Troubleshooting OS/2 - A Guide to the Troubled

Presented at

Warpstock 2015, June 6-7, 2015
Munich, Germany
by
Steven H. Levine

OS/2 is reliable and most of the time, it just runs for days with little or no attention. However, as with any complex system, issues can occur.

OS/2 supports a large number of tools that can be helpful in tracking down and resolving the source of these issues.

Exception Reporting

No one likes exceptions, but they do happen. The kernel defines two types of exception reports

- Process Exception Reports
- Kernel Internal Processing Error (IPE) Exception Reports

Some applications define private exception handlers that augment or replace the default Process Exception report with an application specific reports. Examples include:

- Exceptq Exception Reports
- GCC kLIBC Exception Reports
- eWorkplace/XWorkplace Exception Reports
- Innotek Font Library Exception Reports

Process Exception Reports

Process (aka ring3) exception reports written by the kernel are formatted as

```
01-> 02-25-1999  10:58:35  SYS3175  PID 00b2  TID 0001  Slot 0068
02-> E:\CLASSES\LABS\LAB26\BEDBUG.EXE
03-> c0000005
04-> 1bf94e24
05-> P1=00000001  P2=00000000  P3=XXXXXXXX  P4=XXXXXXXX
06-> EAX=00000000  EBX=00060210  ECX=0002881c  EDX=00060210
07-> ESI=00000001  EDI=00000002
08-> DS=0053  DSACC=d0f3  DSLIM=1fffffff
09-> ES=0053  ESACC=d0f3  ESLIM=1fffffff
10-> FS=150b  FSACC=00f3  FSLIM=00000030
11-> GS=0000  GSACC=****  GSLIM=********
12-> CS:EIP=005b:1bf94e24  CSACC=d0df  CSLIM=1fffffff
13-> SS:ESP=0053:000287ec  SSACC=d0f3  SSLIM=1fffffff
14-> EBP=000287f8  FLG=00012206

15-> DOSCALL1.DLL 0002:00004e24
```

Process exception reports are written to the POPUPLOG.OS2 file in the root of the drive specified by the SUPPRESSPOPUPS CONFIG.SYS statement.

Kernel Internal Processing Error (IPE) Exception Reports

Kernel IPE (aka ring0) exception reports are formatted as

```
1-> <IPE specific Message>

2->  THE SYSTEM DETECTED AN INTERNAL PROCESSING
     ERROR AT LOCATION ##xxxx:yyyyyyyy - aaaa:bbbb

3->  llll , ffff
4->  038600d1
5-> INTERNAL REVISION 6 . 307  DATE: 92/03/01
```

or as

```
1->  TRAP 0002        ERRCD= 0000  ERACC= ****  ERLIM= ********
      EAX= 7d240a58  EBX= ff202fdc  ECX= 00064423  EDX= 00003624
      ESI= fff3272c  EDI= 7d240004  EBP= 00004a44  FLG= 00003202
2->   CS:EIP= 0160 : fff702a6  CSACC= c09d  CSLIM= ffffffff
      SS:ESP= 0030 : 00004a38  SSACC= 1097  SSLIM= 00003fff
3->   DS= 0158  DSACC= c0f3  DSLIM= ffffffff  CR0= fffffffb
4->   ES= 0158  ESACC= c0f3  ESLIM= ffffffff  CR2= 1a060014
      FS= 0000  FSACC= ****  FSLIM= ********
      GS= 0000  GSACC= ****  GSLIM= ********
     THE SYSTEM DETECTED AN INTERNAL PROCESSING
     ERROR AT LOCATION ##0160:fff6453f - 000d:a53f

     60000 , 9084
     038600d1
     INTERNAL REVISION 6 . 307  DATE: 92/03/01
```

Kernel IPE Exception Reports are written to the screen or to a System Dump file in the root of the volume specfied by the TRAPDUMP CONFIG.SYS statement.

Exceptq Exception Reports

are written by applications that support the exceptq exception handler.

- The exception reports are written to files named PPPP_TT.TRP where PPPP is the process id and TT is the thread ordinal.
- The exception reports are written to the working directory.
- Exceptq uses standard .sym files or optimized .xqs files to convert addresses to symbolic names.
- Fm/2 was one of the first applications to support exceptq exception reporting.
- Other applications with exceptq support include the Mozilla apps, the Apache apps, pmmail, pronews weasel, majormajor and more.
- Exceptq is implemented as a shared DLL.
- Exceptq can be installed by rpm/yum.
- Newer versions of exceptq may be available from here.
- Exceptq version 7.1 is not recommended if you are running an SMP kernel.
- The exceptq handler is based on code originally developed by IBM and released as Open Source.

GCC kLIBC Exception Reports

are by generated GCC kLIBC (i.e. libc066.dll) when an application it controls generates and exception. The report format is

```
Killed by SIGILL
pid=0x0061 ppid=0x0024 tid=0x0001 slot=0x009b pri=0x0200 mc=0x0001 N:\MOZILLA-TEST\SEAMONKEY\SEAMONKEY.EXE
LIBC063 ffffffff:ffffffff
cs:eip=005b:20d15cc1 ss:esp=0053:0011eaf8 ebp=0011eb14 ds=0053 es=0053 fs=150b gs=0000 efl=00212206
eax=20d42348 ebx=0011eb50 ecx=20da1618 edx=0011eb50 edi=0000000c esi=20d15cc0
Process dumping was disabled, use DUMPPROC / PROCDUMP to enable it.
```

- The exception reports are written to the standard error output (i.e. the screen).
- The kLIBC exception handler suppresses the POPUPLOG.OS2 report.
- It is easy to miss the kLIBC Exception Report, especially if the application is a GUI application.
- To capture a copy of the report, redirect the standard error output to a file.
- To redirect the output, use the standard command line redirection syntax (1>error.log 2>&1) or use Rich Walsh's Run!.

eWorkplace/XWorkplace Exception Reports

are written when XFLDR.DLL or some DLL it loads generates an exception.

- The eWorkplace report file is named ewptrap.log and the XWorkplace report file is named xwptrap.log.
- The report file is written to the %LOG% directory or root of the boot drive.
- The exception handler is based on the original IBM exceptq exception handler code so the report format is similar.

Innotek Font Library Exception Reports

are written when the Innotek Font library (i.e. ft2lib.dll) generates an exception.

- The report format is similar to the standard process exception report.
- The report file is named ft2excpt.log.
- The report file is written to root of the boot drive.

Tracing

Tracing is extremely valuable when you need to analyze application activity over time. There are a number of general purpose tracing tools.

- OS/2 Trace Facility
- OS2TRACE
- PMSPY

OS/2 Trace Facility

- Implemented by the kernel.
- Provides API level tracing.
- Provides tracing control to drivers and other applications.
- Tracing is controlled by the TRACE.EXE and TRACEFMT.EXE utility programs.
- The trace output formatting is limited.
- Some documented trace points appear to no longer operate.
- Two trace points are known to trap the kernel.
- The Trace Setup Reference is a cookbook setup guide.
- Trace Control Tool provides a convenient wrapper around the trace command syntax. The tool uses configuration files to provide repeatable trace setups.

Dave's Blatsche's OS2TRACE

- Provides API level tracing for ring3 applications.
- Is somewhat instrusive because it patches the binaries.
- Provides excellent output formatting.
- Has a GUI setup wrapper.
- Available from Hobbes. or from Dave's home page.

PMSPY

- Provides PM window message tracing.
- Has good filtering options.
- Output typically requires a programmer to interpret.
- Download from OS/2 Site

Logging

Logging is typically defined as recording unexpected events. This is a fuzzy defintion because

the distinction between log data and trace data is often not black and white.

There are a large number of logging tools. The list includes

- System Error Log
- Mozilla Log Facility
- eWorkplace Unexpected Condition Log
- GCC kLIBC logging
- TCP/IP Syslog Facility
- Innotek Font Library Log Facility

System Error Log

- The log entries are essentially a mini-process dumps with application specific data.
- The log data is often of limited usefulness because the captured data is not well documented.
- The logging facility is implemented by kernel APIs.
- Log record formatting is implemented by SYSLOGPM.

Mozilla Log Facility

- The Mozilla logging facility is comprehensive and configurable.
- It generally requires developer assistance to setup and interpret
- The documentation at Mozilla Logging provides a good overview of what can be done.

eWorkplace/XWorkplace Unexpected Condition Log

- This log facility records unexpected conditions detected by eWorkplace/XWorkplace.
- The eWorkplace log file is named ewplog.log and the XWorkplace log file is named xwplog.log.
- The file is written to the %TMP% directory or the root of the boot drive.

GCC kLIBC Logging

- Records unexpected conditions detected by kLIBC (i.e. libc065.dll).
- The log files have names of the form libc_pppp.log, where pppp is the process id.
- Logging requires installing a log enabled version of the kLIBC runtime DLL.
- The files are written to the working directory.

TCP/IP Syslog Facility

- The syslog facility implements a standard TCP/IP service on well-known UDP port 514.
- IBM includes a syslogd implementation in \tcpip\bin.
- The IBM syslogd is configured with the syslog.cnf files stored in the %ETC% directory.
- A third party implementation known as syslogd v3a, available from hobbes, is more feature complete than the IBM implementation.
- Syslogd v3a is configured with the syslog.conf file stored in the %ETC% directory.

Innotek Font Library Log Facility

- The log facility records detailed font processing activity.
- The log data is written to files named ft2lib_x.log, where x is the library instance.
- The files are written to root of the boot drive.
- Logging is controlled by HKEY_LOCAL_MACHINE\Software\InnoTek\InnoTek Font Engine\Logging registry key.
- These log files are recommended for finding broken font files.

Process Dump Facility

The process dump facility generates dump files in response to otherwise unhandled process exceptions or on demand.

- The facility is configured by DUMPPROCESS statment or PDUMPUSR command.
- The dump files are written to the configured directory.
- The dump files are formatted by Process Dump Formatter (pmdf).
- The Process Dump Facility is documented in x:\OS2\SYSTEM\RAS\PROCDUMP.DOC.
- The Process Dump Reference is a somewhat dated cookbook setup guide.
- The Process Dump Control script implements a convenient wrapper to setup and capture process dumps.

Trap Dump Facility

The trap dump facility generates dump files in response to otherwise unhandled kernel exceptions or on demand.

- The facility is configured by the TRAPDUMP and PDUMPSYS commands.
- System dump files are written to named volume and destroy all content.
- System dump files are formatted by the Process Dump Formatter (pmdf).
- The Trap Dump Facility is documented in x:\OS2\SYSTEM\RAS\PROCDUMP.DOC
- Trap Dump Reference is a somewhat dated cookbook setup guide.
- Dump Trap Screen extracts the trap screen data from a dump file. The utility is designed for systems that do not have a trap dump volume, but that do have a diskette drive.

Process Dump Formatter (pmdf)

- Pmdf formats process dump files, system dump filess and system log facility dump files.
- Pmdf is a generic tool which is configured to match the system that generated the dump file.
- Pmdf is configured with a combination of executable files (i.e. df_ref.exe), symbol (.sym) files and structure definition (.sdf) files.
- The configuration files are built for a specific version of an executable or kernel. However, some files will work for multiple versions of a given excutable.
- For 14.104a kernels, use 14.105 df_ret.exe.
- The pmdfvers.lst file defines the directories that contain the configuration files for a specific system.
- Pmdfvers.lst entries are indexed by the kernel internal version id which is similar to the bldlevel string.
- Kernel Debugger Reference is a cheatsheet reference to the kernel debugger and the Process Dump Formatter.

System Analysis Tools

The system analysis tools analyze and report data that is defined within the kernel

- Theseus implements comprehensive tools for memory and process analysis.
- Pstat implmenents a simple listing of loaded processes and DLLs.
- Psfiles lists all open files.
- Pssems lists all semaphores.
- Top displays process status in various formats.
- Go displays process status in various formats.
- Sid2 displays data available via the DosQuerySysState API in various formats.
- Cadh includes a process process status listing amoung its features.

Debuggers

Debuggers allow a developer to examine and control the operation of the kernel or an application in very fine detail. They are beyond the scope of this presentation, but it is useful to know what is available. The most widely used debuggers are probably

- Kernel debugger
- OpenWatcom debuggers (wdw and wd).
- VAC debuggers (icsdebug and idebug).
- ICAT debugger

Disassemblers

Disassemblers are useful for understanding programs that do not have available source code. They too are beyond the scope of this presentation, but it is useful to know what is available. Some of the widely used disassembly tools are

- pmdf
- ida
- Theseus

Final Thoughts

- Everyone should be aware of the existance of "The OS/2 Debugging Handbook, #0.7a (sg244640.inf)." It is included in the OS/2 Warp Toolkit on your eComStation CD.
- OS/2 Diagnostic Tools is the home for many of the tools and documents discussed here.
- Join the developer community.