
Advanced UNIX tools

(Mostly Solaris, some Linux)

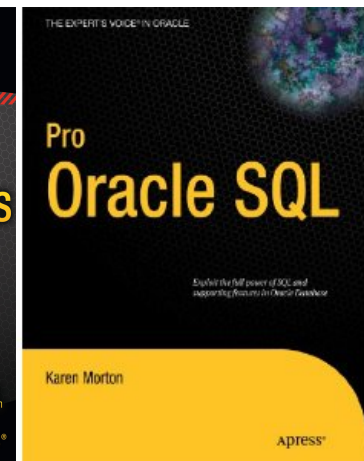
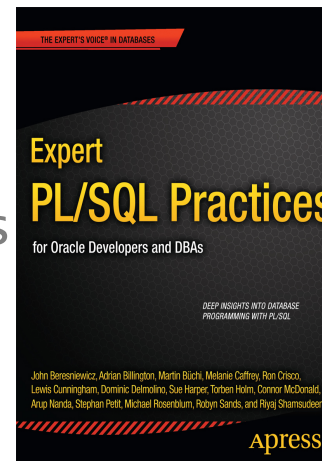
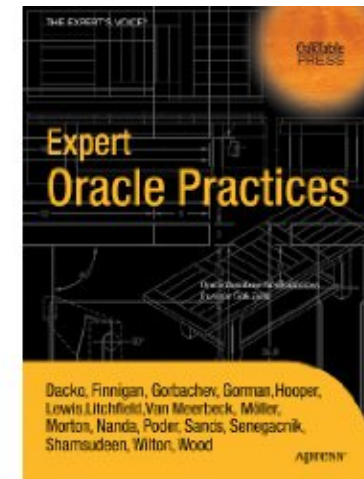
By

Riyaj Shamsudeen

Me



- 18 years using Oracle products/DBA
- OakTable member
- Oracle ACE
- Certified DBA versions 7.0,7.3,8,8i, 9i &10g
- Specializes in RAC, performance tuning, Internals and E-business suite
- Chief DBA with OraInternals
- Email: rshamsud@orainternals.com
- Blog : [orainternals.wordpress](http://orainternals.wordpress.com)
- URL: www.orainternals.com



Program is stuck ?

- Problem statement:

Program is stuck for 2+ hours, program usually completes 30 minutes or so.

Demo: `testcase1.sql`

Truss - trace system calls and signals

```
truss -p 28393
```

```
llseek(18, 0, SEEK_CUR)           = 0x012EF9A4
fstat(18, 0xFFBFA058)             = 0
write(18, " 9 8 7 1 0  o b j\n <"..., 21) = 21
fstat(18, 0xFFBFA058)             = 0
write(18, " 9 8 7 2 0  R > >\n s"..., 18) = 18
fstat(18, 0xFFBFA0C0)             = 0
write(18, " q\n 0 . 0 0  1 4 3 . 7"..., 5700) = 5700
fstat(18, 0xFFBFA100)             = 0
write(18, " e n d s t r e a m\n e n"..., 17) = 17
fstat(18, 0xFFBFA100)             = 0
write(18, " 9 8 7 2 0  o b j\n 5"..., 23) = 23
fstat(18, 0xFFBFA100)             = 0
lseek(17, 0x0216B000, SEEK_SET)    = 0x0216B000
write(17, "c8021686 )\0\0 )D0\0\0\0"..., 4096) = 4096
lseek(17, 0x0219D000, SEEK_SET)    = 0x0219D000
read(17, "\0\0\0\00101\001FF \0\0"..., 4096) = 4096
lseek(17, 0x0219E000, SEEK_SET)    = 0x0219E000
read(17, "d3\007\0\015CC\0\0\0 qB0"..., 4096) = 4096
lseek(17, 0x0216D000, SEEK_SET)    = 0x0216D000
write(17, "\0 \b\0\0\0 qAA18 L o s"..., 4096) = 4096
lseek(17, 0x0219F000, SEEK_SET)    = 0x0219F000
read(17, "\0\0\0\0\0\0\0\0\0\00101\001"..., 4096) = 4096
write(18, " 9 8 7 0 0  o b j\n <"..., 189) = 189
fstat(18, 0xFFBFA058)             = 0
llseek(18, 0, SEEK_CUR)           = 0x012F10F4
fstat(18, 0xFFBFA058)             = 0
write(18, " 9 8 7 4 0  o b j\n <"..., 21) = 21
fstat(18, 0xFFBFA058)             = 0
write(18, " 9 8 7 5 0  R > >\n s"..., 18) = 18
fstat(18, 0xFFBFA0C0)             = 0
write(18, " q\n 0 . 0 0  1 4 3 . 7"..., 5736) = 5736
fstat(18, 0xFFBFA100)             = 0
write(18, " e n d s t r e a m\n e n"..., 17) = 17
fstat(18, 0xFFBFA100)             = 0
```

Process seemingly calling
Many seek, write and fstat calls

For, seek, fstat, write, read calls etc,
first argument is the file descriptor.

For read, write call second argument
is the buffer itself.

What files are this process is writing
to ? How big are those files ?

Truss

Description:

The truss utility traces the system calls and the signal process receives.

Options:

```
truss [-fcaeildD] [- [tTvX] [!] syscall,...] [- [sS] [!] signal,...] [-  
[mM] [!] fault,...] [- [rw] [!] fd,...] [- [uU] [!] lib,... : [:] [!] func,...] [-  
o outfile] com- mand | -p pid...
```

| |
|-----------------------|
| Solaris - truss |
| Hpux- tusc (download) |
| Linux - strace |

Truss

To trace a process and print minimal information

`truss -p <pid>` Example: `truss -p 23898`

To trace a process, follow its children and print minimal information

`truss -f -p <pid>` Example: `truss -f -p 23898`

To trace a process, print timestamp and print minimal information

`truss -d -p <pid>` Example: `truss -d -p 23898`

To trace a process, send output to a file and print minimal information.

`truss -o /tmp/truss.out -p <pid>`

Example: `truss -o /tmp/truss.out -d -p 23898`

Truss – Word of caution

At every system call, truss inspects the process.
This *potentially* could slow down the process.

So, Truss critical processes, only when it is
necessary to do so.

Truss – Few outputs

```
$ truss -d -E -p 1873
```

```
Base time stamp: 1310009834.7781 [ wed Jul 6 22:37:14 CDT 2011 ]
0.0124 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.1128 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.1130 0.0000 mmap(0xFFFFFD7FFC1DE000, 65536, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED, 7, 0) = 0xFFFFFD7FFC1DE000
0.2132 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.3138 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.4142 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.5146 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.6150 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.7163 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
0.8181 0.0000 semtimedop(7, 0xFFFFFD7FFDF9328, 1, 0xFFFFFD7FFDF9340) Err#11 EAGAIN
```

↑
Time stamp displacement
From base timestamp.
Seconds.fraction of sec

↖
Time taken in the system call.

truss & pfiles

Truss:

```
write(18, " 9 8 7 1 0 o b j\n <"..., 21)    = 21  
fstat(18, 0xFFBFA058)                       = 0  
write(18, " 9 8 7 2 0 R > >\n s"..., 18)    = 18
```

Pfiles:

- ❑ pfiles can be used to associate this file ids with file names.
 - ❑ Pfiles lists the files currently opened by a process. In few unix platform, this can be achieved by lsof command also.
-

pfiles

Using these device numbers and Inode numbers, file names can be mapped.

```
pfiles 28393
```

```
28393: ar60runb P_CONC_REQUEST_ID=2452107 STARTDATE='012006'  
      ENDDATE='122006'
```

```
Current rlimit: 4096 file descriptors
```

```
0: S_IFIFO mode:0000 dev:272,0 ino:7325504 uid:11175 gid:100 size:0  
   O_RDWR
```

```
1: S_IFREG mode:0644 dev:233,63004 ino:895220 uid:11175 gid:100 size:0  
   O_WRONLY|O_APPEND|O_CREAT
```

```
2: S_IFREG mode:0644 dev:233,63004 ino:895220 uid:11175 gid:100 size:0  
   O_WRONLY|O_APPEND|O_CREAT
```

```
...
```

```
17: S_IFREG mode:0644 dev:233,63004 ino:895242 uid:11175 gid:100 size:  
    102522880  
     O_RDWR|O_CREAT|O_TRUNC
```

```
18: S_IFREG mode:0644 dev:233,63004 ino:895305 uid:11175 gid:100 size:  
    25491841  
     O_RDWR|O_CREAT|O_TRUNC
```

This is the file_id
In the truss output

This is the device id
Of the form minor,major

Inode number

Pfiles & proc tools

Many tools available, aka proc tools

pflags, pcred, pldd, psig, pstack, pfiles, pwdx,
pstop, prun, pwait, ptree, ptime

WARNINGS

The following proc tools stop their target processes while inspecting them and reporting the results: pfiles, pldd, pmap, and pstack.

A process can do nothing while it is stopped. Stopping a heavily used process in a production environment, even for a short amount of time, can cause severe bottlenecks ..

pmap

- Error message is memory relevant.
 - Process memory need to be monitored and pmap command can give a breakdown of process memory.
-

pmap <pid>

| Address | Kbytes | RSS | Anon | Locked | Mode | Mapped File |
|-----------------|---------------|--------------|--------------|--------|-------|-------------------------|
| 00010000 | 72 | 72 | - | - | r-x-- | java |
| 00030000 | 16 | 16 | 16 | - | rwX-- | java |
| 00034000 | 8744 | 8680 | 8680 | - | rwX-- | [heap] |
| 77980000 | 1224 | 1048 | - | - | r--s- | dev:273,2000 ino:104403 |
| 77CFA000 | 24 | 24 | 24 | - | rw--R | [anon] |
| 77F7A000 | 24 | 24 | 24 | - | rw--R | [anon] |
| 78000000 | 72 | 72 | 72 | - | rwX-- | [anon] |
| 7814C000 | 144 | 144 | 144 | - | rwX-- | [anon] |
| 783E8000 | 32 | 32 | 32 | - | rwX-- | [anon] |
| 78408000 | 8 | 8 | 8 | - | rwX-- | [anon] |
| 78480000 | 752 | 464 | - | - | r--s- | dev:85,0 ino:13789 |
| 7877E000 | 8 | 8 | 8 | - | rw--R | [anon] |
| 78800000 | 36864 | 8192 | 8192 | - | rwX-- | [anon] |
| | | | | | | |
| FF25C000 | 16 | 8 | 8 | - | rwX-- | libcrun.so.1 |
| FF276000 | 8 | 8 | - | - | rwXs- | [anon] |
| FF280000 | 688 | 688 | - | - | r-x-- | libc.so.1 |
| FF33C000 | 32 | 32 | 32 | - | rwX-- | libc.so.1 |
| FF350000 | 16 | 16 | 16 | - | rw--- | [anon] |
| FF360000 | 8 | 8 | 8 | - | rwX-- | [anon] |
| FF370000 | 96 | 96 | - | - | r-x-- | libthread.so.1 |
| FF398000 | 8 | 8 | 8 | - | rwX-- | libthread.so.1 |
| FF39A000 | 8 | 8 | 8 | - | rwX-- | libthread.so.1 |
| FF3A0000 | 8 | 8 | - | - | r-x-- | libc_psr.so.1 |
| FF3B0000 | 184 | 184 | - | - | r-x-- | ld.so.1 |
| FF3EE000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3F0000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3FA000 | 8 | 8 | 8 | - | rwX-- | libdl.so.1 |
| FFB80000 | 24 | - | - | - | ----- | [anon] |
| FFBF0000 | 64 | 64 | 64 | - | rw--- | [stack] |
| ----- | | | | | | |
| total kb | 182352 | 65568 | 26360 | - | | |

Pmap prints a
Nice memory map
of the Process.
Verious heaps and
Stacks are printed here

Total memory foot print
Also printed.

pmap

```
#!/bin/ksh
pid=$1
(( cnt=1000 ))
while [[ $cnt -gt 0 ]];
do
    date
    pmap -x $pid
    pstack $pid
    echo $cnt
    (( cnt=cnt-1 ))
    sleep 10
done
```

Wrote this small shell script, to dump Memory map and stack of this Process, in a loop, every 10 seconds.

pmap

| Address | Kbytes | RSS | Anon | Locked | Mode | Mapped File |
|----------|---------------|-------|-------|--------|-------|-------------------------|
| 00010000 | 72 | 72 | - | - | r-x-- | java |
| 00030000 | 16 | 16 | 16 | - | rwX-- | java |
| 00034000 | 8744 | 8680 | 8680 | - | rwX-- | [heap] |
| 77980000 | 1224 | 1048 | - | - | r--s- | dev:273,2000 ino:104403 |
| 77CFA000 | 24 | 24 | 24 | - | rw--R | [anon] |
| ... | | | | | | |
| FF39A000 | 8 | 8 | 8 | - | rwX-- | libthread.so.1 |
| FF3A0000 | 8 | 8 | - | - | r-x-- | libc_psr.so.1 |
| FF3B0000 | 184 | 184 | - | - | r-x-- | ld.so.1 |
| FF3EE000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3F0000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3FA000 | 8 | 8 | 8 | - | rwX-- | libdl.so.1 |
| FFB80000 | 24 | - | - | - | ----- | [anon] |
| FFBF0000 | 64 | 64 | 64 | - | rw--- | [stack] |
| ----- | ----- | ----- | ----- | ----- | | |
| total kb | 182352 | 65568 | 26360 | - | | |

Process initially started with a memory usage of 182MB

pmap

| Address | Kbytes | RSS | Anon | Locked | Mode | Mapped File |
|----------|---------------|--------|--------|--------|-------|-------------------------|
| 00010000 | 72 | 72 | - | - | r-x-- | java |
| 00030000 | 16 | 16 | 16 | - | rwX-- | java |
| 00034000 | 8808 | 8720 | 8720 | - | rwX-- | [heap] |
| 77980000 | 1224 | 1048 | - | - | r--s- | dev:273,2000 ino:104403 |
| 77CFA000 | 24 | 24 | 24 | - | rw--R | [anon] |
| 77F7A000 | 24 | 24 | 24 | - | rw--R | [anon] |
| 78000000 | 72 | 72 | 72 | - | rwX-- | [anon] |
| 78012000 | 64 | 64 | 64 | - | rwX-- | [anon] |
| 7814C000 | 144 | 144 | 144 | - | rwX-- | [anon] |
| 78170000 | 8 | 8 | 8 | - | rwX-- | [anon] |
| 78172000 | 8 | 8 | 8 | - | rwX-- | [anon] |
| 78174000 | 8 | 8 | 8 | - | rwX-- | [anon] |
| 78176000 | 104 | 104 | 104 | - | rwX-- | [anon] |
| .. | | | | | | |
| FF370000 | 96 | 96 | - | - | r-x-- | libthread.so.1 |
| FF398000 | 8 | 8 | 8 | - | rwX-- | libthread.so.1 |
| FF39A000 | 8 | 8 | 8 | - | rwX-- | libthread.so.1 |
| FF3A0000 | 8 | 8 | - | - | r-x-- | libc_psr.so.1 |
| FF3B0000 | 184 | 184 | - | - | r-x-- | ld.so.1 |
| FF3EE000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3F0000 | 8 | 8 | 8 | - | rwX-- | ld.so.1 |
| FF3FA000 | 8 | 8 | 8 | - | rwX-- | libdl.so.1 |
| FFB80000 | 24 | - | - | - | ----- | [anon] |
| FFBF0000 | 64 | 64 | 64 | - | rw--- | [stack] |
| ----- | ----- | ----- | ----- | ----- | | |
| total kb | 281040 | 210736 | 171528 | - | | |

As the process was running, Process memory usage started to grow.

Problem #2

Program is running for many hours. Recently there was a *minor* code change to the program.

Demo: `testcase2.sql`

pstack

- pstack shows current stack of the process. Let's look at pstack for this java process:

```
pstack 1567
```

```
1567:      oraclesolrac1 (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
```

```
00000000ab1418f pevmsubstr () + 12f
```

```
00000000aad49bf pfrinstr_substr () + 5f
```

```
00000000aac5880 pfrun_no_tool () + 40
```

```
00000000aac6a6f pfrun () + 4df
```

```
00000000ab2e3fa plsql_run () + 2ea
```

```
00000000aaa4a83 peicnt () + 143
```

```
00000000a0fba56 kxexe () + 216
```

```
00000000447b5c7 opiexe () + 2757
```

```
000000004d54695 kpoal8 () + ce5
```

```
000000004472693 opiodr () + 433
```

```
000000008e67f69 ttcip () + 599
```

```
00000000444cfc0 opitsk () + 600
```

```
00000000445bb75 opiino () + 675
```

```
000000004472693 opiodr () + 433
```

```
000000004441f4e opidrv () + 32e
```

```
000000005672197 sou2o () + 57
```

```
00000000159eac9 opimai_real () + 219
```

```
00000000568f2de ssthrdmain () + 14e
```

```
00000000159e89b main () + cb
```

```
00000000159e67c ???????? ()
```

Oradebug short_stack

- Oradebug short_stack also can be used to get process stack.

- Example:

```
SQL> oradebug setmypid
```

```
Statement processed.
```

```
SQL> oradebug short_stack
```

```
ksedsts()+1123<-ksdxfstk()+33<-ksdxen_int()+5127<-ksdxen()+14<-opiodr()+1075<-ttcpip()  
+1433<-opitsk()+1536<-opiino()+1653<-opiodr()+1075<-opidrv()+814<-sou2o()+87<-opimai_real  
( )+537<-ssthrdmain()+334<-main()+203<-_start()+108
```

```
SQL> oradebug short_stack
```

- ksedsts()+1123<-ksdxfstk()+33<-ksdxen_int()+5127<-ksdxen()+14<-opiodr()+1075<-ttcpip()
+1433<-opitsk()+1536<-opiino()+1653<-opiodr()+1075<-opidrv()+814<-sou2o()+87<-opimai_real
()+537<-ssthrdmain()+334<-main()+203<-_start()+108

```
SQL>
```

Thank you for attending!

If you like this presentation, you will love
My upcoming seminar in Aug 2011 & Sep 2011.

<http://blog.tanelpoder.com/seminar/>

Contact info:

Email: rshamsud@gmail.com

Blog : orainternals.wordpress.com

URL : www.orainternals.com

