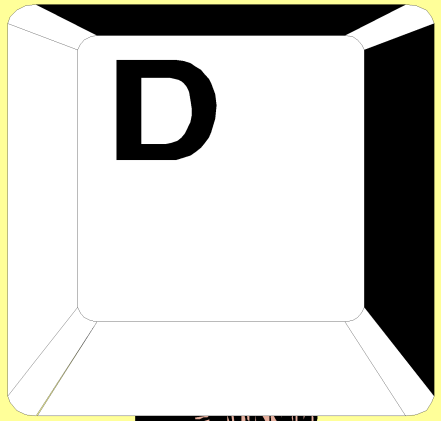
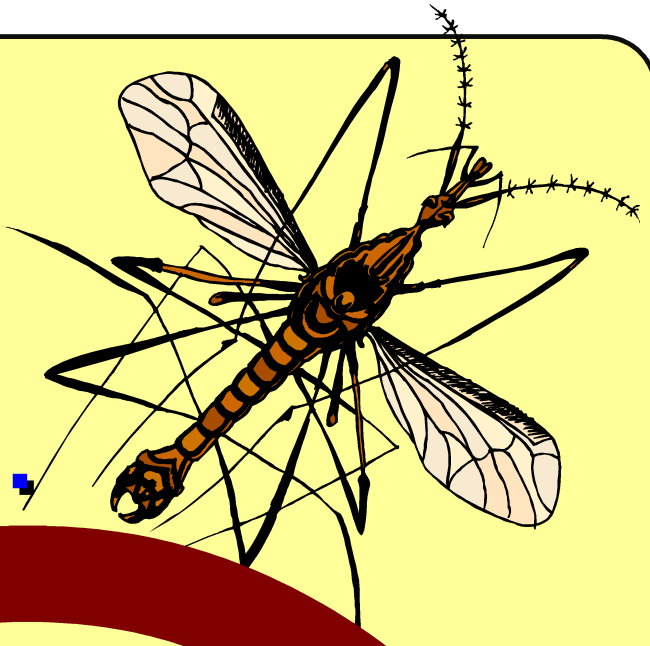


is for Debugging.



Copyright

Environment Variables

```
xargs --null --max-args=1 echo < /proc/2977/environ
 - or -
cat /proc/2977/environ | tr \\0 \\n
```

Process Id (PID)

```
LESS_ADVANCED_PREPROCESSOR=no
OSTYPE=linux
XCURSOR_THEME=DMZ
WINDOWMANAGER=/usr/bin/gnome
G_FILENAME_ENCODING=@locale,UTF-8,...
LESS=-M -I -R
MACHTYPE=x86_64-suse-linux
LOGNAME=awilliam
CVS_RSH=ssh
```

Strings in /proc/{pid}/environ are null terminated without new lines (variable values may contain new lines) so you need to translate them to get a nice display.

Open Files

```
lsof -p 2977 | cut -c23-
```

```
  2w   REG  253,0    98580 13500664 /home/awilliam/.config/banshee-
1/log
  4r   CHR   1,9      0t0     1034 /dev/urandom
  5u  unix 0xffff8801f526ce00      0t0   140620 socket
  6u  0000    0,9        0     3688 anon_inode
  8u  unix 0xffff88021f569c00      0t0   140620 socket
  9u   REG    253,0 26275840 13501376
/home/awilliam/.config/banshee-1/banshee.db
 10u  sock    0,7      0t0   139941 can't identify protocol
 16w  FIFO  0,8      0t0   141490 pipe
 17u  unix 0xffff8801e169c4c0      0t0   141493 socket
 19u  IPv4    134945      0t0      TCP localhost:8089 (LISTEN)
 21u  IPv4    958178      0t0      TCP 10.66.1.101:38333-
>174.37.70.140-static.reverse.softlayer.com:http (ESTABLISHED)
 22u  0000    0,9        0     3688 anon_inode
```

lsof lists all the file like objects open by the specified process.

lsof Output

File Descriptor

Mode (Read/Write)

```
COMMAND PID  USER   FD   TYPE ...
banshee 2977 awilliam   2w   REG ...
banshee 2977 awilliam   3u  unix ...
banshee 2977 awilliam   4r   CHR ...
banshee 2977 awilliam   5u  unix ...
banshee 2977 awilliam   6u  0000 ...
banshee 2977 awilliam  19u  IPv4 ...

...              DEVICE SIZE/OFF     NODE NAME
...              253,0     2850 1468891 ~/.config/banshee-1/log
...0xffff88012bd711c0       0t0   32583 socket
...                1,9       0t0    1034 /dev/urandom
...0xffff88012bf0d200       0t0   33067 socket
...                0,9         0    3674 anon_inode
...              32638       0t0     TCP localhost:8089 (LISTEN)
```

-Z will also add a column of the SELinux security context.

fuser

```
fuser -u  /home/awilliam/.config/banshee-1/log
```

```
/home/awilliam/.config/banshee-1/log:  2977(awilliam)
```

```
fuser -n tcp -u 8089
```

```
8089/tcp:             2977(awilliam)
```

What?

```
ls -l /proc/2977/exe
```

```
lrwxrwxrwx 1 awilliam users 0 Jul 24 21:03
/proc/2977/exe -> /usr/bin/mono
```

```
cat /proc/2977/cmdline | tr \\0 \\n
```

```
banshee
/usr/lib64/banshee/Banshee.exe
--redirect-log
--play-enqueued
```

Where

```
ls -l /proc/2977/cwd
```

```
lrwxrwxrwx 1 awilliam users 0 Jul 25 06:05
/proc/2977/cwd -> /home/awilliam
```

```
cat /proc/2977/environ | tr \\0 \\n | grep ^PATH
cat /proc/2977/environ | tr \\0 \\n | grep ^LD
cat /proc/2977/environ | tr \\0 \\n | grep ^PY
```

```
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11R6/bin:/usr/ga
mes:/opt/kde3/bin:/usr/lib/mit/bin:/usr/lib/mit/sbin
LD_LIBRARY_PATH=/usr/lib64/banshee:/usr/lib64/banshee/Extensio
ns:/usr/lib64/banshee/Backends:/usr/lib64
PYTHONSTARTUP=/etc/pythonstart
```

What kind

file /usr/lib64/banshee/Banshee.exe

/usr/lib64/banshee/Banshee.exe: **PE32** executable (console) Intel 80386, Mono/.Net assembly, for MS Windows

See binfmt_misc for how file types get mapped to execution handlers.

file /usr/bin/mono

/usr/bin/mono: **ELF** 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.16, BuildID[sha1]=0x18bba2a79cc5fec9bf0df6b29371959cd12c03b1, stripped

Depends on

ldd /usr/bin/mono

To really test your **LD_*** variables need to be the same as those of the running process; see "**Environment Variables**"

```
linux-vdso.so.1 =>  (0x000
libm.so.6 => /lib64/libm.so.6 (0x00007f5ead65d000)
librt.so.1 => /lib64/librt.so.1 (0x00007f5ead455000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f5ead251000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f5ead034000)
libc.so.6 => /lib64/libc.so.6 (0x00007f5eacca4000)
/lib64/ld-linux-x86-64.so.2 (0x00007f5ead8b4000)
```

Linker errors "*bubble up*". So if any library cannot be loaded (found) executing the executable will typically fail with "File not found" although the executable file itself is ***clearly right there!***

Limits

cat /proc/2977/limits

Limit	Soft Limit	Hard Limit	Units
Max cpu time	unlimited	unlimited	seconds
Max file size	unlimited	unlimited	bytes
Max processes	63863	63863	processes
Max open files	**1024**	**4096**	**files**
Max locked memory	65536	65536	bytes
Max pending signals	63863	63863	signals
Max msgqueue size	819200	819200	bytes
Max nice priority	0	0	
Max realtime priority	0	0	
Max realtime timeout	unlimited	unlimited	us

Max open file is the most likely setting to cause troubles; many modern applications can easily open 4,096 files; you'll certainly pass this limit on a Samba file-server.

Process I/O

cat /proc/2977/io

```
rchar: 24396601
wchar: 65012571
syscr: 23966
syscw: 29911
read_bytes: 31813632
write_bytes: 56659968
cancelled_write_bytes: 2625536
```

For profiling and performance do not neglect looking at I./O; everyone likes to look at CPU and memory, I/O is more likely to be the choke point.

System I/O

dstat

dstat displays a nice summary of overall system I/O activity.

```
----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int   csw
  0   1  98   0   0   0|  40k  194k|   0     0 |   0     0 | 541   318
  0   0 100   0   0   0|   0     0 | 429B  429B|   0     0 |1027    44
  0   1  99   0   0   0|   0    32k| 126B  338B|   0     0 | 972    37 2
  0   7  92   1   0   0|   0   456k| 297B  461B|   0     0 | 992   642
```

Where is the I/O going? strace it!

```
strace-e trace=read,write,open,close \
  /bin/cat /usr/lib64/libglib-2.0.so > /dev/null
```

```
open("/usr/lib64/libglib-2.0.so", O_RDONLY) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3"..., 32768) = 32768
write(1, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1"..., 32768) = 32768
`
```

File Handle

Bytes Read/Written

You can attach to a running process with strace by using the -p *PID* argument.

The -T arguement will add the duration of each system call to the output,

Logging Detailed System Wide I/O

echo "1" > /proc/sys/vm/block_dump

```
[ 2032.934178] postmaster(11528): READ block 5058592 on dm-3 (16 sectors)
[ 2032.934200] postmaster(11528): READ block 5058624 on dm-3 (32 sectors)
[ 2032.934240] postmaster(11528): READ block 3172800 on dm-3 (16 sectors)
[ 2032.945328] banshee-1(11267): dirtied inode 1051864 (banshee.db-
journal) on dm-0
[ 2032.945336] banshee-1(11267): dirtied inode 1051864 (banshee.db-
journal) on dm-0
[ 2033.042671] python(11518): READ block 9017928 on dm-2 (32 sectors)
[ 2033.055771] python(11518): dirtied inode 267260 (expatbuilder.pyc) on
dm-2
[ 2033.055808] python(11518): READ block 9017960 on dm-2 (40 sectors)
[ 2033.412972] nautilus(11078): dirtied inode 410492
```

Logs to the kernel ring buffer.

echo "0" > /proc/sys/vm/block_dump

Count the system calls

strace -c -p 7774

```
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 63.44    0.019996          40       505           fsync
  9.52    0.003000          27       112           fdatasync
  9.52    0.003000           6       502           ftruncate
  6.97    0.002198           0      9790           read
  5.32    0.001676           1      2570       557 open
  1.83    0.000576          24        24           brk
  1.63    0.000514           0      5628           write
  1.03    0.000325          36         9           munmap
  0.32    0.000102           0      2121      2040 unlink`
```

This example is connecting to a PostgreSQL worker performing a VACUUM FULL;

You can connect via PID and record stats until you hit your break key (Ctrl-C, usually)

How many network connections?

sudo ss --summary

```
Total: 639 (kernel 707)
TCP:   46 (estab 18, closed 9, orphaned 0, synrecv 0, timewait 9/0),
ports 40

Transport Total     IP        IPv6
*           707       -         -
RAW           2         2         0
UDP          19        12         7
TCP          37        30         7
INET         58        44        14
FRAG          0         0         0
```

Who is listening?

sudo ss --listen --numeric --processes

```
Recv-Q Send-Q  Local Address:Port Peer Address:Port
0      128     *:4369           *:*    users:(("epmd",1520,3))
0      10      :::5298          :::*   users:(("telepathy-..",2901,7))
0      128     127.0.0.1:8307 *:*    users:(("hostd-worker",2319,35))
0      128     ::1:8307         :::*   users:(("hostd-worker",2319,34))
0      10      *:53654          *:*    users:(("seahorse-...",2551,7))
0      128     :::22            :::*   users:(("sshd",1389,4))
0      128     *:22             *:*    users:(("sshd",1389,3))
0      128     127.0.0.1:631  *:*    users:(("cupsd",1160,8))
0      128     ::1:631          :::*   users:(("cupsd",1160,7))
0      128     *:55672          *:*    users:(("beam.smp",1813,17))
0      128     127.0.0.1:5432 *:*    users:(("postmaster",1717,4))
0      128     ::1:5432         :::*   users:(("postmaster",1717,3))
0      10      127.0.0.1:8089 *:*    users:(("banshee",4221,19))
```

What listens should be an important input in your firewall configuration.

Detailed Network Statistics

netstat --interfaces

```
Iface   MTU Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TXERR TXDRP TXOVR Flg
eth0   1500   0 46425      0      0      0 38928     0      0      0 BMRU
lo    16436   0   474      0      0      0   474     0      0      0 LRU
Vmnet1 1500   0     0      0      0      0    36     0      0      0 BMRU
vmnet8 1500   0     0      0      0      0    36     0      0      0 BMRU
```

Boring! And lot really useful; lack of interface errors does not mean your network is feeling well.

netstat --statistics

```
Tcp:
    423 active connections openings
    12 passive connection openings
    1 failed connection attempts
    10 connection resets received
    8 connections established
    24 segments retransmited
    0 bad segments received.
    206 resets sent
```

Much more interesting, far more useful.

What about my socket?

My IRC clients local end-point is TCP/59475; lsof told me this.

```
ss --extended --processes --info  '( sport == :59475 )'
```

```
State   Recv-Q Send-Q  Local Address:Port         Peer Address:Port
ESTAB       0      0   10.66.1.101:59475         213.179.58.83:ircu
  timer:(keepalive,74min,0) users:(("xchat",4085,9))
  uid:1000 ino:47314 sk:ffff8800c543b100
  ts sack cubic wscale:7,6 rto:352 rtt:149.875/1 ato:40 cwnd:5
    send 386.5Kbps rcv_rtt:797.25 rcv_space:22626
```

Data You've Sent

ACK Time Out

Round-Trip Time

Retransmission Time Out

TCP Window Scale