atal.01: failed to resume link (SControl 0)
[2051176.582569] atal.00: SATA link up 1.5 Gbps (SStatus 113 SControl 300)
[2051176.582587] atal.01: SATA link down (SStatus 0 SControl 0)
[2051176.582602] atal.01: link offline, clearing class 3 to NONE
[2051176.591695] atal.00: configured for UDMA/100
[2051176.593059] atal: EH complete
[2051857.182429] atal.00: hard resetting link
[2051857.486555] atal.01: hard resetting link
[2051858.493155] atal.01: failed to resume link (SControl 0)
[2051858.644195] atal.00: SATA link up 1.5 Gbps (SStatus 113 SControl 300)
[2051858.644210] atal.01: SATA link down (SStatus 0 SControl 0)
[2051858.644222] atal.01: link offline, clearing class 3 to NONE
[2051858.669167] atal.00: configured for UDMA/100
[2051858.670618] atal: EH complete

```

www.linuxtechi.com

<https://www.linuxtechi.com/10-tips-dmesg-command-linux-geeks/>

◆ Byobu F9 Toggles (10m)

```
Welcome to Ubuntu Natty (development branch) (GNU/Linux 2.6.38-7-generic x86_
64)

* Documentation: https://help.ubuntu.com/

203 packages can be updated.
0 updates are security updates.

*** System restart required ***
archemedes@server:~$
```



```
0*$ shell 1-$ shell archemedes@server 192.168.1.50 Menu:<F9>
U Ubuntu natty ^480kbps v16kbps (R) 203! 6d15h 0.06 4x0.8GHz 3.4GB,20% 201
```

<https://www.howtogeek.com/58487/how-to-easily-multitask-in-a-linux-terminal-with-byobu/>

◆ Tour of Linux (40m)

- [% who](#)
- [% w](#)
- [% date](#)
- [% uname -a](#)
- [% last | tail](#)
- [% sensors](#)
- [% cat /proc/cpuinfo](#)
- [% sudo hdparm -I /dev/sda1](#)
- [% ip addr](#)
- [% cat /proc/devices](#)
- [% cat /proc/interrupts](#)
- [% sudo lshw -short](#)
- [% sudo lshw](#)
- [% df](#)
- [% lsblk](#)
- [% cat /proc/partitions](#)
- [% iostat](#)
- [% vmstat](#)
- [% free -m](#)
- [% cat /proc/meminfo](#)
- [% cat /proc/3185/limits](#)
- [% cat /proc/3185/maps | grep '\['](#)
- [% cat /proc/3185/maps | grep 's'](#)
- [% ls -tl /dev](#)
- [% ls -tl /proc](#)
- [% ls /proc/*/cmdline | awk '{printf \\$0" ";system\("cat " \\$0\);print ""}' | sed 's/\[^a-z0-9\\-\]/ /g' | sed 's/ */ /g' | colrm 70 | sed 's/.proc.//' | sort -n](#)
- [% ls -tl /var/log](#)
- [% find /usr/src/linux-headers-4.4.0-21 | grep 'include.linux.*.h'](#)
- [% lsof | head -1](#)
- [% sudo lsof | grep ' /tmp\\'](#)
- [% lsof | head -1](#)

% who % w % date % uname -a % last | tail % sensors % cat /proc/cpuinfo % sudo hdparm -I /dev/sda1 % ip addr % cat /proc/devices % cat /proc/interrupts % sudo lshw -short % sudo lshw % df % lsblk % cat /proc/partitions % iostat % vmstat % free -m % cat /proc/meminfo % cat /proc/3185/limits % cat /proc/3185/maps | grep '[' % cat /proc/3185/maps | grep 's' % ls -tl /dev % ls -tl /proc % ls /proc/*/cmdline | awk '{printf \$0" ";system("cat " \$0);print ""}' | sed 's/[^a-z0-9\\-]/ /g' | sed 's/ */ /g' | colrm 70 | sed 's/.proc.//' | sort -n

```
";system("cat " $0);print ""}|sed 's/^[a-z0-9\\-]/ /g' | sed 's/ */ /g' | colrm 70 | sed 's/.proc./' | sort -n % ls -tl /var/log % find /usr/src/linux-headers-4.4.0-21 | grep 'include.linux.*\\.h' % lsof | head -1 % sudo lsof | grep '\\tmpV' % lsof | head -1 % lsof -i 4 % nmap 0.0.0.0 % nmap 165.227.104.180 % cat /proc/locks % cat /proc/ioports % cat /proc/iomem % ps -u guest % ps -u root % top -b -n 1 % sudo slabtop -o
```

◆ Top, VM, and Threads Demo (40m)


```

#include <stdio.h>
#include <stdlib.h>
int main() {
    srand(0);
    float *c;
    int n = 100*1000*1000;
    c = malloc(n*sizeof(float));

    int i;
    for (i=0; i<n; i++) c[i] = 1.0;
    // printf("%p ::%c::\n", c,c[1]);
    // printf("%s", "sleeping:"); fflush(stdout);
    // for (i=1; i<=10000; i++) { system("sleep 10"); if (0) printf("%d",i); fflush(stdout); }

    int j;
    for (i=1; i<=10000; i++) {
        for (j=1; j<n; j+=10000) c[i-1] = rand();
        system("sleep 10");
        if (0) printf("%d",i);
        fflush(stdout);
    }
}

```

pig.c:

```

BEGIN {
    # for (i=1; i<=8; i++) system("./pig &")

    mystring = ""
    for (i=1; i<=10; i++) mystring = mystring "0123456789"

    n = 10*1000*1000
    stime = systime(); print "allocating"
    for (i=1;i<=n;i++) sto[i] = mystring
    print "allocated"; print systime()-stime " seconds"

    m = 10*1000*1000
    stime = systime(); print "reading"
    for (t=1; t<=m; t++) {
        from = int(rand()*m)+1; into = int(rand()*m)+1
        sto[into] = sto[from]
    }
    print "read"; print systime()-stime " seconds"
}

```

```
# cleanup any pigs
while ("ps -u guest" | getline)
  if ($NF == "pig")
    system("kill -9 " $1)
```

◆ Tour of Logs (20m)

- pigs in memory, run test.awk
- pigs swapping, run test.awk
- TOP: f (-PR,NI,S,+PPID,nTH,+CODE,DATA,nMin), t, t, m, m, d.5, z, x, H, >, 0, 1
- Chromium/mlb.com

Combined Log Format

Another commonly used format string is called the Combined Log Format. It can be used as follows.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
CustomLog log/access_log combined
```

This format is exactly the same as the Common Log Format, with the addition of two more fields. Each of the additional fields uses the percent-directive `%{header}i`, where *header* can be any HTTP request header. The access log under this format will look like:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
"http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"
```

The additional fields are:

"http://www.example.com/start.html" ("%{Referer}i")

The "Referer" (sic) HTTP request header. This gives the site that the client reports having been referred from. (This should be the page that links to or includes /apache_pb.gif).

"Mozilla/4.08 [en] (Win98; I ;Nav)" ("%{User-agent}i")

The User-Agent HTTP request header. This is the identifying information that the client browser reports about itself.

Multiple Access Logs

Multiple access logs can be created simply by specifying multiple [CustomLog](#) directives in the configuration file. For example, the following directives will create three access logs. The first contains the basic CLF information, while the second and third contain referer and browser information. The last two [CustomLog](#) lines show how to mimic the effects of the ReferLog and AgentLog directives.

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
CustomLog logs/referer_log "%{Referer}i -> %U"
CustomLog logs/agent_log "%{User-agent}i"
```

This example also shows that it is not necessary to define a nickname with the [LogFormat](#) directive. Instead, the log format can be specified directly in the [CustomLog](#) directive.

<https://httpd.apache.org/docs/2.4/logs.html>

◆ Tour of Logs (20m)

- [% /var/log# ls](#)
- [% /var/log# head auth.log](#)
- [% /var/log# grep 'invalid user.*from' auth.log | tail](#)
- [% /var/log# grep 'invalid user.*from' auth.log | awk '{c\[\\$11\]++} END {for \(i in c\) print c\[i\],i}' | sort -nr | head -25](#)
- [% /var/log# grep 'invalid user.*from' auth.log | awk '{c\[\\$11\]++} END {for \(i in c\) print c\[i\],i}' | sort -nr | tail -25](#)
- [% /var/log# grep 'invalid.* from' auth.log | awk '{c\[\\$13\]++}END{for\(i in c\) print c\[i\],i}' | sort -nr | head](#)
- [% /var/log# grep 'invalid.* from' auth.log | awk '{c\[\\$13\]++}END{for\(i in c\) print c\[i\],i}' | sort -nr | awk '{system\("nslookup " \\$2\)}' | grep 'name =' | head](#)
- [% /var/log# tail syslog](#)
- [% /var/log/apt# cat history.log](#)
- [% /var/log# cat postgresql/postgresql-9.5-main.log.1](#)
- [% /var/log# tail apache2/access.log.1](#)
- [% /var/log# grep 'Mozilla.*NT' apache2/access.log* | tail](#)
- [% /var/log# grep -v ' / ' apache2/access.log* | grep -v 'week\[123\]' | sed 's/\[^:\]*:/' | grep -v Binary](#)
- [% /var/log# grep -v ' / ' apache2/access.log* | grep -v 'week\[123\]' | sed 's/\[^:\]*:/' | grep -v Binary | awk '{system\("nslookup " \\$1\)}' | grep 'name ='](#)
- [% /var/log# head -25 ~/.bash_history | sed 's/@.*:.*@xx.xx.xx.xx/'](#)

```
% /var/log# ls
```

```
alternatives.log      alternatives.log.9.gz  dist-upgrade         dpkg.log.8.gz       syslog.1
alternatives.log.1    apache2              dpkg.log             dpkg.log.9.gz       syslog.2.gz
alternatives.log.10.gz apt                  dpkg.log.1          fsck                 syslog.3.gz
alternatives.log.11.gz auth.log             dpkg.log.10.gz      kern.log             syslog.4.gz
alternatives.log.12.gz auth.log.1           dpkg.log.11.gz      kern.log.1           syslog.5.gz
alternatives.log.2.gz auth.log.2.gz        dpkg.log.12.gz      kern.log.2.gz        syslog.6.gz
alternatives.log.3.gz auth.log.3.gz        dpkg.log.2.gz       kern.log.3.gz        syslog.7.gz
alternatives.log.4.gz auth.log.4.gz        dpkg.log.3.gz       kern.log.4.gz        sysstat
alternatives.log.5.gz btmp                 dpkg.log.4.gz       lastlog              unattended-upgrades
alternatives.log.6.gz btmp.1              dpkg.log.5.gz       lxd                  wtmp
alternatives.log.7.gz cloud-init.log       dpkg.log.6.gz      postgresql            wtmp.1
```

```
% /var/log# ls % /var/log# head auth.log % /var/log# grep 'invalid user.*from' auth.log | tail % /var/log# grep 'invalid user.*from' auth.log | awk
'{c[$11]++} END {for (i in c) print c[i],i}' | sort -nr | head -25 % /var/log# grep 'invalid user.*from' auth.log | awk '{c[$11]++} END {for (i in c)
print c[i],i}' | sort -nr | tail -25 % /var/log# grep 'invalid.* from' auth.log | awk '{c[$13]++}END{for(i in c) print c[i],i}' | sort -nr | head % /var/log#
```

```
grep 'invalid.* from' auth.log | awk '{c[$13]++}END{for(i in c) print c[i],i}' | sort -nr | awk '{system("nslookup " $2)}' | grep 'name =' | head %  
/var/log# tail syslog % /var/log/apt# cat history.log % /var/log# cat postgresql/postgresql-9.5-main.log.1 % /var/log# tail apache2/access.log.1 %  
/var/log# grep 'Mozilla.*NT' apache2/access.log* | tail % /var/log# grep -v ' / ' apache2/access.log* | grep -v 'week[123]' | sed 's/[^:]*://' | grep -v  
Binary
```

