Investigating C++ Applications in Production on Linux and Windows

Sasha Goldshtein

Software Engineer, Google Research

goldshtn

goldshtn

The Plan

- This is a talk on profiling and investigating C++ applications in production on Linux and Windows
- You’ll learn:

❑ To obtain and analyze dumps of C++ apps

❑ Which production-ready tracing tools can be used with C++ apps

❑ To obtain CPU profiles and flame graphs

❑ To identify memory leaking call stacks

Tools And Operating Systems Supported

	Linux	Windows	macOS
CPU sampling	perf, BCC	ETW	Instruments, dtrace
Dynamic tracing	perf, SystemTap, BCC	⊖	dtrace
Static tracing	perf, SystemTap, BCC	ETW	dtrace
Dump generation	core_pattern, gcore	Procdump, WER	kern.corefile, gcore
Dump analysis	gdb, lldb	Visual Studio, WinDbg	gdb, lldb

This talk

⚠ Mind The Overhead

- Any observation can change the state of the system, but some observations are worse than others
- Diagnostic tools have overhead
  - Check the docs
  - Try on a test system first
  - Measure degradation introduced by the tool

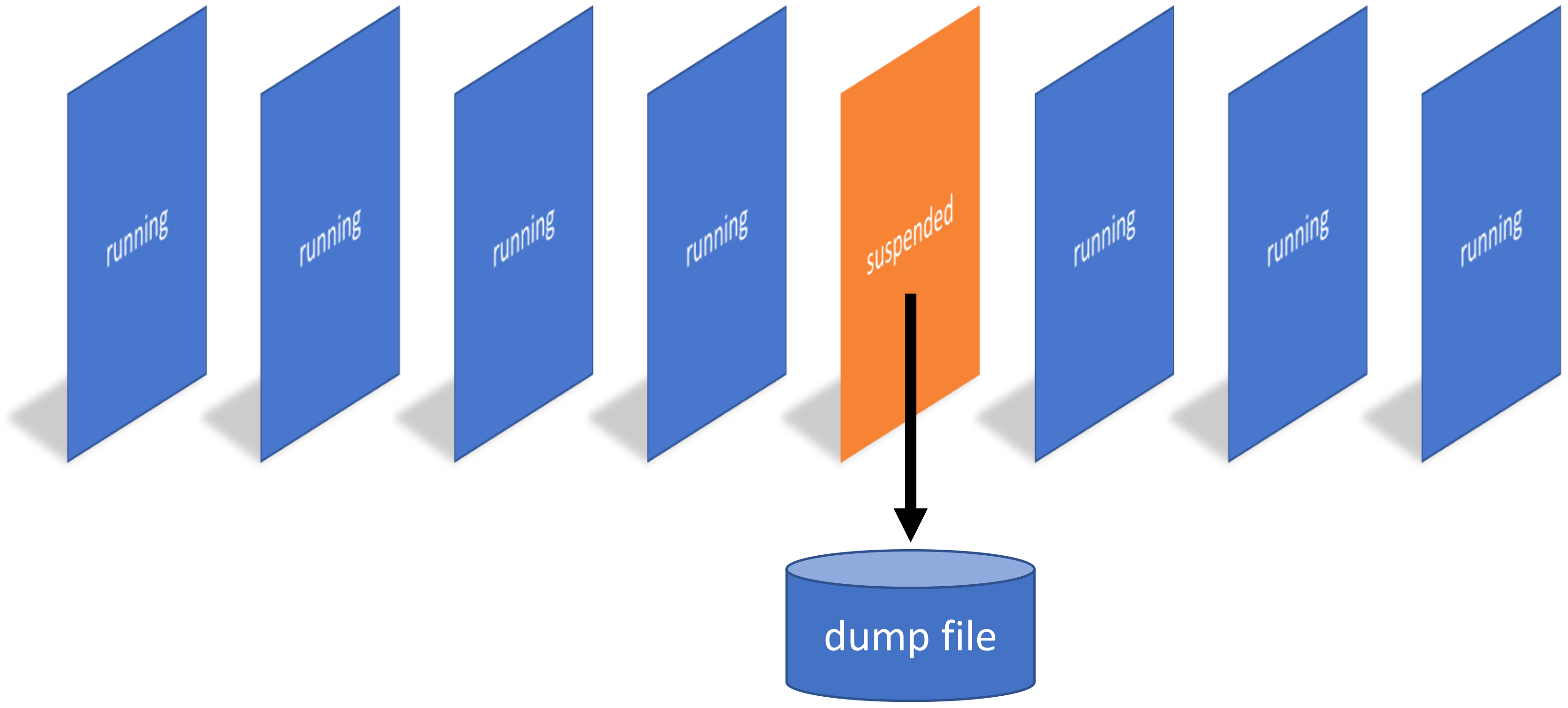
OVERHEAD

This traces various kernel page cache functions and maintains in-kernel counts, which are asynchronously copied to user-space. While the rate of operations can be very high (>1G/sec) we can have up to 34% overhead, this is still a relatively efficient way to trace these events, and so the overhead is expected to be small for normal workloads. Measure in a test environment.

—*`man cachestat`* *(from BCC)*

Dump Files/Core Dumps

- A dump file (core dump) is a memory snapshot of a running process
- Can be generated **on crash** or **on demand**



Generating Dump Files

Linux

- **/proc/sys/kernel/core_pattern** configures the core file name or application to process the crash
- **ulimit -c** controls maximum core file size (often 0 by default)
- **gcore** (part of gdb) can create a core dump on demand

Windows

- **HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps** configures the crash dump folder, count, and type (full/mini)
- **Procdump** (Sysinternals tool) can create a dump on demand

Basic Dump Analysis

Linux

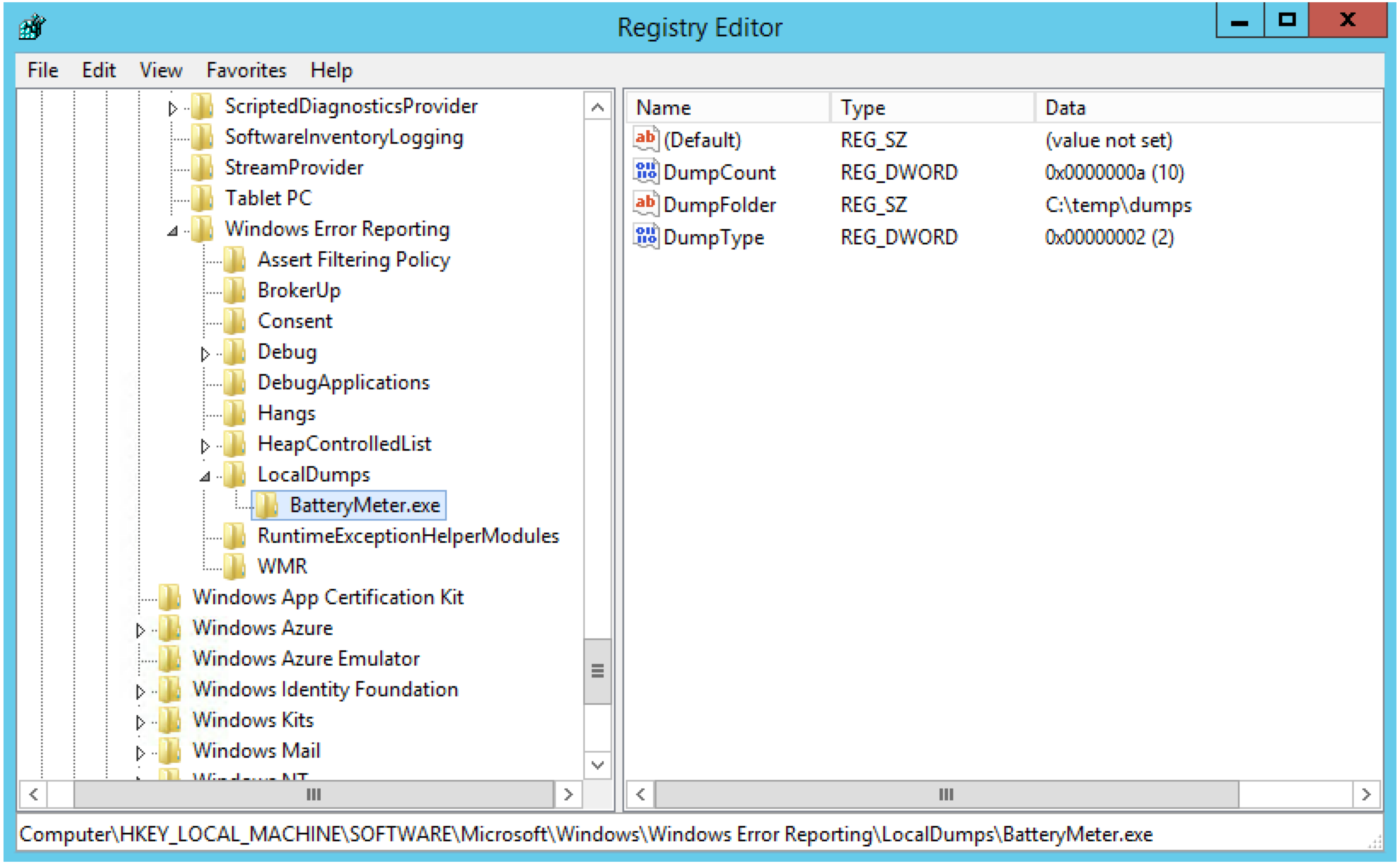
-
  ```
  gdb /path/exe -c core
                -ex "bt"
  ```
- Further automatic analysis possible using gdb or lldb Python API

Windows

- Visual Studio dump summary

-
  ```
  WinDbg -z app.dmp
         -c "!analyze -v"
  ```
- Further automatic analysis possible using WinDbg scripting language or dbgeng.dll

Demo:
Dump Generation And Analysis

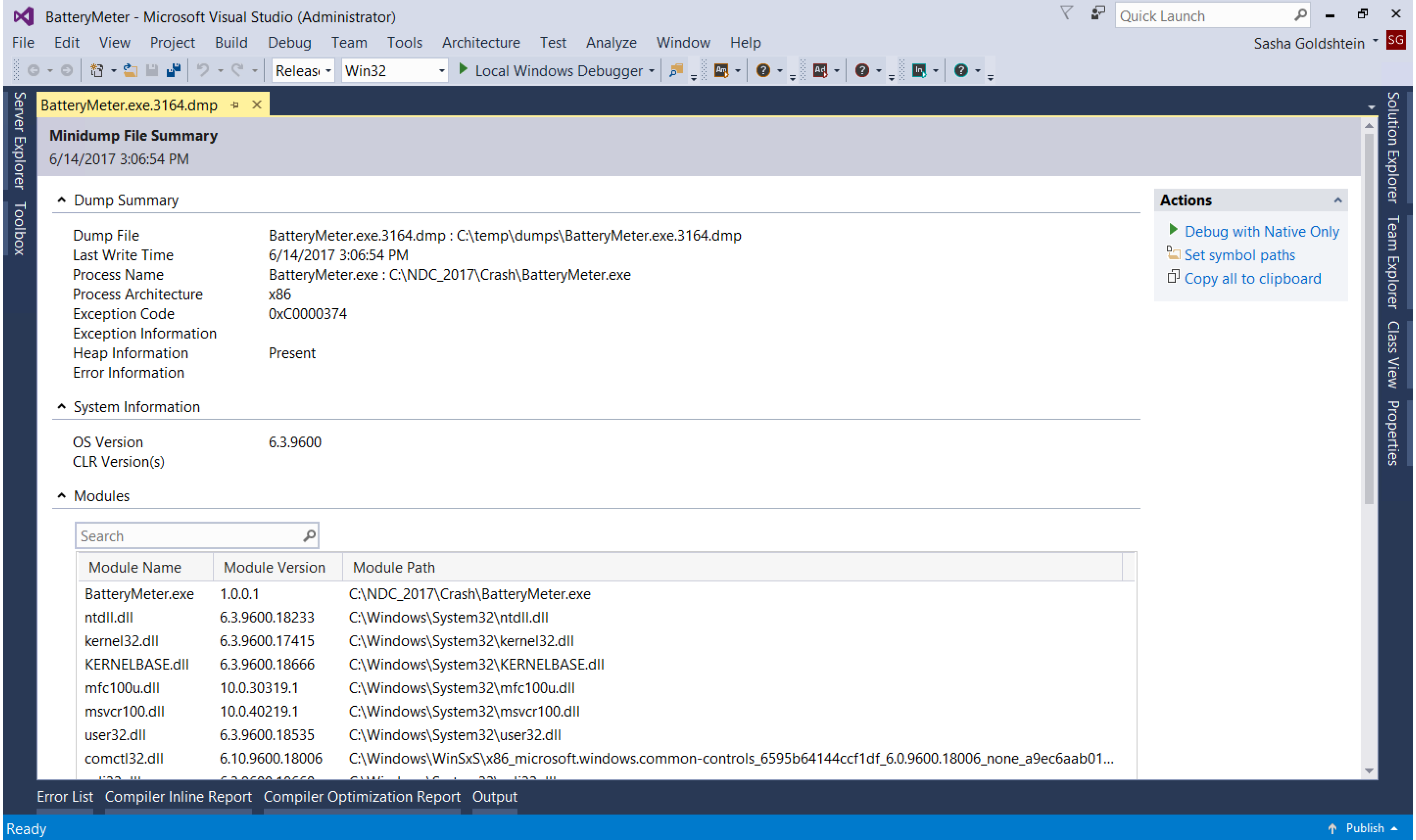

Registry Editor

File Edit View Favorites Help

- ScriptedDiagnosticsProvider
- SoftwareInventoryLogging
- StreamProvider
- Tablet PC
- Windows Error Reporting
  - Assert Filtering Policy
  - BrokerUp
  - Consent
  - Debug
  - DebugApplications
  - Hangs
  - HeapControlledList
  - LocalDumps
    - BatteryMeter.exe
  - RuntimeExceptionHelperModules
  - WMR
- Windows App Certification Kit
- Windows Azure
- Windows Azure Emulator
- Windows Identity Foundation
- Windows Kits
- Windows Mail

Name	Type	Data
(Default)	REG_SZ	(value not set)
DumpCount	REG_DWORD	0x0000000a (10)
DumpFolder	REG_SZ	C:\temp\dumps
DumpType	REG_DWORD	0x00000002 (2)

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps\BatteryMeter.exe



Minidump File Summary
6/14/2017 3:06:54 PM

Dump Summary

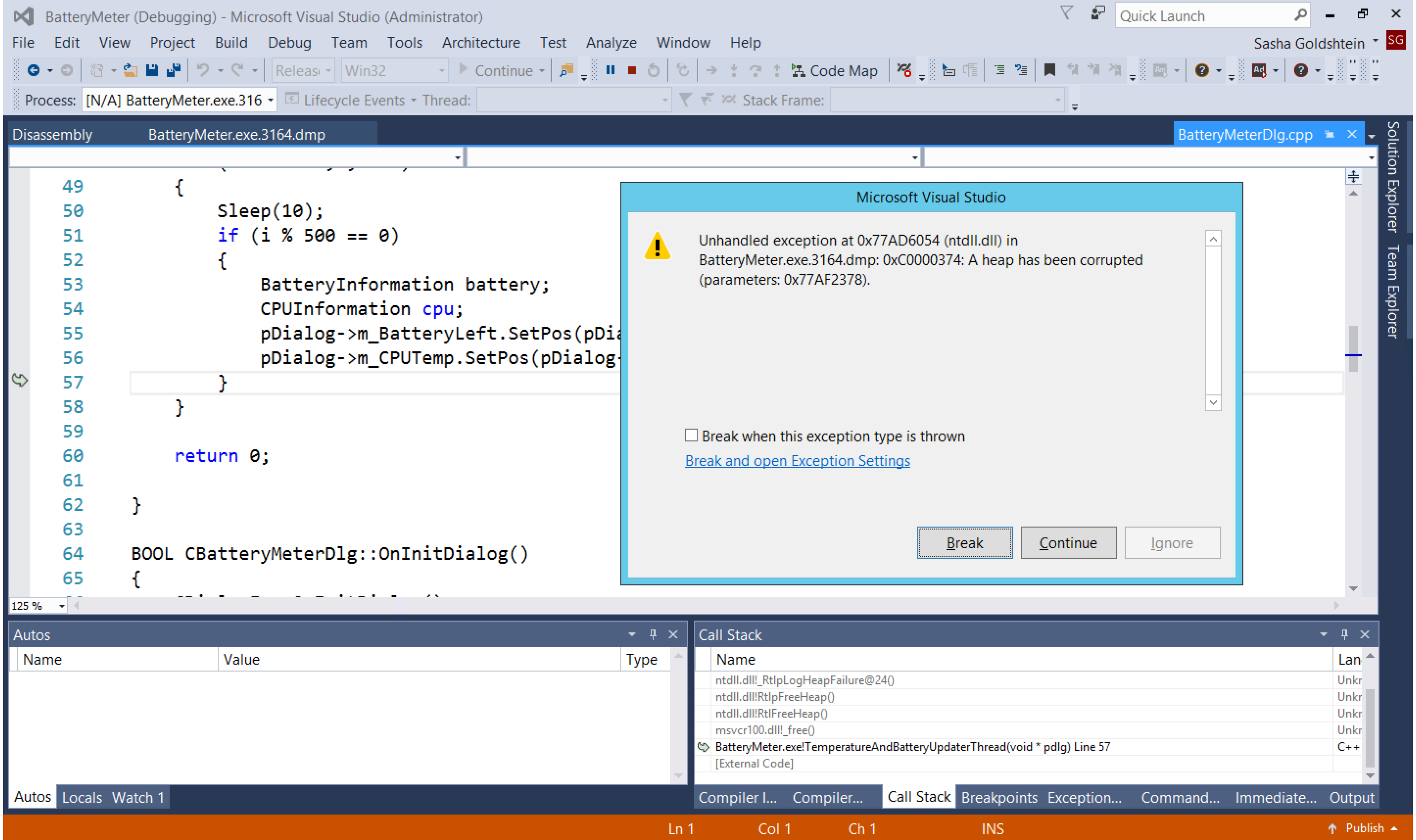
Dump File	BatteryMeter.exe.3164.dmp : C:\temp\dumps\BatteryMeter.exe.3164.dmp
Last Write Time	6/14/2017 3:06:54 PM
Process Name	BatteryMeter.exe : C:\NDC_2017\Crash\BatteryMeter.exe
Process Architecture	x86
Exception Code	0xC0000374
Exception Information	
Heap Information	Present
Error Information	

System Information

OS Version	6.3.9600
CLR Version(s)	

Modules

Module Name	Module Version	Module Path
BatteryMeter.exe	1.0.0.1	C:\NDC_2017\Crash\BatteryMeter.exe
ntdll.dll	6.3.9600.18233	C:\Windows\System32\ntdll.dll
kernel32.dll	6.3.9600.17415	C:\Windows\System32\kernel32.dll
KERNELBASE.dll	6.3.9600.18666	C:\Windows\System32\KERNELBASE.dll
mfc100u.dll	10.0.30319.1	C:\Windows\System32\mfc100u.dll
msvcr100.dll	10.0.40219.1	C:\Windows\System32\msvcr100.dll
user32.dll	6.3.9600.18535	C:\Windows\System32\user32.dll
comctl32.dll	6.10.9600.18006	C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.9600.18006_none_a9ec6aab01...


BatteryMeter (Debugging) - Microsoft Visual Studio (Administrator)
Quick Launch
File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help
Sasha Goldshtein
Release Win32 Continue Code Map
Process: [N/A] BatteryMeter.exe.316 Lifecycle Events Thread: Stack Frame:
Disassembly
BatteryMeter.exe.3164.dmp
BatteryMeterDlg.cpp
Solution Explorer
Team Explorer
125 %
Autos
Name Value Type
Autos Locals Watch 1
Compiler I... Compiler... Call Stack Breakpoints Exception... Command... Immediate... Output
Ln 1 Col 1 Ch 1 INS
Publish

```
    {
        Sleep(10);
        if (i % 500 == 0)
        {
            BatteryInformation battery;
            CPUInformation cpu;
            pDialog->m_BatteryLeft.SetPos(pDia
            pDialog->m_CPUTemp.SetPos(pDialog-
        }
    }

    return 0;

}

BOOL CBatteryMeterDlg::OnInitDialog()
{
```

Microsoft Visual Studio

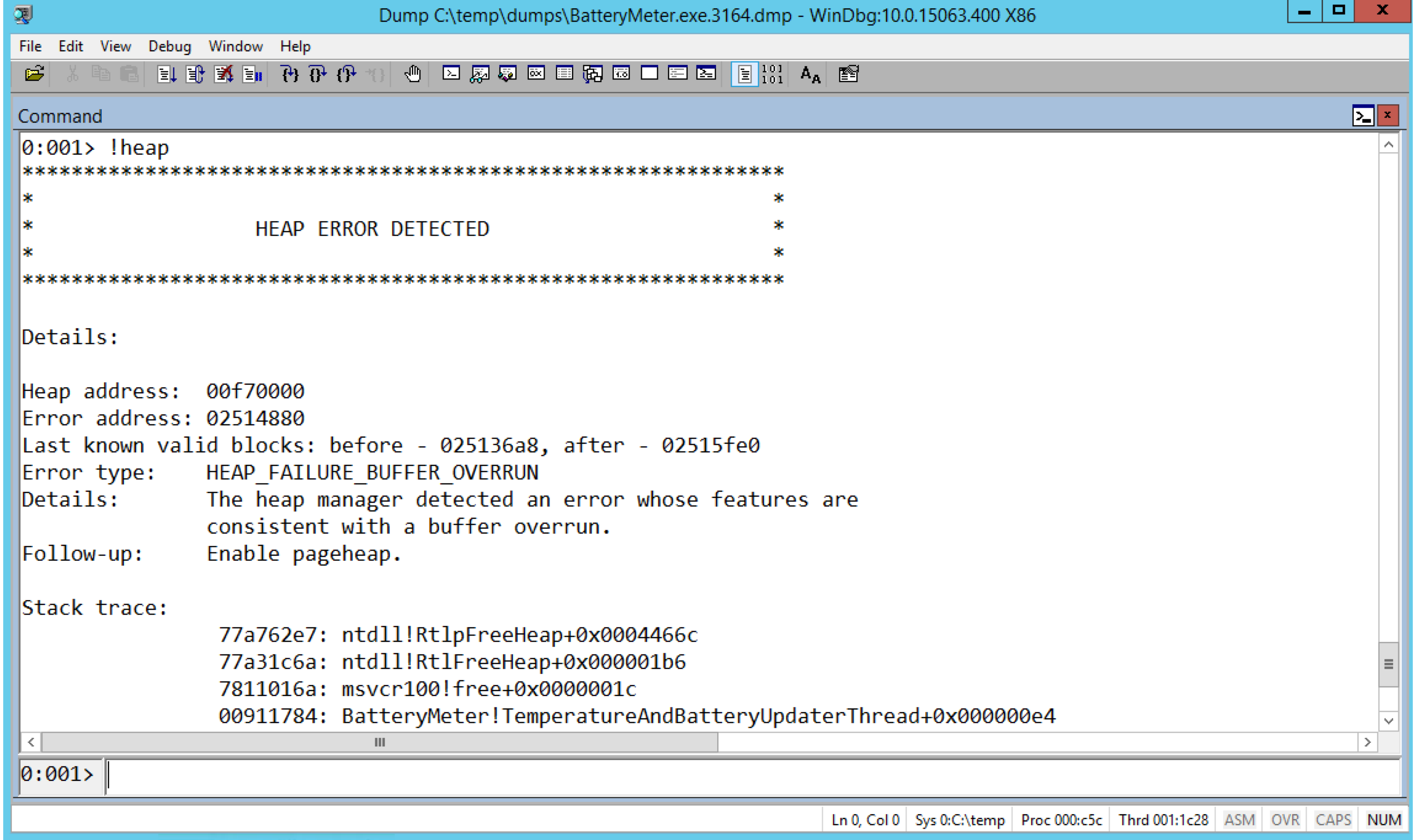
Unhandled exception at 0x77AD6054 (ntdll.dll) in BatteryMeter.exe.3164.dmp: 0xC0000374: A heap has been corrupted (parameters: 0x77AF2378).

☐ Break when this exception type is thrown

Break and open Exception Settings

Call Stack

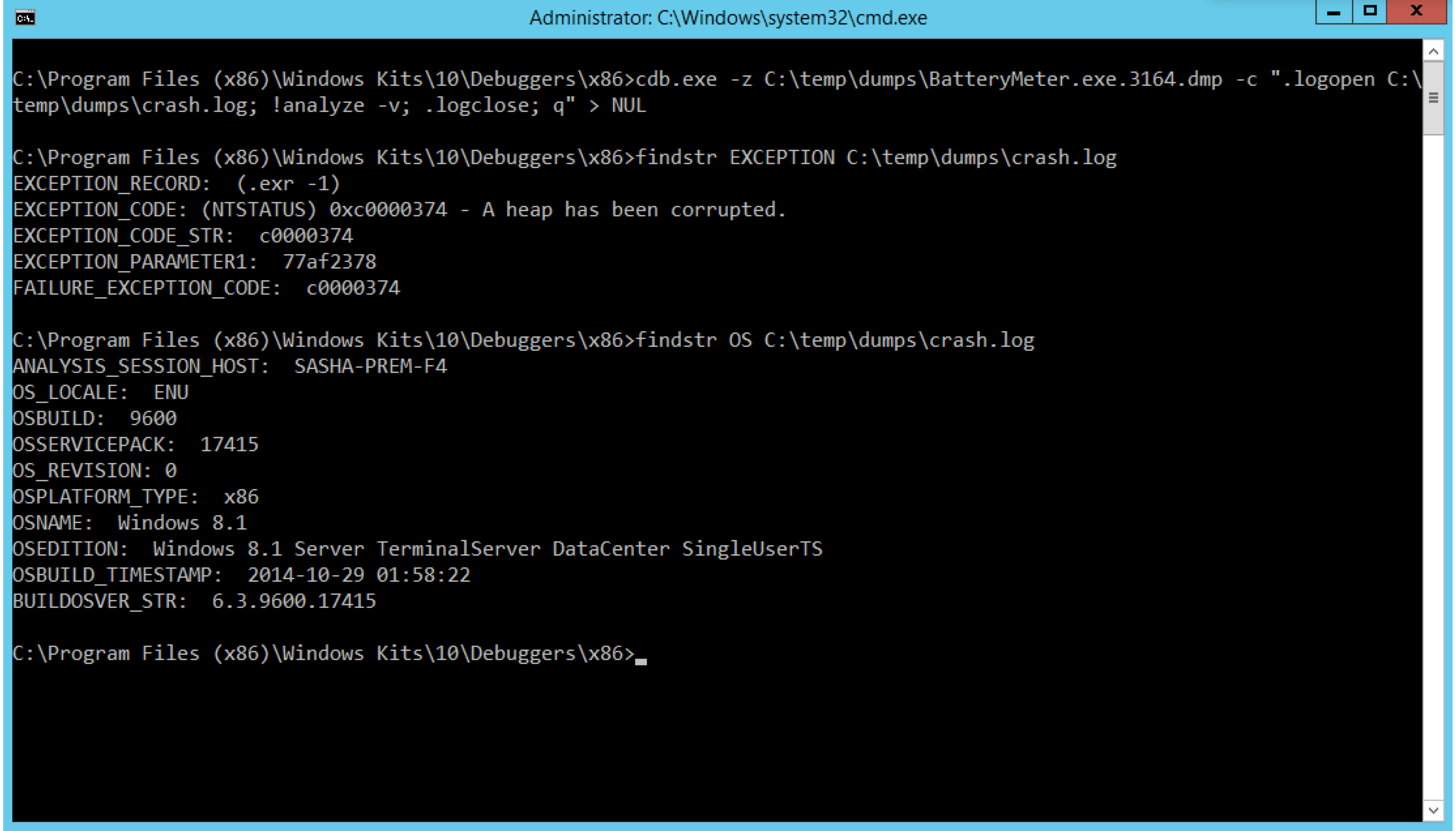
Name	Lan
ntdll.dll!_RtlpLogHeapFailure@24()	Unkr
ntdll.dll!RtlpFreeHeap()	Unkr
ntdll.dll!RtlFreeHeap()	Unkr
msvcr100.dll!_free()	Unkr
BatteryMeter.exe!TemperatureAndBatteryUpdaterThread(void * pdlg) Line 57	C++
[External Code]	


Dump C:\temp\dumps\BatteryMeter.exe.3164.dmp - WinDbg:10.0.15063.400 X86
File Edit View Debug Window Help
Command

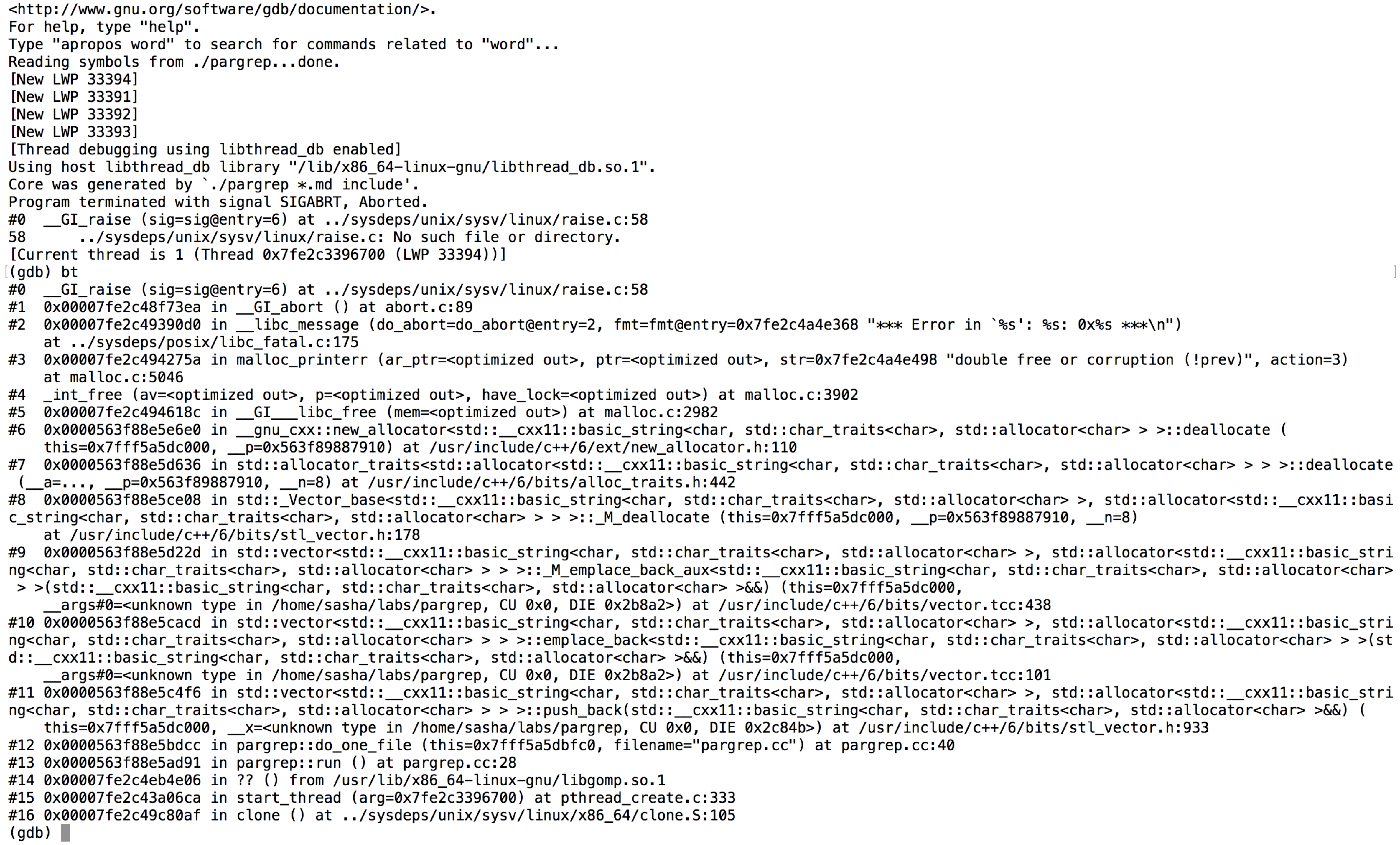
```
0:001> !heap
**************************************************************
*                                                            *
*                  HEAP ERROR DETECTED                       *
*                                                            *
**************************************************************

Details:

Heap address:  00f70000
Error address: 02514880
Last known valid blocks: before - 025136a8, after - 02515fe0
Error type:    HEAP_FAILURE_BUFFER_OVERRUN
Details:       The heap manager detected an error whose features are
               consistent with a buffer overrun.
Follow-up:     Enable pageheap.

Stack trace:
                77a762e7: ntdll!RtlpFreeHeap+0x0004466c
                77a31c6a: ntdll!RtlFreeHeap+0x000001b6
                7811016a: msvcr100!free+0x0000001c
                00911784: BatteryMeter!TemperatureAndBatteryUpdaterThread+0x000000e4
```

```
0:001>
```

Ln 0, Col 0 | Sys 0:C:\temp | Proc 000:c5c | Thrd 001:1c28 | ASM | OVR | CAPS | NUM



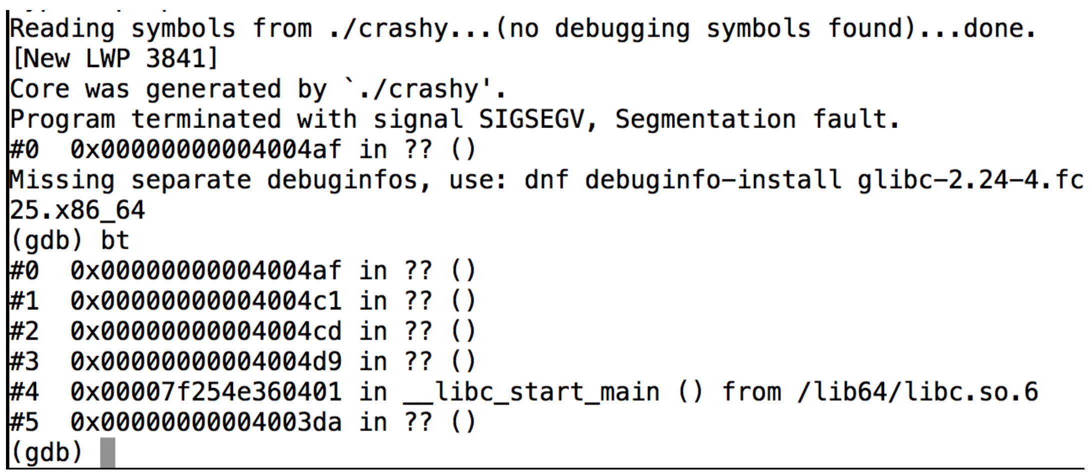
```
C:\Program Files (x86)\Windows Kits\10\Debuggers\x86>cdb.exe -z C:\temp\dumps\BatteryMeter.exe.3164.dmp -c ".logopen C:\
temp\dumps\crash.log; !analyze -v; .logclose; q" > NUL

C:\Program Files (x86)\Windows Kits\10\Debuggers\x86>findstr EXCEPTION C:\temp\dumps\crash.log
EXCEPTION_RECORD:  (.exr -1)
EXCEPTION_CODE: (NTSTATUS) 0xc0000374 - A heap has been corrupted.
EXCEPTION_CODE_STR:  c0000374
EXCEPTION_PARAMETER1:  77af2378
FAILURE_EXCEPTION_CODE:  c0000374

C:\Program Files (x86)\Windows Kits\10\Debuggers\x86>findstr OS C:\temp\dumps\crash.log
ANALYSIS_SESSION_HOST:  SASHA-PREM-F4
OS_LOCALE:  ENU
OSBUILD:  9600
OSSERVICEPACK:  17415
OS_REVISION: 0
OSPLATFORM_TYPE:  x86
OSNAME:  Windows 8.1
OSEDITION:  Windows 8.1 Server TerminalServer DataCenter SingleUserTS
OSBUILD_TIMESTAMP:  2014-10-29 01:58:22
BUILDOSVER_STR:  6.3.9600.17415

C:\Program Files (x86)\Windows Kits\10\Debuggers\x86>_
```



```
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./pargrep...done.
[New LWP 33394]
[New LWP 33391]
[New LWP 33392]
[New LWP 33393]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `./pargrep *.md include'.
Program terminated with signal SIGABRT, Aborted.
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:58
58      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
[Current thread is 1 (Thread 0x7fe2c3396700 (LWP 33394))]
(gdb) bt
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:58
#1  0x00007fe2c48f73ea in __GI_abort () at abort.c:89
#2  0x00007fe2c49390d0 in __libc_message (do_abort=do_abort@entry=2, fmt=fmt@entry=0x7fe2c4a4e368 "*** Error in `%s': %s: 0x%s ***\n")
    at ../sysdeps/posix/libc_fatal.c:175
#3  0x00007fe2c494275a in malloc_printerr (ar_ptr=<optimized out>, ptr=<optimized out>, str=0x7fe2c4a4e498 "double free or corruption (!prev)", action=3)
    at malloc.c:5046
#4  _int_free (av=<optimized out>, p=<optimized out>, have_lock=<optimized out>) at malloc.c:3902
#5  0x00007fe2c494618c in __GI___libc_free (mem=<optimized out>) at malloc.c:2982
#6  0x0000563f88e5e6e0 in __gnu_cxx::new_allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > >::deallocate (
    this=0x7fff5a5dc000, __p=0x563f89887910) at /usr/include/c++/6/ext/new_allocator.h:110
#7  0x0000563f88e5d636 in std::allocator_traits<std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > >::deallocate
 (__a=..., __p=0x563f89887910, __n=8) at /usr/include/c++/6/bits/alloc_traits.h:442
#8  0x0000563f88e5ce08 in std::_Vector_base<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basi
c_string<char, std::char_traits<char>, std::allocator<char> > > >::_M_deallocate (this=0x7fff5a5dc000, __p=0x563f89887910, __n=8)
    at /usr/include/c++/6/bits/stl_vector.h:178
#9  0x0000563f88e5d22d in std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_stri
ng<char, std::char_traits<char>, std::allocator<char> > > >::_M_emplace_back_aux<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>
 > >(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&&) (this=0x7fff5a5dc000,
    __args#0=<unknown type in /home/sasha/labs/pargrep, CU 0x0, DIE 0x2b8a2>) at /usr/include/c++/6/bits/vector.tcc:438
#10 0x0000563f88e5cacd in std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_stri
ng<char, std::char_traits<char>, std::allocator<char> > > >::emplace_back<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > >(st
d::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&&) (this=0x7fff5a5dc000,
    __args#0=<unknown type in /home/sasha/labs/pargrep, CU 0x0, DIE 0x2b8a2>) at /usr/include/c++/6/bits/vector.tcc:101
#11 0x0000563f88e5c4f6 in std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_stri
ng<char, std::char_traits<char>, std::allocator<char> > > >::push_back(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&&) (
    this=0x7fff5a5dc000, __x=<unknown type in /home/sasha/labs/pargrep, CU 0x0, DIE 0x2c84b>) at /usr/include/c++/6/bits/stl_vector.h:933
#12 0x0000563f88e5bdcc in pargrep::do_one_file (this=0x7fff5a5dbfc0, filename="pargrep.cc") at pargrep.cc:40
#13 0x0000563f88e5ad91 in pargrep::run () at pargrep.cc:28
#14 0x00007fe2c4eb4e06 in ?? () from /usr/lib/x86_64-linux-gnu/libgomp.so.1
#15 0x00007fe2c43a06ca in start_thread (arg=0x7fe2c3396700) at pthread_create.c:333
#16 0x00007fe2c49c80af in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:105
(gdb) 
```

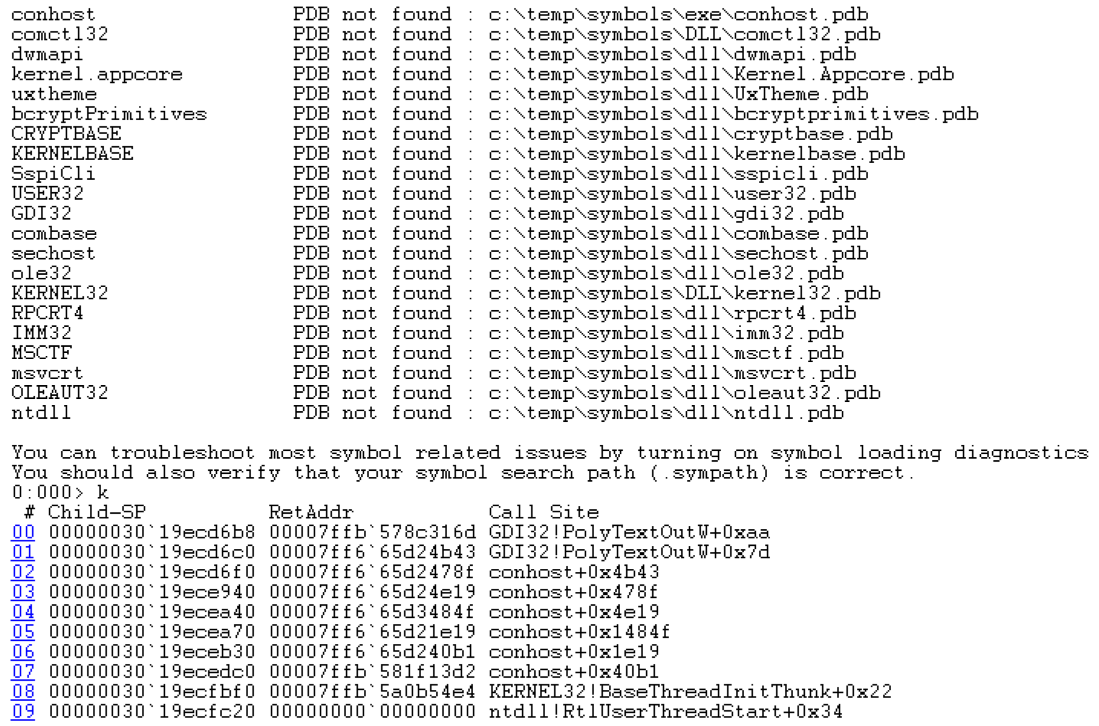
Five Things That Will Happen To You If You Don't Have Symbolic Debug Information

Linux



```
Reading symbols from ./crashy...(no debugging symbols found)...done.
[New LWP 3841]
Core was generated by `./crashy'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x00000000004004af in ?? ()
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.24-4.fc
25.x86_64
(gdb) bt
#0  0x00000000004004af in ?? ()
#1  0x00000000004004c1 in ?? ()
#2  0x00000000004004cd in ?? ()
#3  0x00000000004004d9 in ?? ()
#4  0x00007f254e360401 in __libc_start_main () from /lib64/libc.so.6
#5  0x00000000004003da in ?? ()
(gdb) 
```

Windows



```
conhost                 PDB not found : c:\temp\symbols\exe\conhost.pdb
comctl32                PDB not found : c:\temp\symbols\DLL\comctl32.pdb
dwmapi                  PDB not found : c:\temp\symbols\dll\dwmapi.pdb
kernel.appcore          PDB not found : c:\temp\symbols\dll\Kernel.Appcore.pdb
uxtheme                 PDB not found : c:\temp\symbols\dll\UxTheme.pdb
bcryptPrimitives        PDB not found : c:\temp\symbols\dll\bcryptprimitives.pdb
CRYPTBASE               PDB not found : c:\temp\symbols\dll\cryptbase.pdb
KERNELBASE              PDB not found : c:\temp\symbols\dll\kernelbase.pdb
SspiCli                 PDB not found : c:\temp\symbols\dll\sspicli.pdb
USER32                  PDB not found : c:\temp\symbols\dll\user32.pdb
GDI32                   PDB not found : c:\temp\symbols\dll\gdi32.pdb
combase                 PDB not found : c:\temp\symbols\dll\combase.pdb
sechost                 PDB not found : c:\temp\symbols\dll\sechost.pdb
ole32                   PDB not found : c:\temp\symbols\dll\ole32.pdb
KERNEL32                PDB not found : c:\temp\symbols\DLL\kernel32.pdb
RPCRT4                  PDB not found : c:\temp\symbols\dll\rpcrt4.pdb
IMM32                   PDB not found : c:\temp\symbols\dll\imm32.pdb
MSCTF                   PDB not found : c:\temp\symbols\dll\msctf.pdb
msvcrt                  PDB not found : c:\temp\symbols\dll\msvcrt.pdb
OLEAUT32                PDB not found : c:\temp\symbols\dll\oleaut32.pdb
ntdll                   PDB not found : c:\temp\symbols\dll\ntdll.pdb

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics
You should also verify that your symbol search path (.sympath) is correct.
0:000> k
 # Child-SP          RetAddr           Call Site
00 00000030`19ecd6b8 00007ffb`578c316d GDI32!PolyTextOutW+0xaa
01 00000030`19ecd6c0 00007ff6`65d24b43 GDI32!PolyTextOutW+0x7d
02 00000030`19ecd6f0 00007ff6`65d2478f conhost+0x4b43
03 00000030`19ece940 00007ff6`65d24e19 conhost+0x478f
04 00000030`19ecea40 00007ff6`65d3484f conhost+0x4e19
05 00000030`19ecea70 00007ff6`65d21e19 conhost+0x1484f
06 00000030`19eceb30 00007ff6`65d240b1 conhost+0x1e19
07 00000030`19ecedc0 00007ffb`581f13d2 conhost+0x40b1
08 00000030`19ecfbf0 00007ffb`5a0b54e4 KERNEL32!BaseThreadInitThunk+0x22
09 00000030`19ecfc20 00000000`00000000 ntdll!RtlUserThreadStart+0x34
```

Getting Debug Information

Linux

- Compile with **-g**
  - Separate debuginfo using **objcopy** and **strip**
- Debuginfo packages may be available for your distro:

```
apt install mypkg-dbg
dnf debuginfo-install mypkg
```

Windows

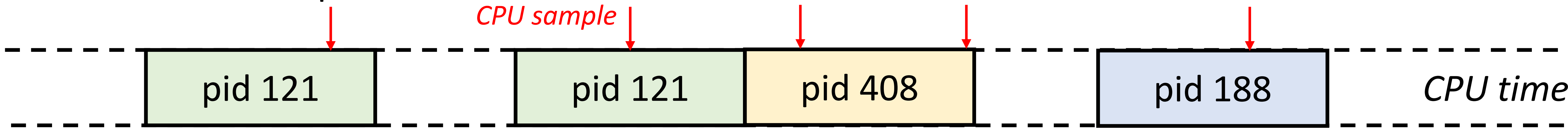
- Compile with **/Zi /DEBUG:FULL**
  - Symbols can be stripped using **pdbcopy** (public vs. private)
- Microsoft public symbol server:

```
setx /m _NT_SYMBOL_PATH
  …http://msdl.microsoft.com/download/symbols
```

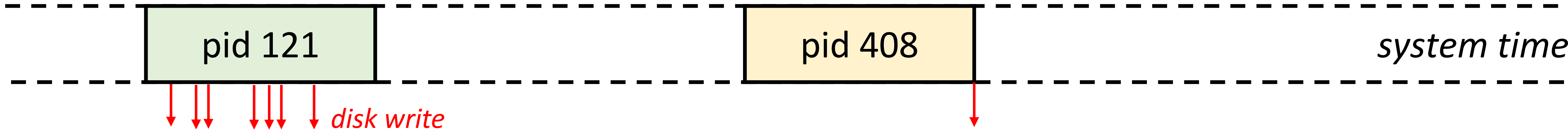
- You can host your own symbol server using **symstore**

Sampling vs. Tracing

- **Sampling** works by getting a snapshot or a call stack every N occurrences of an interesting event
  - For most events, implemented in the PMU using overflow counters and interrupts

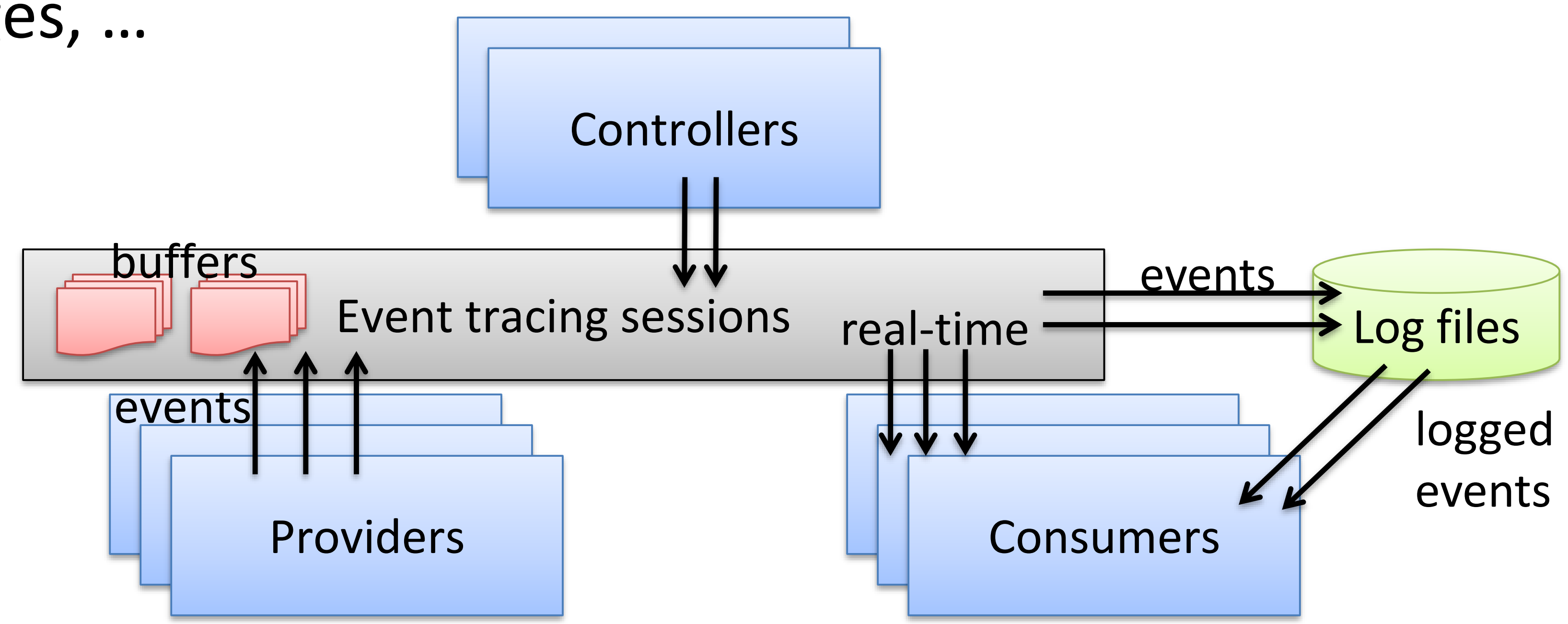


- **Tracing** works by getting a message or a call stack at every occurrence of an interesting event



Event Tracing For Windows

- High-performance facility for emitting 100K+ log events per second with rich payloads and stack trace support
- CPU samples, file accesses, image loads, heap allocs, threads, window messages, …



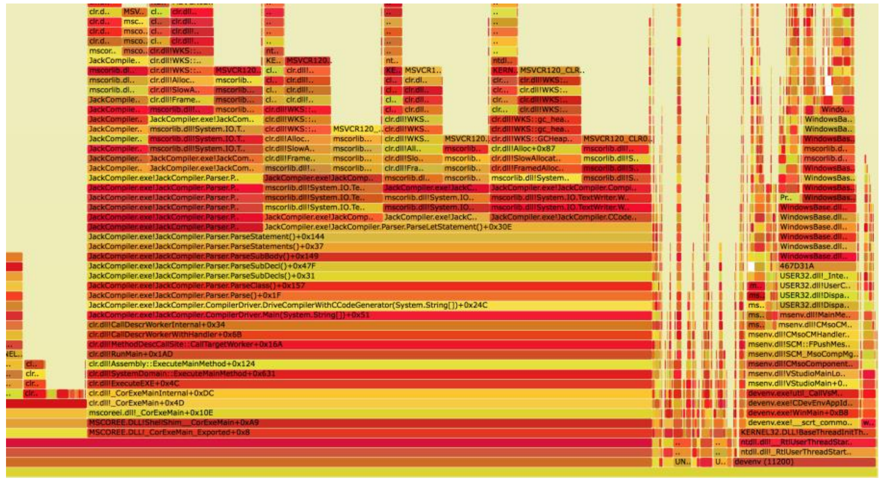
perf

- **perf** is a Linux multi-tool for performance investigations
- Capable of both tracing and sampling
- Developed in the kernel tree, must match running kernel's version

- Debian-based: `apt install linux-tools-common`
- RedHat-based: `yum install perf`

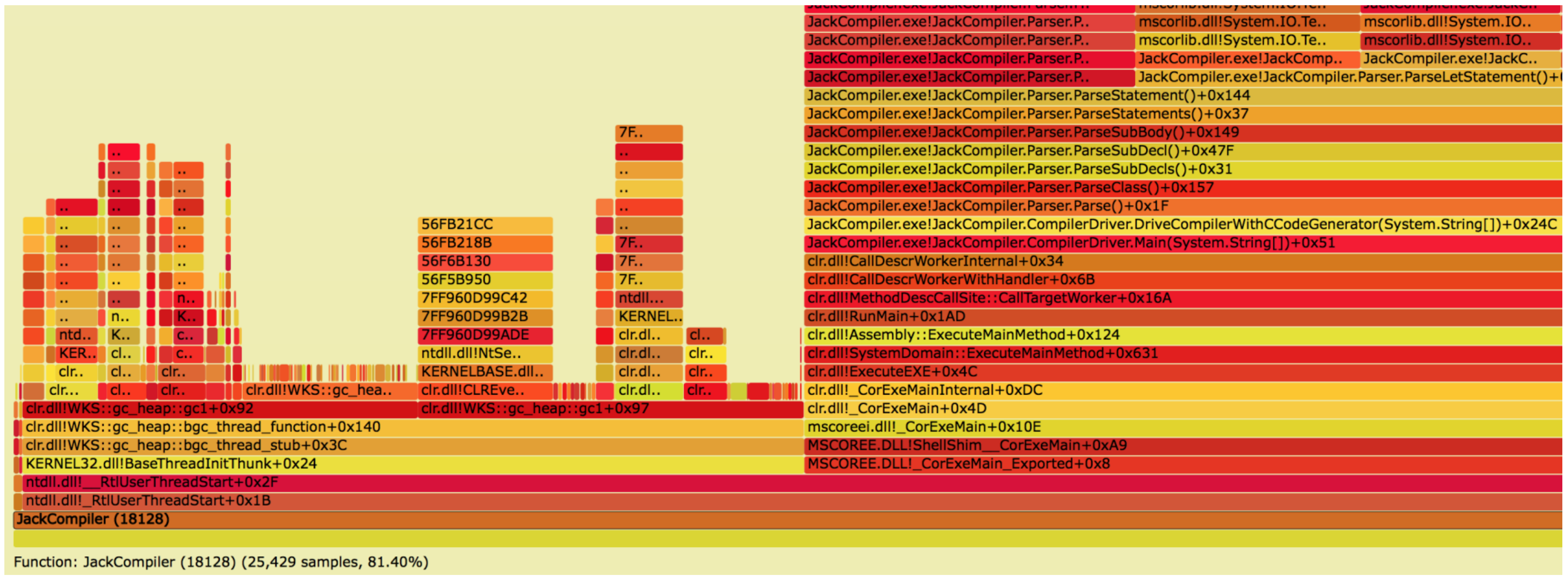
Flame Graphs

- A visualization method (adjacency graph), very useful for stack traces, invented by Brendan Gregg
  - http://www.brendangregg.com/flamegraphs.html
- Turns thousands of stack trace pages into a single interactive graph
- Example scenarios:
  - Identify CPU hotspots on the system/application
  - Show stacks that perform heavy disk accesses
  - Find threads that block for a long time and the stack where they do it



Reading a Flame Graph

- Each rectangle is a function
- Y-axis: caller-callee
- X-axis: sorted stacks (not time)
- Wider frames are more common
- Supports zoom, find
- Filter with grep 😎



Frame Pointer Omission

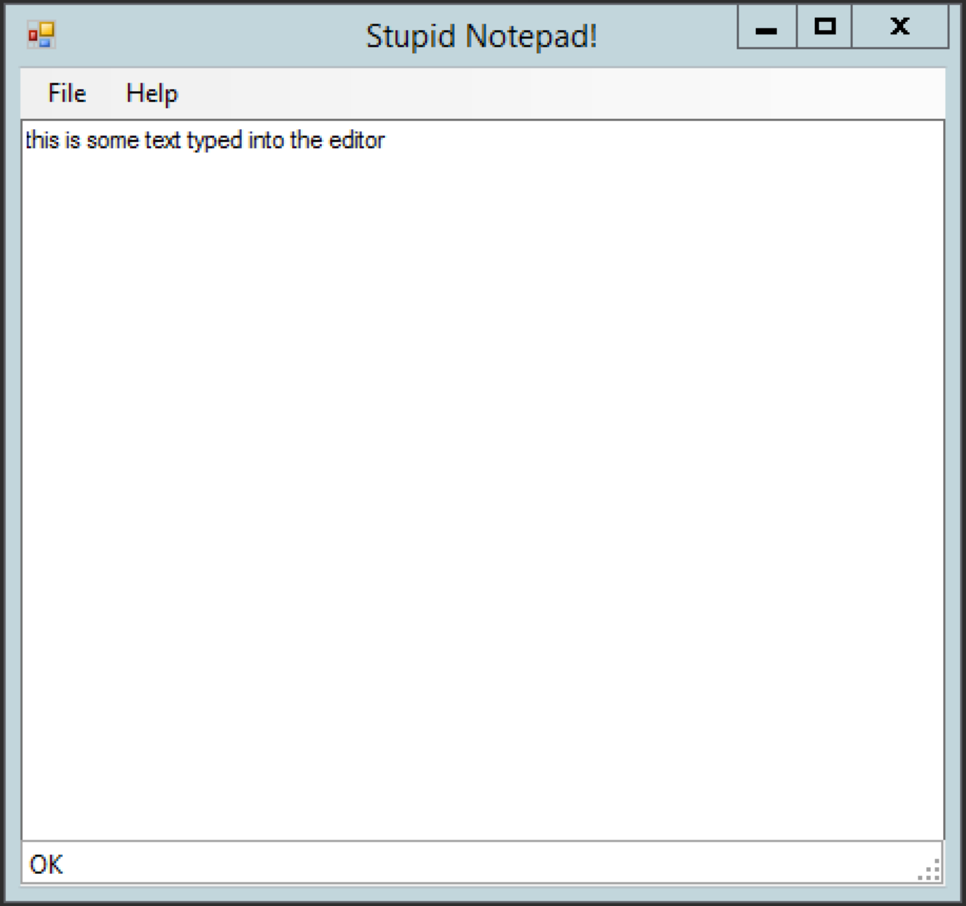
Linux

- Most tools will fail to resolve call stacks if FPO is used
- Given debug information, some stack walkers (e.g. **perf**) can use libunwind to walk FPO stacks
- Disable FPO using **-fno-omit-frame-pointer**

Windows

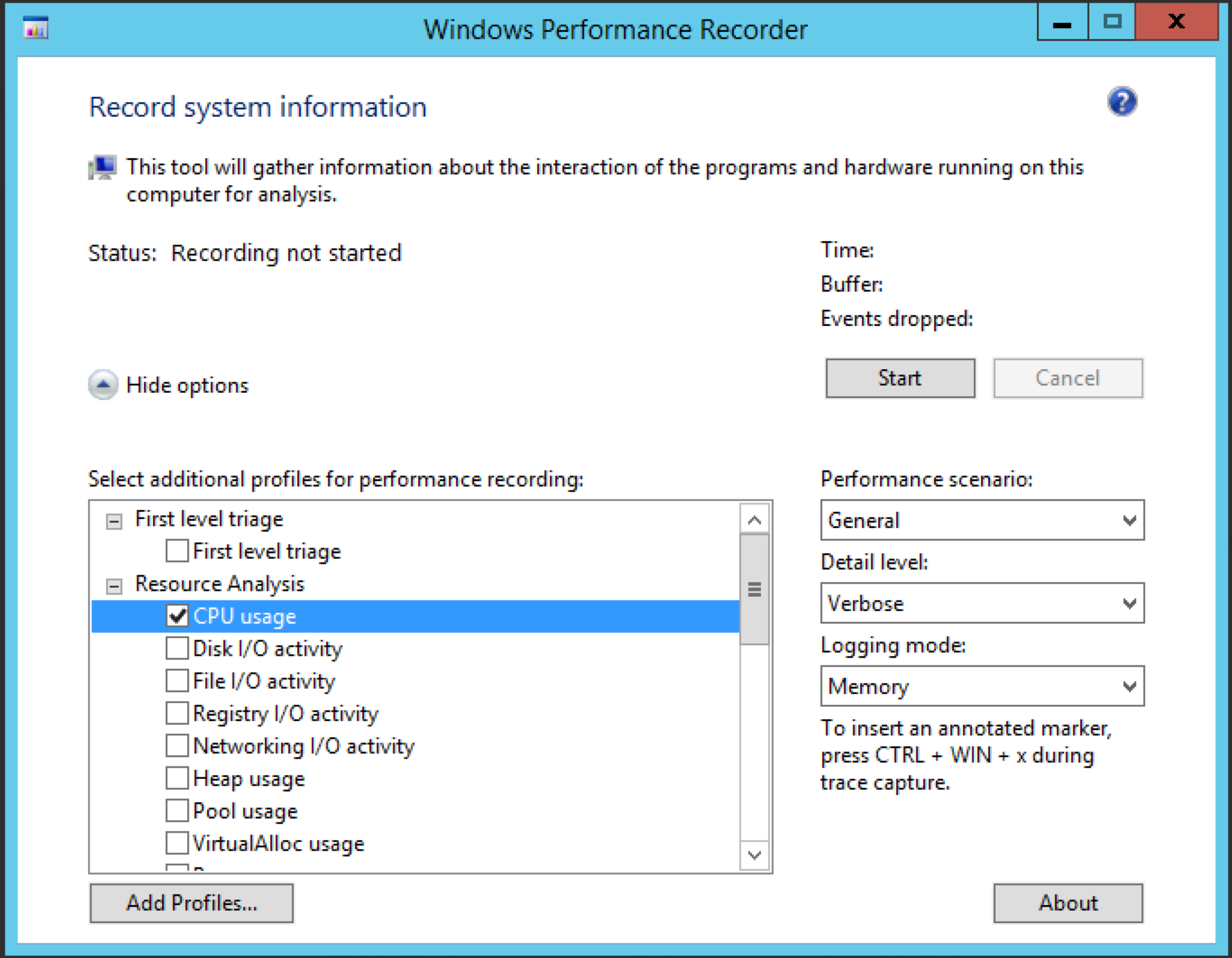
- ETW won’t collect accurate event call stacks if FPO is used
- FPO is turned off by default in Visual C++ (**/Oy-**)

Demo:
CPU Profiling With Flame Graphs



this is some text typed into the editor

OK



Record system information

This tool will gather information about the interaction of the programs and hardware running on this computer for analysis.

Status: Recording not started

Time:
Buffer:
Events dropped:

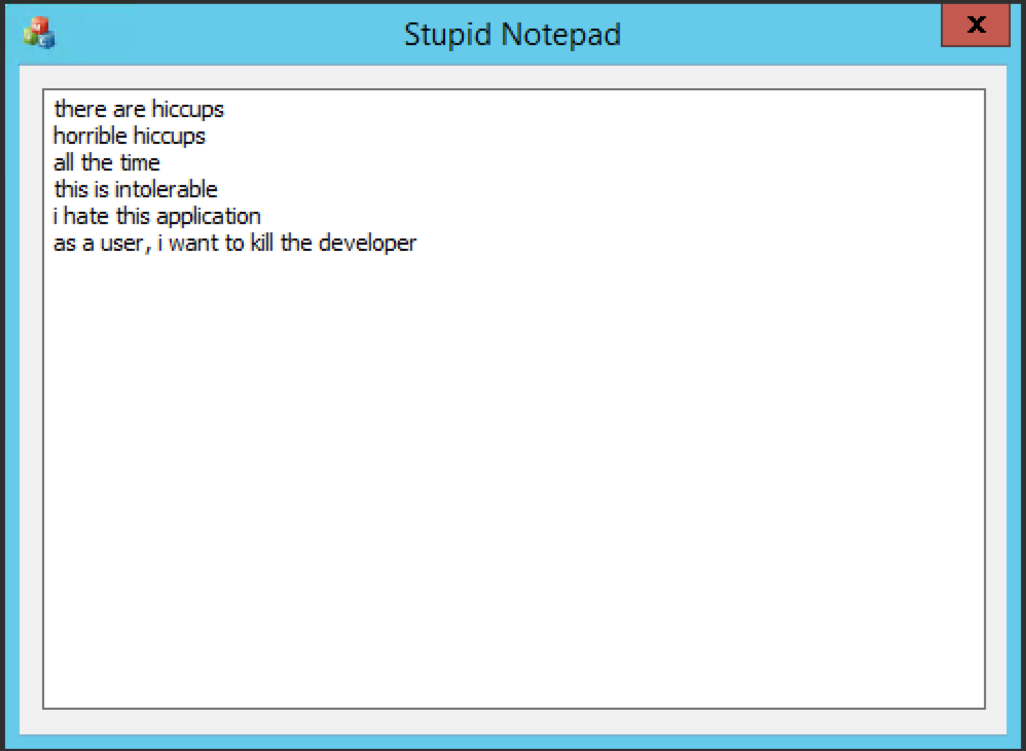
Select additional profiles for performance recording:

- First level triage
  - [ ] First level triage
- Resource Analysis
  - [x] CPU usage
  - [ ] Disk I/O activity
  - [ ] File I/O activity
  - [ ] Registry I/O activity
  - [ ] Networking I/O activity
  - [ ] Heap usage
  - [ ] Pool usage
  - [ ] VirtualAlloc usage

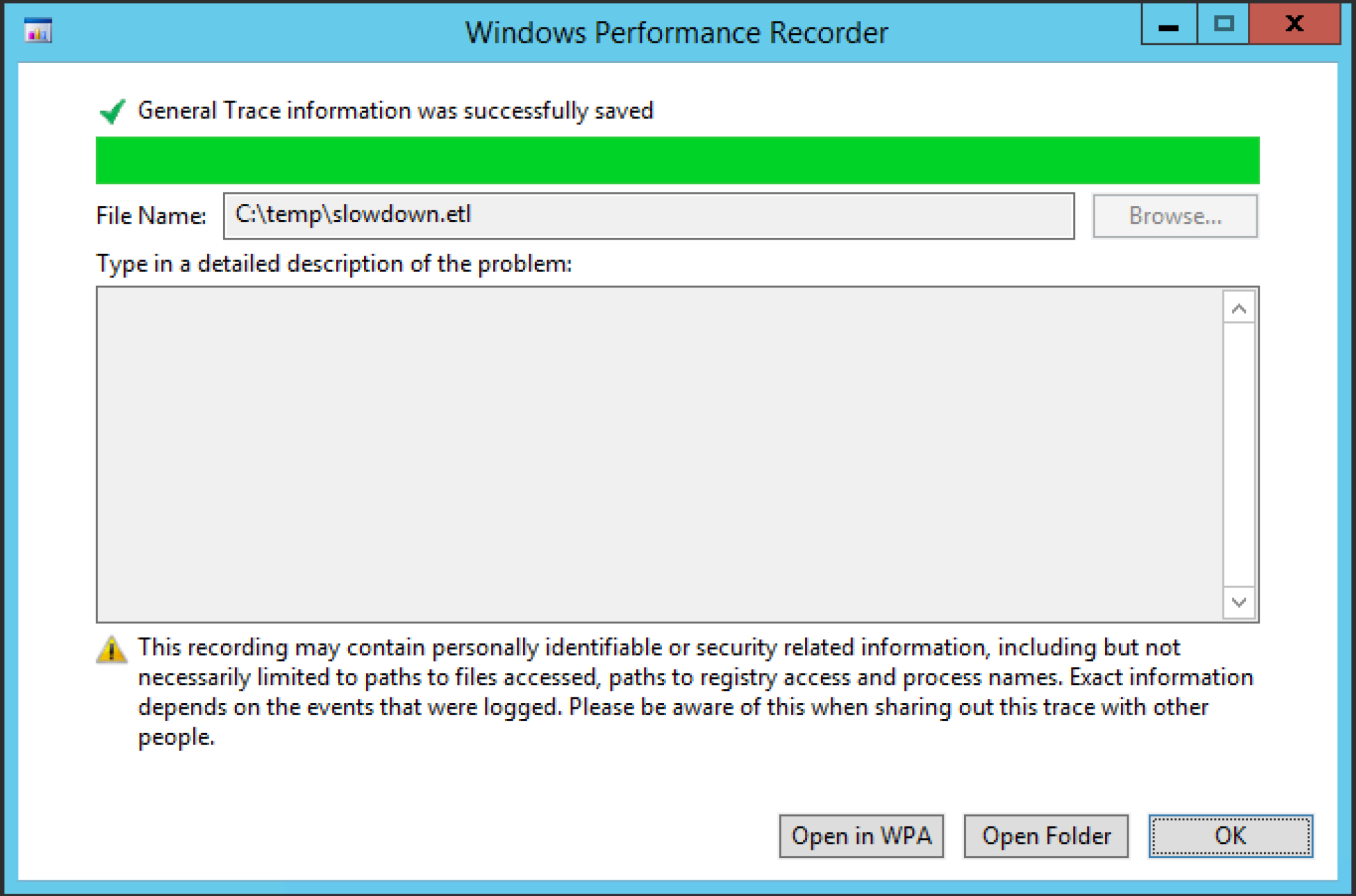
Performance scenario: General
Detail level: Verbose
Logging mode: Memory

To insert an annotated marker, press CTRL + WIN + x during trace capture.

Windows Server 2012 R2


Stupid Notepad

```
there are hiccups
horrible hiccups
all the time
this is intolerable
i hate this application
as a user, i want to kill the developer
```


Windows Performance Recorder

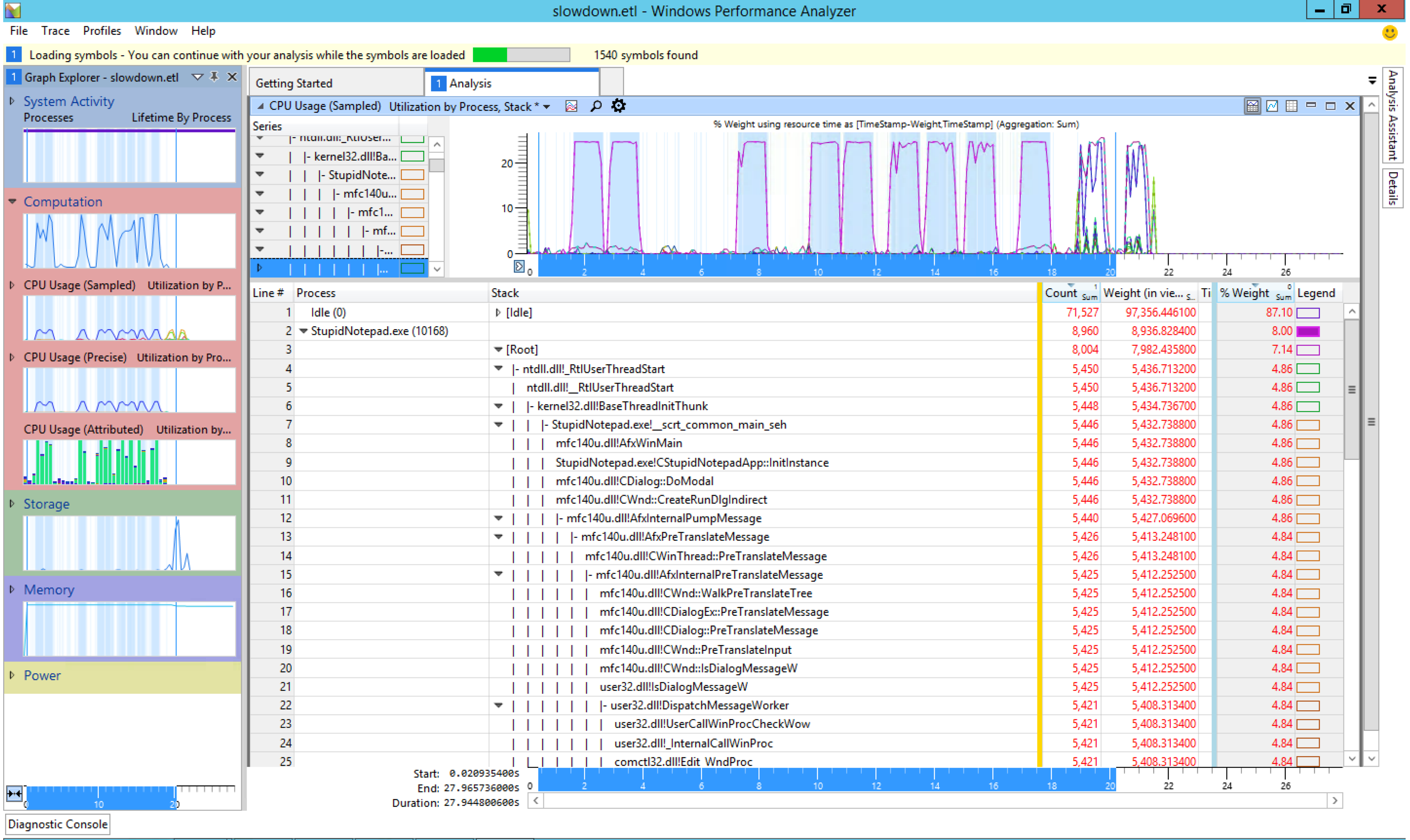
✓ General Trace information was successfully saved

File Name: C:\temp\slowdown.etl

Type in a detailed description of the problem:

This recording may contain personally identifiable or security related information, including but not necessarily limited to paths to files accessed, paths to registry access and process names. Exact information depends on the events that were logged. Please be aware of this when sharing out this trace with other people.

Windows Server 2012 R2


slowdown.etl - Windows Performance Analyzer
File
Trace
Profiles
Window
Help
Graph Explorer - slowdown.etl
System Activity
Processes
Lifetime By Process
Computation
CPU Usage (Sampled) Utilization by P...
CPU Usage (Precise) Utilization by Pro...
CPU Usage (Attributed) Utilization by...
Storage
Memory
Power
Diagnostic Console
Getting Started
Analysis
CPU Usage (Sampled) Utilization by Process, Stack *
Series
Analysis Assistant
Details

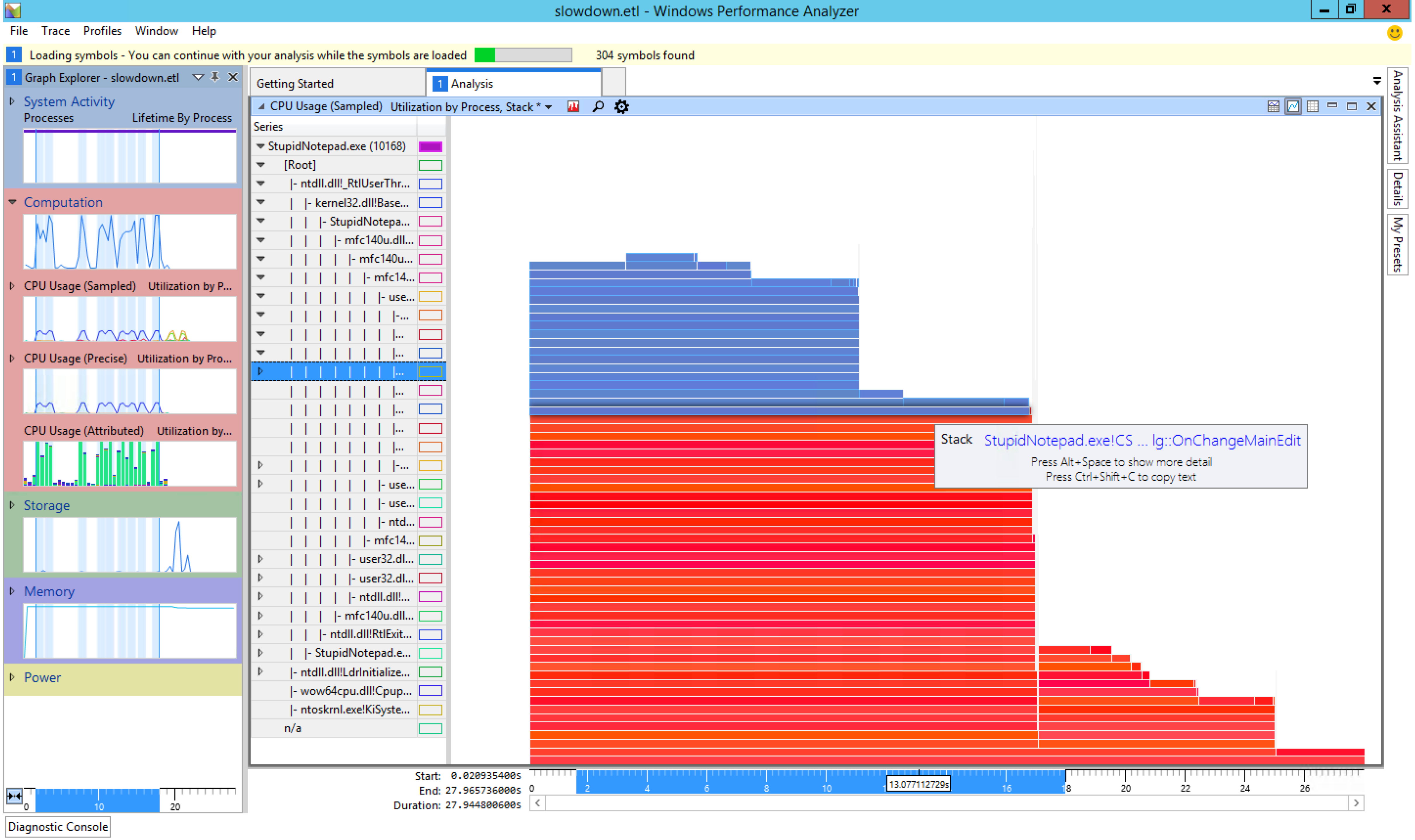
Loading symbols - You can continue with your analysis while the symbols are loaded   1540 symbols found

% Weight using resource time as [TimeStamp-Weight,TimeStamp] (Aggregation: Sum)

```
|- ntdll.dll!_RtlUser...
|  |- kernel32.dll!Ba...
|  |  |- StupidNote...
|  |  |  |- mfc140u...
|  |  |  |  |- mfc1...
|  |  |  |  |  |- mf...
|  |  |  |  |  |  |-...
|  |  |  |  |  |  |...
```

```
Line # Process                      Stack                                                  Count Sum  Weight (in vie... Sum  Ti  % Weight Sum  Legend
1      Idle (0)                     [Idle]                                                    71,527        97,356.446100            87.10
2      StupidNotepad.exe (10168)                                                               8,960         8,936.828400             8.00
3                                   [Root]                                                     8,004         7,982.435800             7.14
4                                   |- ntdll.dll!_RtlUserThreadStart                           5,450         5,436.713200             4.86
5                                   |  ntdll.dll!__RtlUserThreadStart                          5,450         5,436.713200             4.86
6                                   |  |- kernel32.dll!BaseThreadInitThunk                     5,448         5,434.736700             4.86
7                                   |  |  |- StupidNotepad.exe!__scrt_common_main_seh          5,446         5,432.738800             4.86
8                                   |  |  |  mfc140u.dll!AfxWinMain                            5,446         5,432.738800             4.86
9                                   |  |  |  StupidNotepad.exe!CStupidNotepadApp::InitInstance 5,446        5,432.738800             4.86
10                                  |  |  |  mfc140u.dll!CDialog::DoModal                      5,446         5,432.738800             4.86
11                                  |  |  |  mfc140u.dll!CWnd::CreateRunDlgIndirect            5,446         5,432.738800             4.86
12                                  |  |  |  |- mfc140u.dll!AfxInternalPumpMessage             5,440         5,427.069600             4.86
13                                  |  |  |  |  |- mfc140u.dll!AfxPreTranslateMessage          5,426         5,413.248100             4.84
14                                  |  |  |  |  |  mfc140u.dll!CWinThread::PreTranslateMessage 5,426        5,413.248100             4.84
15                                  |  |  |  |  |  |- mfc140u.dll!AfxInternalPreTranslateMessage 5,425      5,412.252500             4.84
16                                  |  |  |  |  |  |  mfc140u.dll!CWnd::WalkPreTranslateTree   5,425         5,412.252500             4.84
17                                  |  |  |  |  |  |  mfc140u.dll!CDialogEx::PreTranslateMessage 5,425       5,412.252500             4.84
18                                  |  |  |  |  |  |  mfc140u.dll!CDialog::PreTranslateMessage 5,425         5,412.252500             4.84
19                                  |  |  |  |  |  |  mfc140u.dll!CWnd::PreTranslateInput      5,425         5,412.252500             4.84
20                                  |  |  |  |  |  |  mfc140u.dll!CWnd::IsDialogMessageW       5,425         5,412.252500             4.84
21                                  |  |  |  |  |  |  user32.dll!IsDialogMessageW              5,425         5,412.252500             4.84
22                                  |  |  |  |  |  |  |- user32.dll!DispatchMessageWorker      5,421         5,408.313400             4.84
23                                  |  |  |  |  |  |  |  user32.dll!UserCallWinProcCheckWow   5,421         5,408.313400             4.84
24                                  |  |  |  |  |  |  |  user32.dll!_InternalCallWinProc      5,421         5,408.313400             4.84
25                                  |  |  |  |  |  |  |  comctl32.dll!Edit_WndProc            5,421         5,408.313400             4.84
```

```
   Start:  0.020935400s
     End: 27.965736000s
Duration: 27.944800600s
```


slowdown.etl - Windows Performance Analyzer

File Trace Profiles Window Help

Loading symbols - You can continue with your analysis while the symbols are loaded 304 symbols found

Graph Explorer - slowdown.etl

System Activity
Processes Lifetime By Process
Computation
CPU Usage (Sampled) Utilization by P...
CPU Usage (Precise) Utilization by Pro...
CPU Usage (Attributed) Utilization by...
Storage
Memory
Power

Getting Started | Analysis

CPU Usage (Sampled) Utilization by Process, Stack *

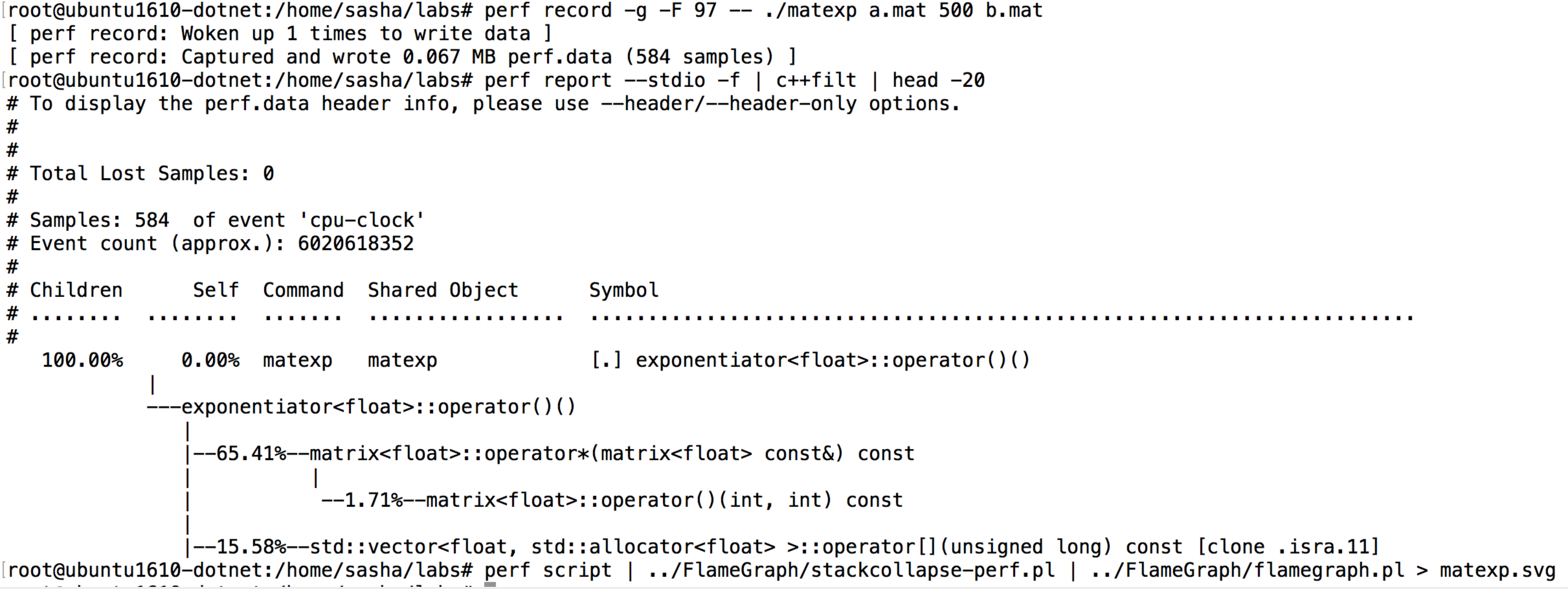
```
Series
StupidNotepad.exe (10168)
  [Root]
    |- ntdll.dll!_RtlUserThr...
    |  |- kernel32.dll!Base...
    |  |  |- StupidNotepa...
    |  |  |  |- mfc140u.dll...
    |  |  |  |  |- mfc140u...
    |  |  |  |  |  |- mfc14...
    |  |  |  |  |  |  |- use...
    |  |  |  |  |  |  |  |-...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |...
    |  |  |  |  |  |  |  |-...
    |  |  |  |  |  |  |- use...
    |  |  |  |  |  |  |- use...
    |  |  |  |  |  |  |- ntd...
    |  |  |  |  |  |- mfc14...
    |  |  |  |  |- user32.dl...
    |  |  |  |  |- user32.dl...
    |  |  |  |  |- ntdll.dll!...
    |  |  |  |- mfc140u.dll...
    |  |  |- ntdll.dll!RtlExit...
    |  |- StupidNotepad.e...
    |- ntdll.dll!LdrInitialize...
    |- wow64cpu.dll!Cpup...
    |- ntoskrnl.exe!KiSyste...
  n/a
```

Stack StupidNotepad.exe!CS ... lg::OnChangeMainEdit
Press Alt+Space to show more detail
Press Ctrl+Shift+C to copy text

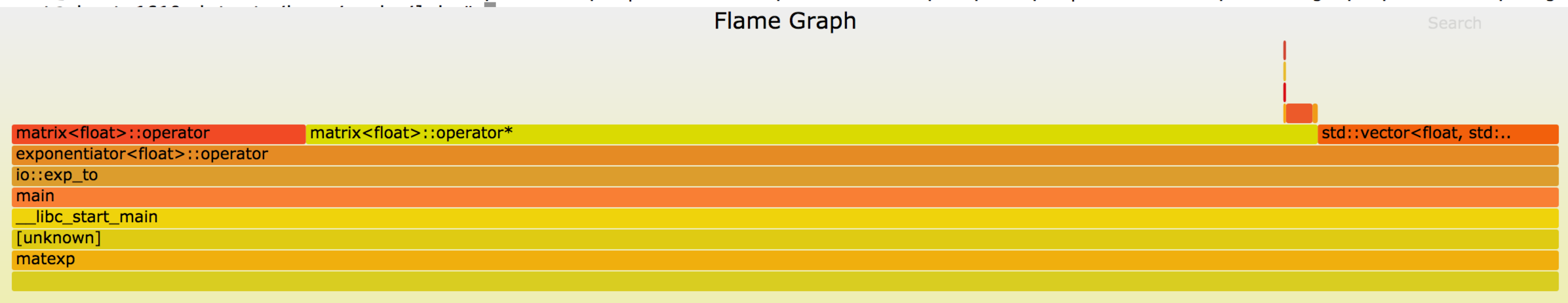
```
Start:    0.020935400s
End:     27.965736000s
Duration: 27.944800600s
```

13.077112729s

Diagnostic Console



```
root@ubuntu1610-dotnet:/home/sasha/labs# perf record -g -F 97 -- ./matexp a.mat 500 b.mat
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.067 MB perf.data (584 samples) ]
root@ubuntu1610-dotnet:/home/sasha/labs# perf report --stdio -f | c++filt | head -20
# To display the perf.data header info, please use --header/--header-only options.
#
#
# Total Lost Samples: 0
#
# Samples: 584  of event 'cpu-clock'
# Event count (approx.): 6020618352
#
# Children      Self  Command  Shared Object      Symbol
# ........  ........  .......  .................  ...................................................................
#
   100.00%     0.00%  matexp   matexp             [.] exponentiator<float>::operator()()
            |
            ---exponentiator<float>::operator()()
               |
               |--65.41%--matrix<float>::operator*(matrix<float> const&) const
               |          |
               |           --1.71%--matrix<float>::operator()(int, int) const
               |
               |--15.58%--std::vector<float, std::allocator<float> >::operator[](unsigned long) const [clone .isra.11]
root@ubuntu1610-dotnet:/home/sasha/labs# perf script | ../FlameGraph/stackcollapse-perf.pl | ../FlameGraph/flamegraph.pl > matexp.svg
```



Memory Leak Analysis

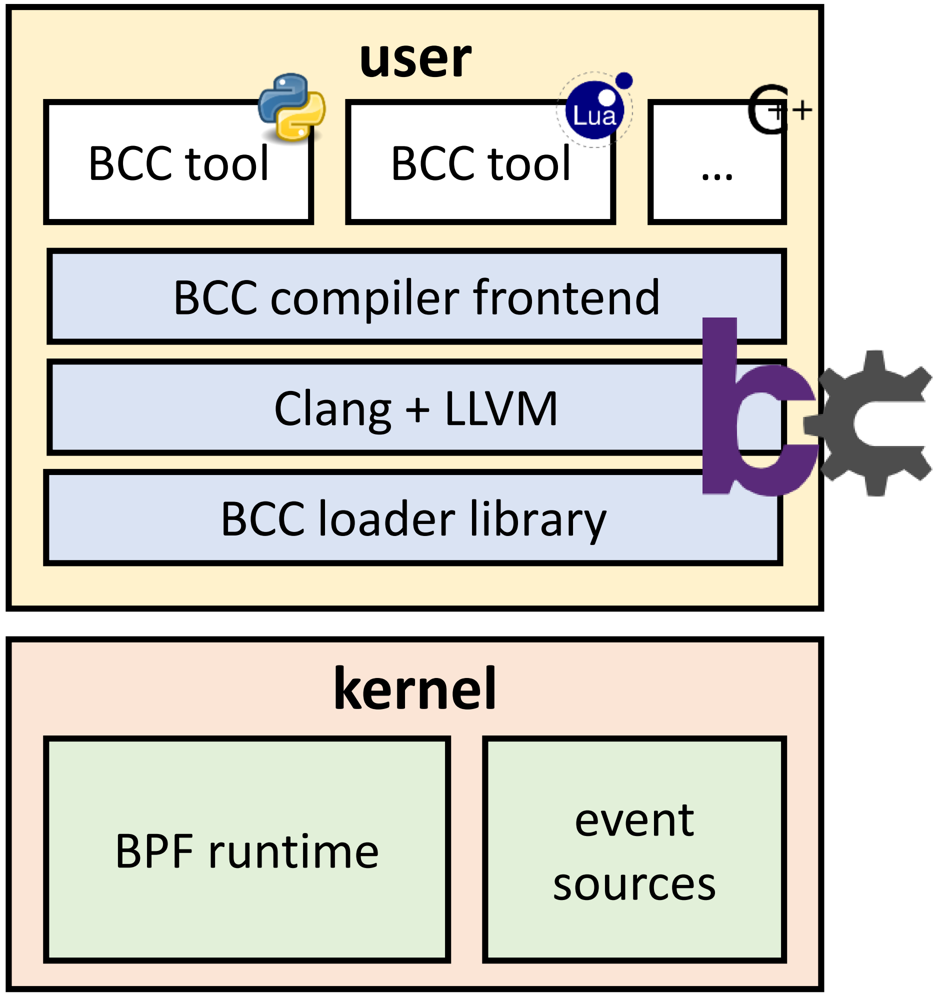
1. Record call stack and size for each allocation (`malloc`)
2. Remove outstanding allocation info for each deallocation (`free`)
3. When desired, dump all outstanding allocation sizes and stacks

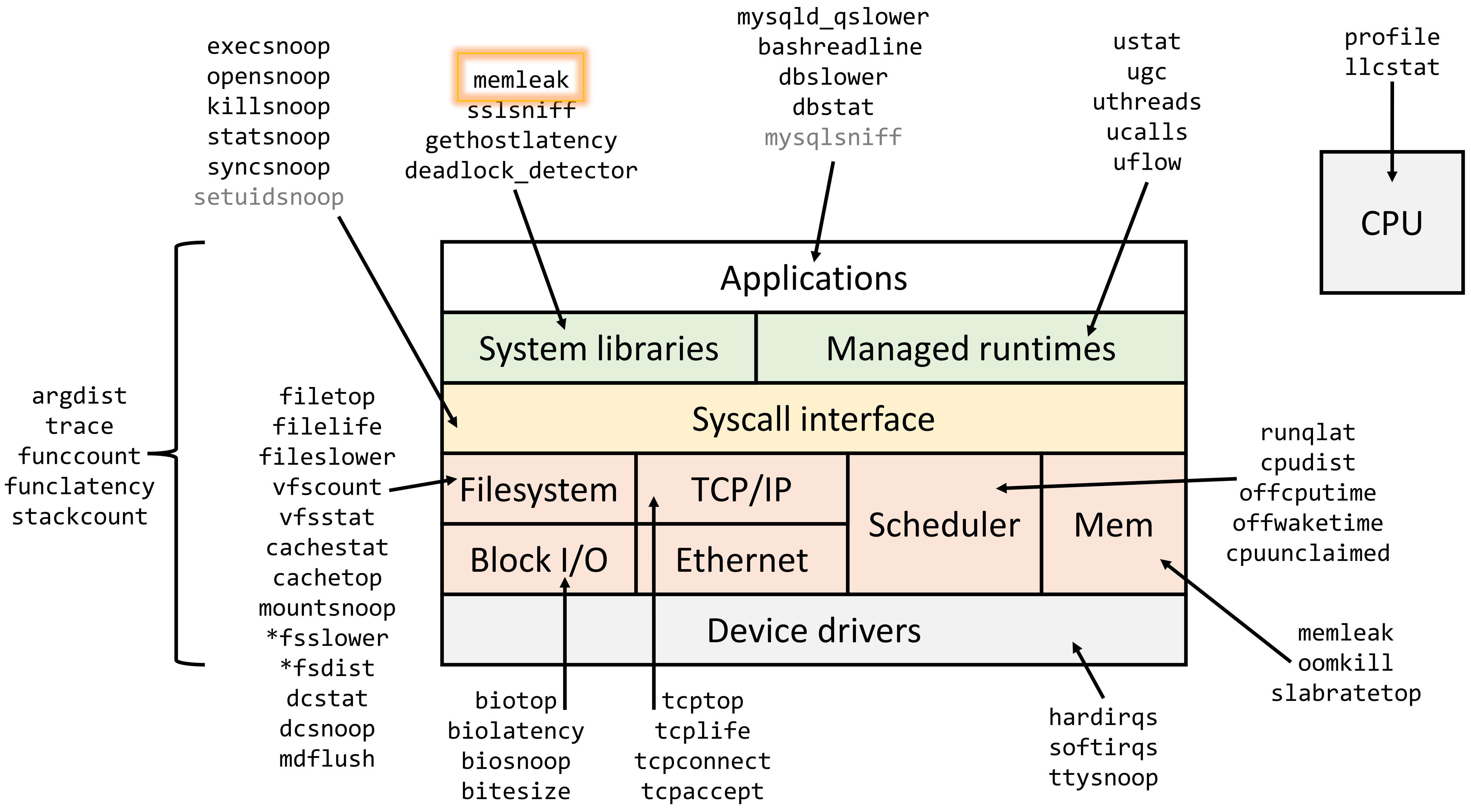
```
[PID 1225 /usr/local/bin/myapp]
  8192 outstanding bytes in 16 allocations from stack:
    __libc_malloc
    operator new
    myapp::factory<factory>::make_factory_factory
    myapp::main
```

- Note: this works for any resource, not just memory

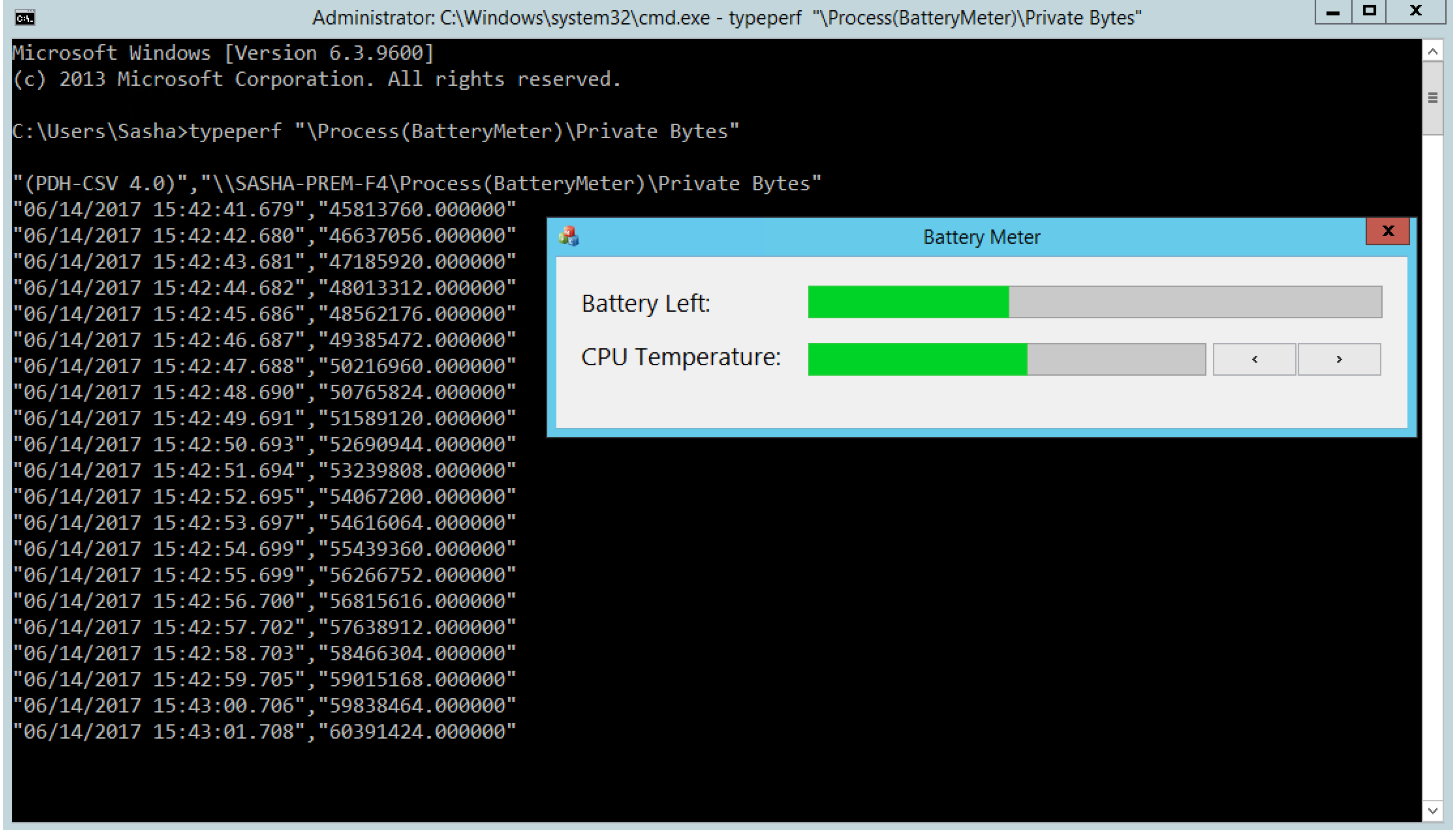
The BCC BPF Front-End

- https://github.com/iovisor/bcc
- BPF Compiler Collection (BCC) is a BPF frontend library and a massive collection of performance tools
  - Contributors from Facebook, PLUMgrid, Netflix, Sela
- Helps build BPF-based tools in high-level languages
  - Python, Lua, C++



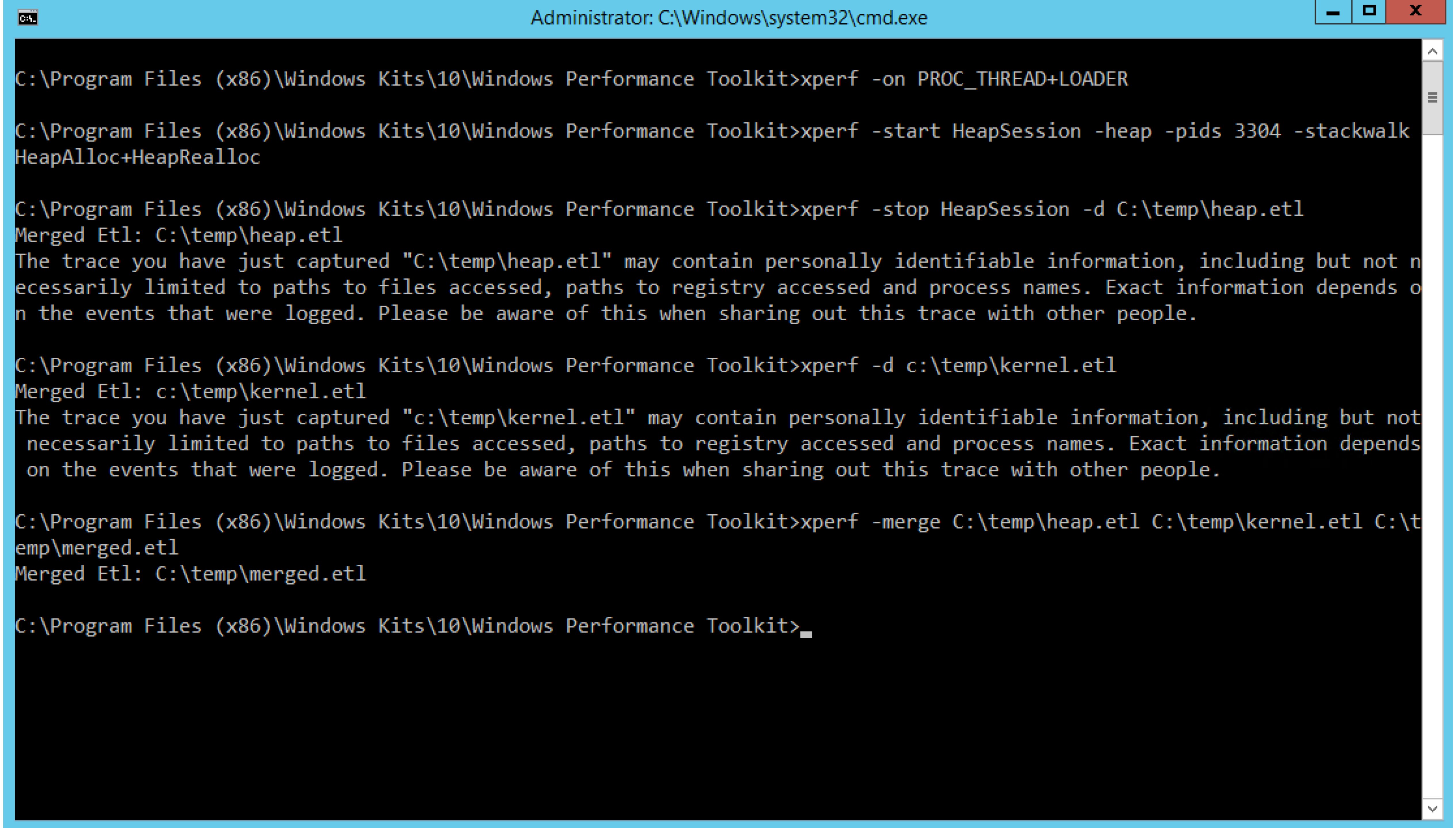

execsnoop
opensnoop
killsnoop
statsnoop
syncsnoop
setuidsnoop
memleak
sslsniff
gethostlatency
deadlock_detector
mysqld_qslower
bashreadline
dbslower
dbstat
mysqlsniff
ustat
ugc
uthreads
ucalls
uflow
profile
llcstat
CPU
Applications
System libraries
Managed runtimes
Syscall interface
Filesystem
TCP/IP
Scheduler
Mem
Block I/O
Ethernet
Device drivers
argdist
trace
funccount
funclatency
stackcount
filetop
filelife
fileslower
vfscount
vfsstat
cachestat
cachetop
mountsnoop
*fsslower
*fsdist
dcstat
dcsnoop
mdflush
runqlat
cpudist
offcputime
offwaketime
cpuunclaimed
biotop
biolatency
biosnoop
bitesize
tcptop
tcplife
tcpconnect
tcpaccept
hardirqs
softirqs
ttysnoop
memleak
oomkill
slabratetop

Demo:
Memory Leak Diagnostics

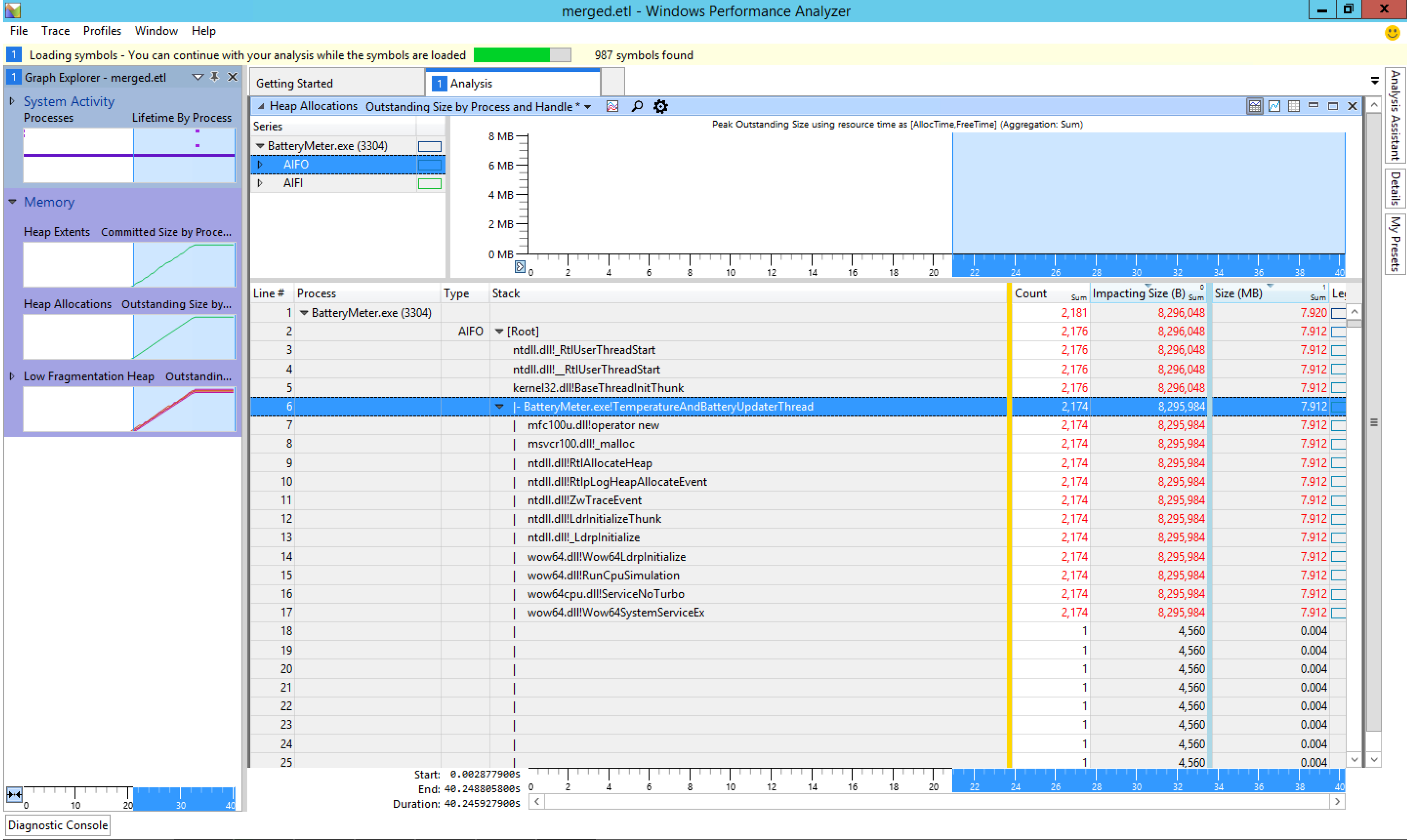


Administrator: C:\Windows\system32\cmd.exe - typeperf "\Process(BatteryMeter)\Private Bytes"

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Sasha>typeperf "\Process(BatteryMeter)\Private Bytes"

"(PDH-CSV 4.0)","\\SASHA-PREM-F4\Process(BatteryMeter)\Private Bytes"
"06/14/2017 15:42:41.679","45813760.000000"
"06/14/2017 15:42:42.680","46637056.000000"
"06/14/2017 15:42:43.681","47185920.000000"
"06/14/2017 15:42:44.682","48013312.000000"
"06/14/2017 15:42:45.686","48562176.000000"
"06/14/2017 15:42:46.687","49385472.000000"
"06/14/2017 15:42:47.688","50216960.000000"
"06/14/2017 15:42:48.690","50765824.000000"
"06/14/2017 15:42:49.691","51589120.000000"
"06/14/2017 15:42:50.693","52690944.000000"
"06/14/2017 15:42:51.694","53239808.000000"
"06/14/2017 15:42:52.695","54067200.000000"
"06/14/2017 15:42:53.697","54616064.000000"
"06/14/2017 15:42:54.699","55439360.000000"
"06/14/2017 15:42:55.699","56266752.000000"
"06/14/2017 15:42:56.700","56815616.000000"
"06/14/2017 15:42:57.702","57638912.000000"
"06/14/2017 15:42:58.703","58466304.000000"
"06/14/2017 15:42:59.705","59015168.000000"
"06/14/2017 15:43:00.706","59838464.000000"
"06/14/2017 15:43:01.708","60391424.000000"
```


Administrator: C:\Windows\system32\cmd.exe

```
C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>xperf -on PROC_THREAD+LOADER

C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>xperf -start HeapSession -heap -pids 3304 -stackwalk
HeapAlloc+HeapRealloc

C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>xperf -stop HeapSession -d C:\temp\heap.etl
Merged Etl: C:\temp\heap.etl
The trace you have just captured "C:\temp\heap.etl" may contain personally identifiable information, including but not n
ecessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends o
n the events that were logged. Please be aware of this when sharing out this trace with other people.

C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>xperf -d c:\temp\kernel.etl
Merged Etl: c:\temp\kernel.etl
The trace you have just captured "c:\temp\kernel.etl" may contain personally identifiable information, including but not
 necessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends
 on the events that were logged. Please be aware of this when sharing out this trace with other people.

C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>xperf -merge C:\temp\heap.etl C:\temp\kernel.etl C:\t
emp\merged.etl
Merged Etl: C:\temp\merged.etl

C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit>_
```



Loading symbols - You can continue with your analysis while the symbols are loaded 987 symbols found

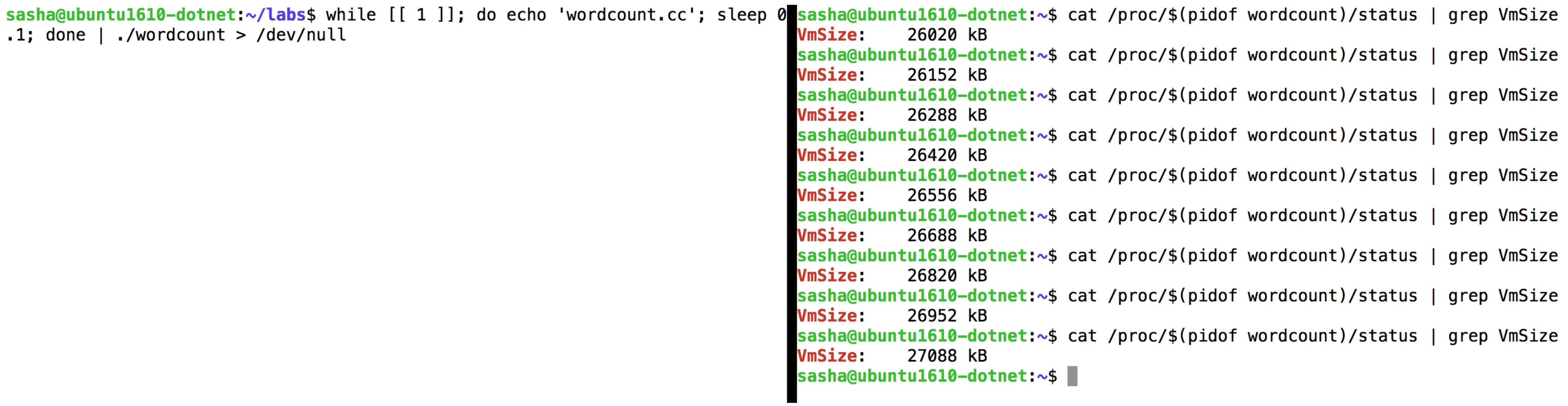
Heap Allocations Outstanding Size by Process and Handle *

Peak Outstanding Size using resource time as [AllocTime,FreeTime] (Aggregation: Sum)

Series
BatteryMeter.exe (3304)
AIFO
AIFI

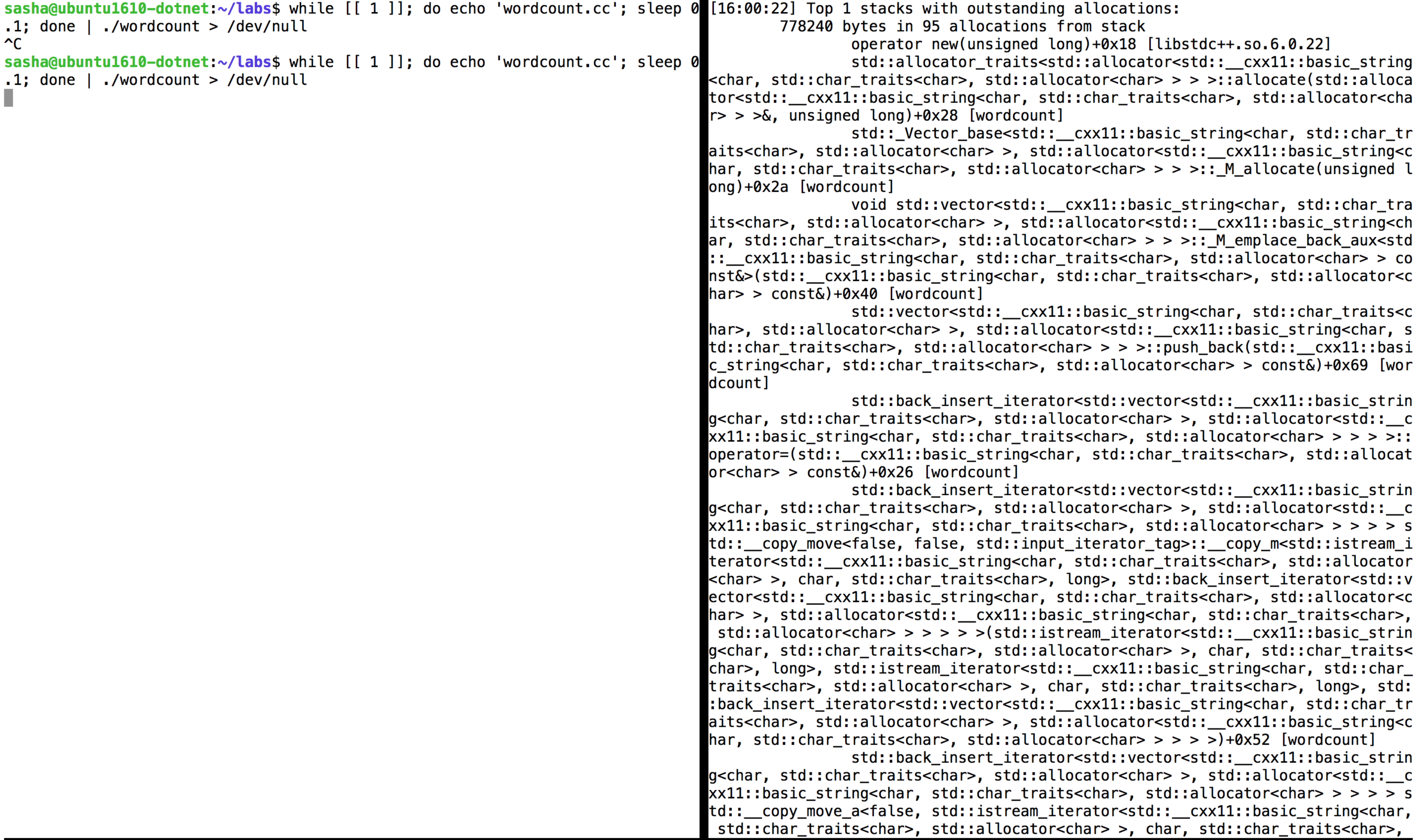
Line #	Process	Type	Stack	Count Sum	Impacting Size (B) Sum	Size (MB) Sum
1	BatteryMeter.exe (3304)			2,181	8,296,048	7.920
2		AIFO	[Root]	2,176	8,296,048	7.912
3			ntdll.dll!_RtlUserThreadStart	2,176	8,296,048	7.912
4			ntdll.dll!__RtlUserThreadStart	2,176	8,296,048	7.912
5			kernel32.dll!BaseThreadInitThunk	2,176	8,296,048	7.912
6			|- BatteryMeter.exe!TemperatureAndBatteryUpdaterThread	2,174	8,295,984	7.912
7			| mfc100u.dll!operator new	2,174	8,295,984	7.912
8			| msvcr100.dll!_malloc	2,174	8,295,984	7.912
9			| ntdll.dll!RtlAllocateHeap	2,174	8,295,984	7.912
10			| ntdll.dll!RtlpLogHeapAllocateEvent	2,174	8,295,984	7.912
11			| ntdll.dll!ZwTraceEvent	2,174	8,295,984	7.912
12			| ntdll.dll!LdrInitializeThunk	2,174	8,295,984	7.912
13			| ntdll.dll!_LdrpInitialize	2,174	8,295,984	7.912
14			| wow64.dll!Wow64LdrpInitialize	2,174	8,295,984	7.912
15			| wow64.dll!RunCpuSimulation	2,174	8,295,984	7.912
16			| wow64cpu.dll!ServiceNoTurbo	2,174	8,295,984	7.912
17			| wow64.dll!Wow64SystemServiceEx	2,174	8,295,984	7.912
18			|	1	4,560	0.004
19			|	1	4,560	0.004
20			|	1	4,560	0.004
21			|	1	4,560	0.004
22			|	1	4,560	0.004
23			|	1	4,560	0.004
24			|	1	4,560	0.004
25			|	1	4,560	0.004

Start: 0.002877900s
End: 40.248805800s
Duration: 40.245927900s



```
sasha@ubuntu1610-dotnet:~/labs$ while [[ 1 ]]; do echo 'wordcount.cc'; sleep 0
.1; done | ./wordcount > /dev/null
```

```
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26020 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26152 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26288 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26420 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26556 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26688 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26820 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   26952 kB
sasha@ubuntu1610-dotnet:~$ cat /proc/$(pidof wordcount)/status | grep VmSize
VmSize:	   27088 kB
sasha@ubuntu1610-dotnet:~$ 
```

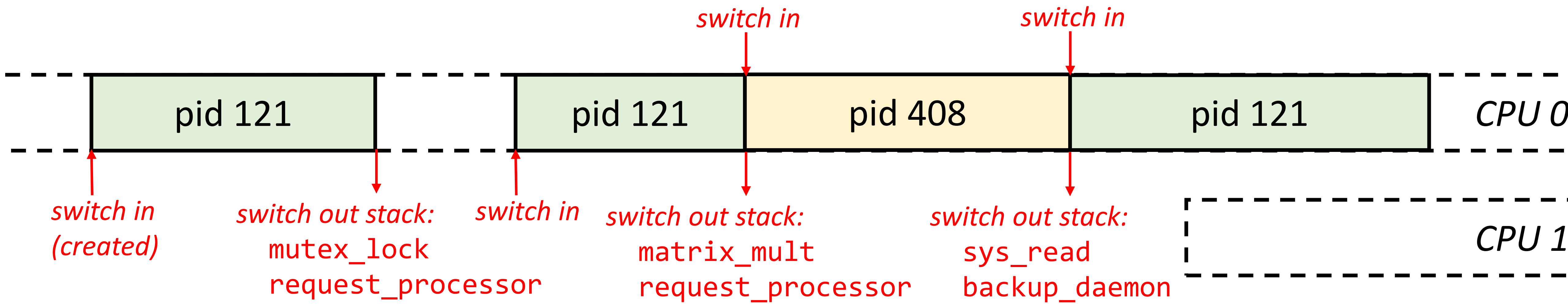


```
sasha@ubuntu1610-dotnet:~/labs$ while [[ 1 ]]; do echo 'wordcount.cc'; sleep 0
.1; done | ./wordcount > /dev/null
^C
sasha@ubuntu1610-dotnet:~/labs$ while [[ 1 ]]; do echo 'wordcount.cc'; sleep 0
.1; done | ./wordcount > /dev/null
```

```
[16:00:22] Top 1 stacks with outstanding allocations:
        778240 bytes in 95 allocations from stack
                operator new(unsigned long)+0x18 [libstdc++.so.6.0.22]
                std::allocator_traits<std::allocator<std::__cxx11::basic_string
<char, std::char_traits<char>, std::allocator<char> > > >::allocate(std::alloca
tor<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<cha
r> > >&, unsigned long)+0x28 [wordcount]
                std::_Vector_base<std::__cxx11::basic_string<char, std::char_tr
aits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<c
har, std::char_traits<char>, std::allocator<char> > > >::_M_allocate(unsigned l
ong)+0x2a [wordcount]
                void std::vector<std::__cxx11::basic_string<char, std::char_tra
its<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<ch
ar, std::char_traits<char>, std::allocator<char> > > >::_M_emplace_back_aux<std
::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > co
nst&>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<c
har> > const&)+0x40 [wordcount]
                std::vector<std::__cxx11::basic_string<char, std::char_traits<c
har>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char, s
td::char_traits<char>, std::allocator<char> > > >::push_back(std::__cxx11::basi
c_string<char, std::char_traits<char>, std::allocator<char> > const&)+0x69 [wor
dcount]
                std::back_insert_iterator<std::vector<std::__cxx11::basic_strin
g<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__c
xx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > >::
operator=(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocat
or<char> > const&)+0x26 [wordcount]
                std::back_insert_iterator<std::vector<std::__cxx11::basic_strin
g<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__c
xx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > > s
td::__copy_move<false, false, std::input_iterator_tag>::__copy_m<std::istream_i
terator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator
<char> >, char, std::char_traits<char>, long>, std::back_insert_iterator<std::v
ector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<c
har> >, std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>,
 std::allocator<char> > > > > >(std::istream_iterator<std::__cxx11::basic_strin
g<char, std::char_traits<char>, std::allocator<char> >, char, std::char_traits<
char>, long>, std::istream_iterator<std::__cxx11::basic_string<char, std::char_
traits<char>, std::allocator<char> >, char, std::char_traits<char>, long>, std:
:back_insert_iterator<std::vector<std::__cxx11::basic_string<char, std::char_tr
aits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<c
har, std::char_traits<char>, std::allocator<char> > > > >)+0x52 [wordcount]
                std::back_insert_iterator<std::vector<std::__cxx11::basic_strin
g<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__c
xx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > > s
td::__copy_move_a<false, std::istream_iterator<std::__cxx11::basic_string<char,
 std::char_traits<char>, std::allocator<char> >, char, std::char_traits<char>,
```

Blocked Thread Investigation

- CPU sampling only identifies time spent on-CPU
- Blocked time is a concern for most applications
  - Sleep, wait, lock, disk, network, database, …
- Blocked time can be traced using context switch events
  - Windows ETW flag **CSwitch**, Linux kernel tracepoint **sched:sched_switch**



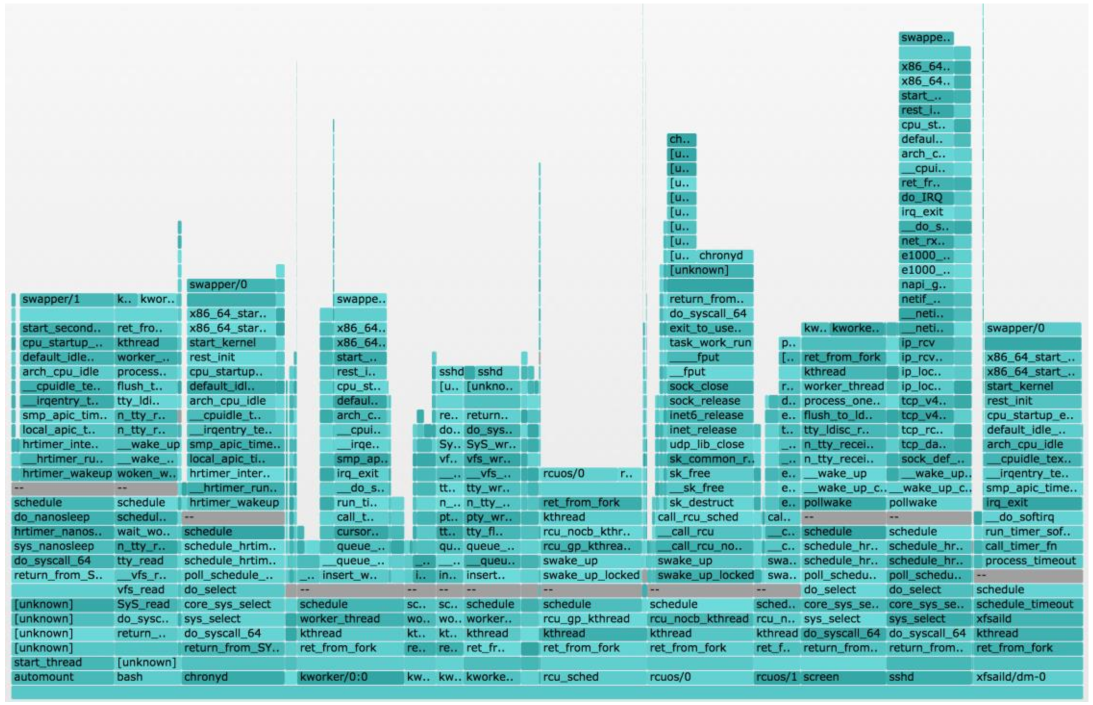
switch in
switch in
pid 121
pid 121
pid 408
pid 121
CPU 0
switch in (created)
switch out stack: mutex_lock request_processor
switch in
switch out stack: matrix_mult request_processor
switch out stack: sys_read backup_daemon
CPU 1

Enriching The Data

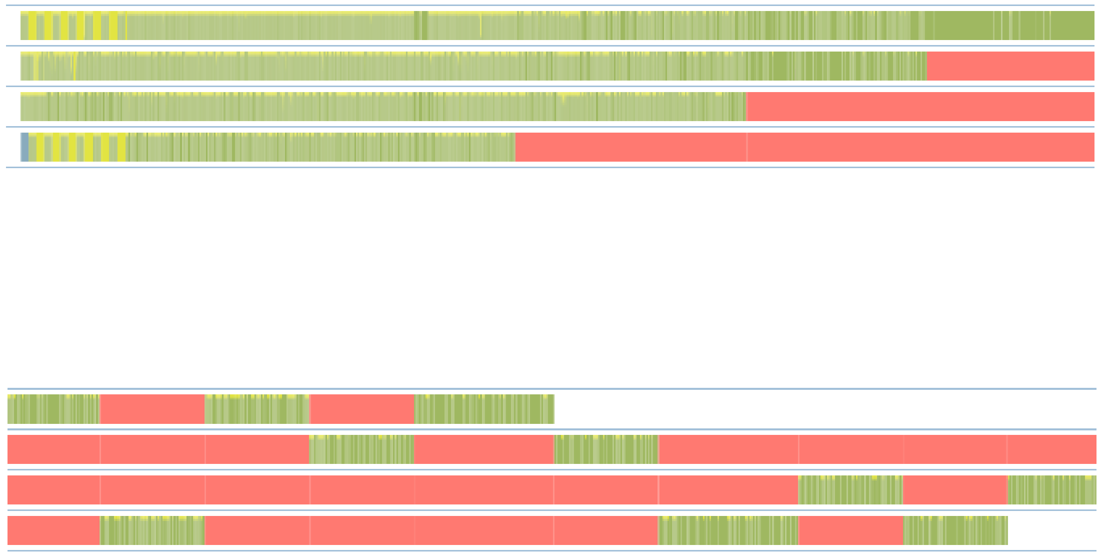
- Lock wait stacks and durations can be associated with the lock
    - Which locks are causing the most contention in this application?
    - How long does thread 123 typically have to wait for lock ABC?
- Context switch events contain the previous thread, so a wake chain can be established
    - Thread 123 was woken by thread 456, which released lock ABC
    - Application thread 456 was woken by GC thread 678, which had suspended it to perform a garbage collection

Enriched Wake Data

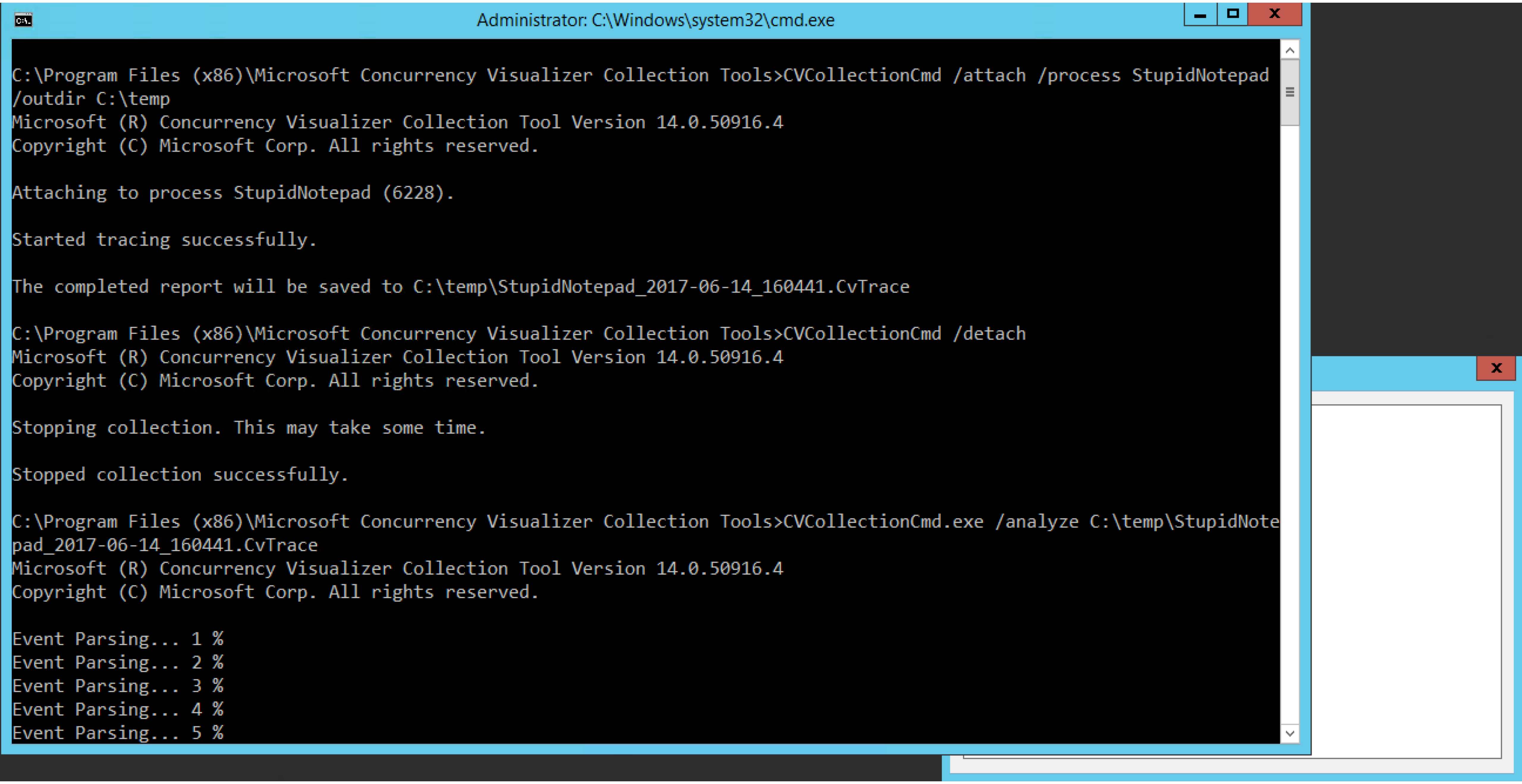
Linux
offwaketime from BCC



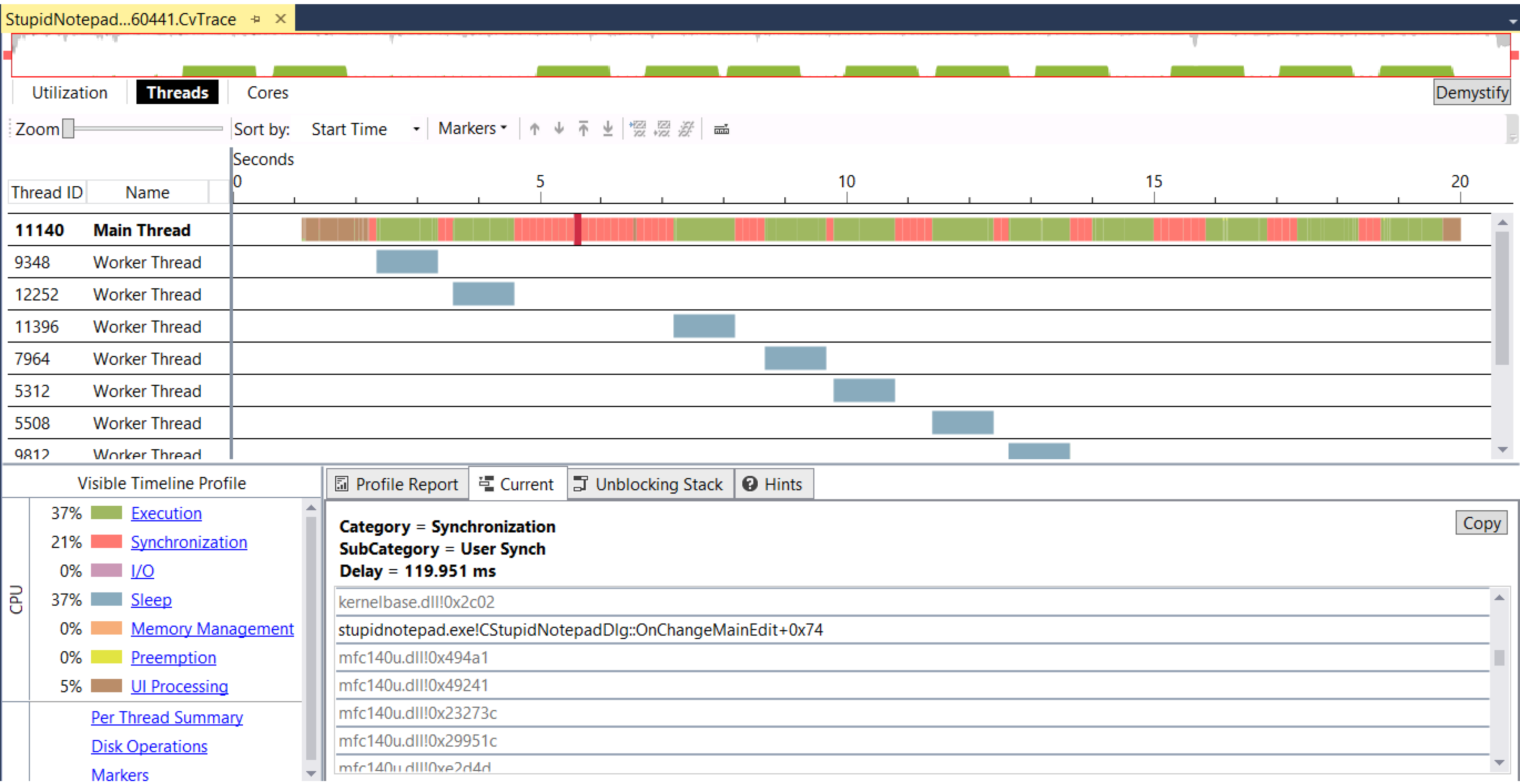
Windows
Visual Studio Concurrency Visualizer



Demo:
Blocked Thread Analysis



```
C:\Program Files (x86)\Microsoft Concurrency Visualizer Collection Tools>CVCollectionCmd /attach /process StupidNotepad
/outdir C:\temp
Microsoft (R) Concurrency Visualizer Collection Tool Version 14.0.50916.4
Copyright (C) Microsoft Corp. All rights reserved.

Attaching to process StupidNotepad (6228).

Started tracing successfully.

The completed report will be saved to C:\temp\StupidNotepad_2017-06-14_160441.CvTrace

C:\Program Files (x86)\Microsoft Concurrency Visualizer Collection Tools>CVCollectionCmd /detach
Microsoft (R) Concurrency Visualizer Collection Tool Version 14.0.50916.4
Copyright (C) Microsoft Corp. All rights reserved.

Stopping collection. This may take some time.

Stopped collection successfully.

C:\Program Files (x86)\Microsoft Concurrency Visualizer Collection Tools>CVCollectionCmd.exe /analyze C:\temp\StupidNote
pad_2017-06-14_160441.CvTrace
Microsoft (R) Concurrency Visualizer Collection Tool Version 14.0.50916.4
Copyright (C) Microsoft Corp. All rights reserved.

Event Parsing... 1 %
Event Parsing... 2 %
Event Parsing... 3 %
Event Parsing... 4 %
Event Parsing... 5 %
```



StupidNotepad...60441.CvTrace

Utilization | Threads | Cores | Demystify

Zoom | Sort by: Start Time | Markers

Seconds

Thread ID	Name
11140	**Main Thread**
9348	Worker Thread
12252	Worker Thread
11396	Worker Thread
7964	Worker Thread
5312	Worker Thread
5508	Worker Thread
9812	Worker Thread

Visible Timeline Profile

CPU

37%	🟩	Execution
21%	🟥	Synchronization
0%	🟪	I/O
37%	🟦	Sleep
0%	🟧	Memory Management
0%	🟨	Preemption
5%	🟫	UI Processing

Per Thread Summary

Disk Operations

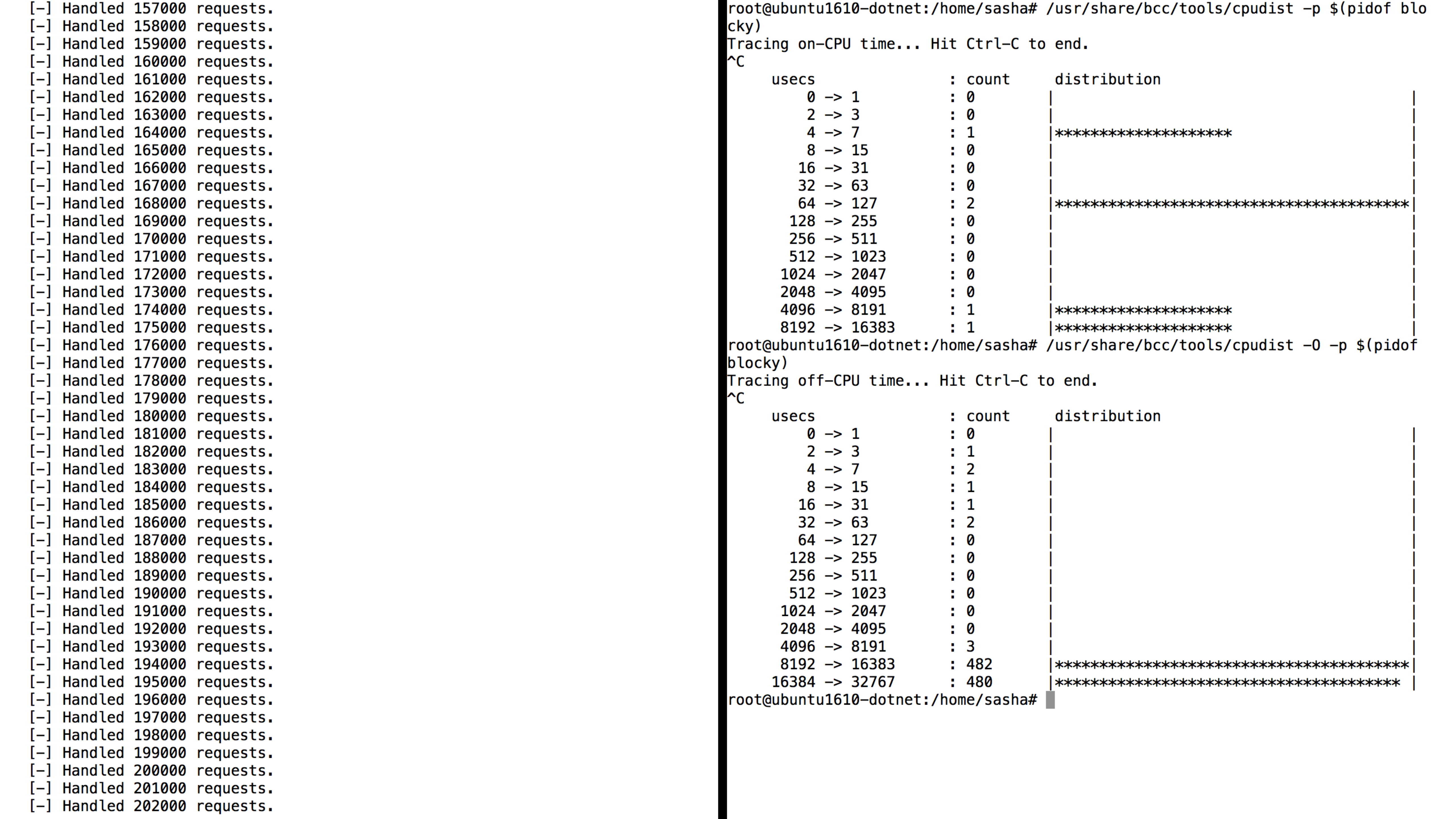
Markers

Profile Report | Current | Unblocking Stack | Hints

Copy

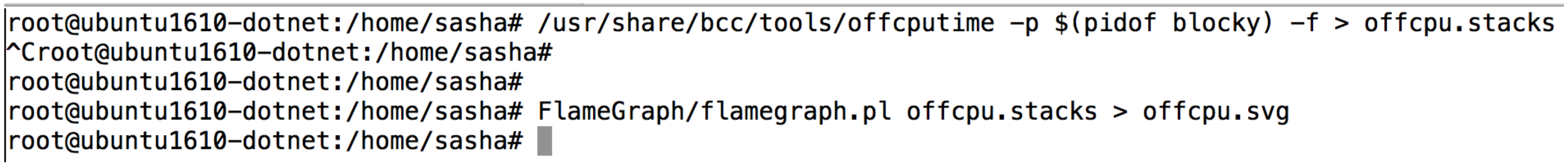
Category = Synchronization
SubCategory = User Synch
Delay = 119.951 ms

```
kernelbase.dll!0x2c02
stupidnotepad.exe!CStupidNotepadDlg::OnChangeMainEdit+0x74
mfc140u.dll!0x494a1
mfc140u.dll!0x49241
mfc140u.dll!0x23273c
mfc140u.dll!0x29951c
mfc140u.dll!0xe2d4d
```

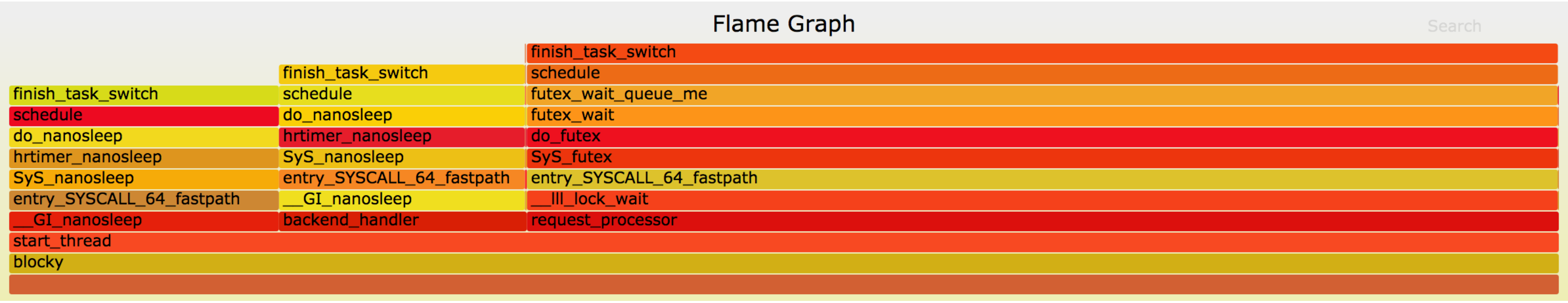


```
[-] Handled 157000 requests.
[-] Handled 158000 requests.
[-] Handled 159000 requests.
[-] Handled 160000 requests.
[-] Handled 161000 requests.
[-] Handled 162000 requests.
[-] Handled 163000 requests.
[-] Handled 164000 requests.
[-] Handled 165000 requests.
[-] Handled 166000 requests.
[-] Handled 167000 requests.
[-] Handled 168000 requests.
[-] Handled 169000 requests.
[-] Handled 170000 requests.
[-] Handled 171000 requests.
[-] Handled 172000 requests.
[-] Handled 173000 requests.
[-] Handled 174000 requests.
[-] Handled 175000 requests.
[-] Handled 176000 requests.
[-] Handled 177000 requests.
[-] Handled 178000 requests.
[-] Handled 179000 requests.
[-] Handled 180000 requests.
[-] Handled 181000 requests.
[-] Handled 182000 requests.
[-] Handled 183000 requests.
[-] Handled 184000 requests.
[-] Handled 185000 requests.
[-] Handled 186000 requests.
[-] Handled 187000 requests.
[-] Handled 188000 requests.
[-] Handled 189000 requests.
[-] Handled 190000 requests.
[-] Handled 191000 requests.
[-] Handled 192000 requests.
[-] Handled 193000 requests.
[-] Handled 194000 requests.
[-] Handled 195000 requests.
[-] Handled 196000 requests.
[-] Handled 197000 requests.
[-] Handled 198000 requests.
[-] Handled 199000 requests.
[-] Handled 200000 requests.
[-] Handled 201000 requests.
[-] Handled 202000 requests.
```

```
root@ubuntu1610-dotnet:/home/sasha# /usr/share/bcc/tools/cpudist -p $(pidof blo
cky)
Tracing on-CPU time... Hit Ctrl-C to end.
^C
     usecs               : count     distribution
         0 -> 1          : 0        |                                        |
         2 -> 3          : 0        |                                        |
         4 -> 7          : 1        |********************                    |
         8 -> 15         : 0        |                                        |
        16 -> 31         : 0        |                                        |
        32 -> 63         : 0        |                                        |
        64 -> 127        : 2        |****************************************|
       128 -> 255        : 0        |                                        |
       256 -> 511        : 0        |                                        |
       512 -> 1023       : 0        |                                        |
      1024 -> 2047       : 0        |                                        |
      2048 -> 4095       : 0        |                                        |
      4096 -> 8191       : 1        |********************                    |
      8192 -> 16383      : 1        |********************                    |
root@ubuntu1610-dotnet:/home/sasha# /usr/share/bcc/tools/cpudist -O -p $(pidof
blocky)
Tracing off-CPU time... Hit Ctrl-C to end.
^C
     usecs               : count     distribution
         0 -> 1          : 0        |                                        |
         2 -> 3          : 1        |                                        |
         4 -> 7          : 2        |                                        |
         8 -> 15         : 1        |                                        |
        16 -> 31         : 1        |                                        |
        32 -> 63         : 2        |                                        |
        64 -> 127        : 0        |                                        |
       128 -> 255        : 0        |                                        |
       256 -> 511        : 0        |                                        |
       512 -> 1023       : 0        |                                        |
      1024 -> 2047       : 0        |                                        |
      2048 -> 4095       : 0        |                                        |
      4096 -> 8191       : 3        |                                        |
      8192 -> 16383      : 482      |****************************************|
     16384 -> 32767      : 480      |*************************************** |
root@ubuntu1610-dotnet:/home/sasha# 
```



```
root@ubuntu1610-dotnet:/home/sasha# /usr/share/bcc/tools/offcputime -p $(pidof blocky) -f > offcpu.stacks
^Croot@ubuntu1610-dotnet:/home/sasha#
root@ubuntu1610-dotnet:/home/sasha#
root@ubuntu1610-dotnet:/home/sasha# FlameGraph/flamegraph.pl offcpu.stacks > offcpu.svg
root@ubuntu1610-dotnet:/home/sasha#
```



File, Disk, And Network I/O

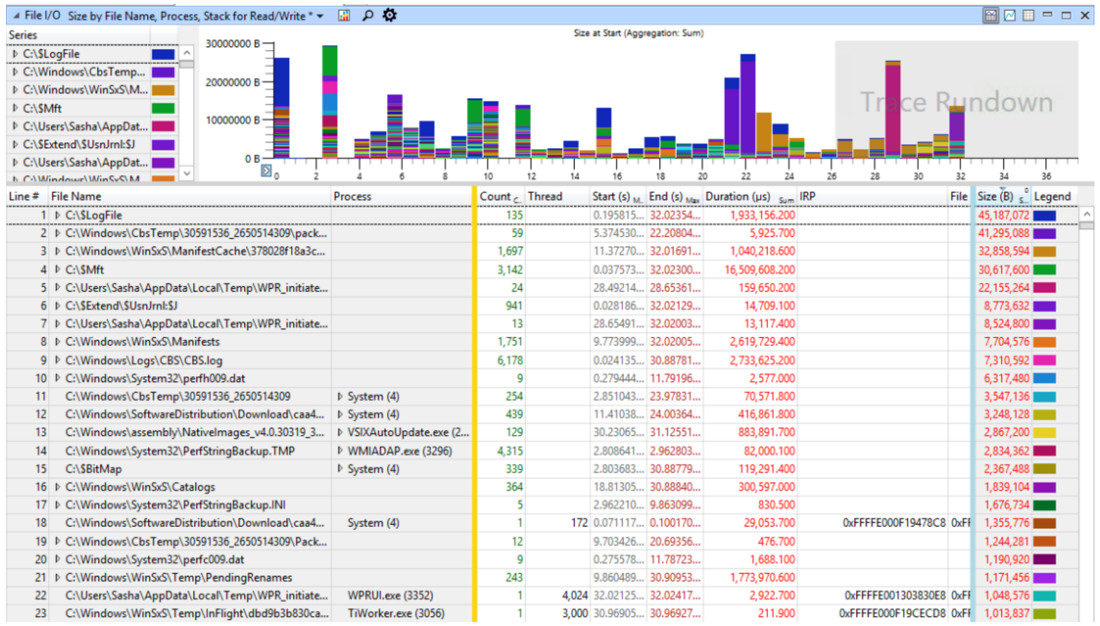
- Dedicated kernel events exist to trace various types of I/O
  - Windows ETW flags **DiskIO**, **FileIO**, **NetworkTrace**
  - Linux kernel tracepoints **block:***, **xfs/ext4/…:***, kprobes on **tcp_***, **vfs_***
- Reports may include:
  - Histogram of I/O operation latencies
  - Summary of files accessed, including size and number of reads/writes
  - Summary of active TCP connections, including size and number of recv/send
  - List of file accesses larger than or slower than a particular threshold

File I/O Summary

Linux
filetop from BCC

```
Tracing... Output every 5 secs. Hit Ctrl-C to end
08:24:24 loadavg: 0.14 0.04 0.01 2/142 3680

TID    COMM             READS  WRITES R_Kb    W_Kb    T FILE
3673   cksum            1083   0      69288   0       R libbcc.so.0.2.0
3676   cksum            1083   0      69288   0       R libbcc.so.0.2.0
3679   cksum            948    0      60656   0       R libbcc.so.0.2.0
3676   cksum            371    0      23732   0       R RecordMySQLQuery
3679   cksum            371    0      23732   0       R RecordMySQLQuery
3673   cksum            371    0      23732   0       R RecordMySQLQuery
3679   cksum            370    0      23624   0       R RandomRead
3676   cksum            370    0      23624   0       R CPUDistribution
3679   cksum            370    0      23624   0       R CPUDistribution
3679   cksum            370    0      23624   0       R HelloWorld
3673   cksum            370    0      23624   0       R HelloWorld
3676   cksum            370    0      23624   0       R HelloWorld
3673   cksum            370    0      23624   0       R RandomRead
3676   cksum            370    0      23624   0       R RandomRead
3676   cksum            370    0      23620   0       R FollyRequestContext
3673   cksum            370    0      23620   0       R FollyRequestContext
3679   cksum            370    0      23620   0       R TCPSendStack
```

Windows
WPA file I/O summary table



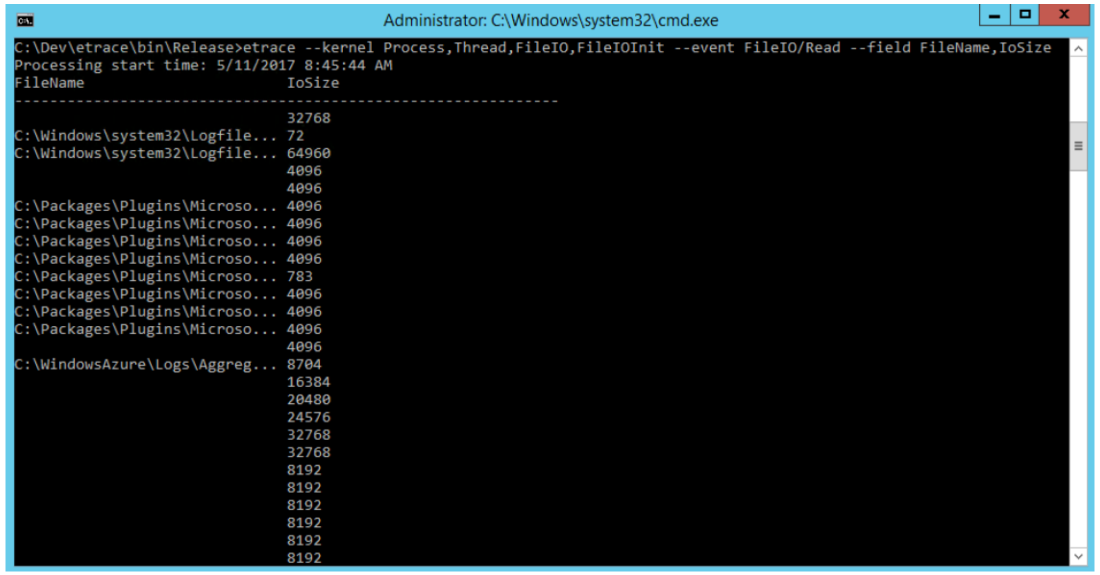
Line #	File Name	Process	Count	Thread	Start (s)	End (s)	Duration (µs)	IRP	File	Size (B)	Legend
1	C:\$LogFile		135		0.195815...	32.02354...	1,933,156.200			45,187,072	🟦
2	C:\Windows\CbsTemp\30591536_2650514309\pack...		59		5.374530...	22.20804...	5,925.700			41,295,088	🟪
3	C:\Windows\WinSxS\ManifestCache\378028f18a3c...		1,697		11.37270...	32.01691...	1,040,218.600			32,858,594	🟧
4	C:\$Mft		3,142		0.037573...	32.02300...	16,509,608.200			30,617,600	🟩
5	C:\Users\Sasha\AppData\Local\Temp\WPR_initiate...		24		28.49214...	28.65361...	159,650.200			22,155,264	🟥
6	C:\$Extend\$UsnJrnl:$J		941		0.028186...	32.02129...	14,709.100			8,773,632	🟪
7	C:\Users\Sasha\AppData\Local\Temp\WPR_initiate...		13		28.65491...	32.02003...	13,117.400			8,524,800	🟪
8	C:\Windows\WinSxS\Manifests		1,751		9.773999...	32.02005...	2,619,729.400			7,704,576	🟧
9	C:\Windows\Logs\CBS\CBS.log		6,178		0.024135...	30.88781...	2,733,625.200			7,310,592	🟪
10	C:\Windows\System32\perfh009.dat		9		0.279444...	11.79196...	2,577.000			6,317,480	🟦
11	C:\Windows\CbsTemp\30591536_2650514309	System (4)	254		2.851043...	23.97831...	70,571.800			3,547,136	🟦
12	C:\Windows\SoftwareDistribution\Download\caa4...	System (4)	439		11.41038...	24.00364...	416,861.800			3,248,128	🟨
13	C:\Windows\assembly\NativeImages_v4.0.30319_3...	VSIXAutoUpdate.exe (2...	129		30.23065...	31.12551...	883,891.700			2,867,200	🟨
14	C:\Windows\System32\PerfStringBackup.TMP	WMIADAP.exe (3296)	4,315		2.808641...	2.962803...	82,000.100			2,834,362	🟥
15	C:\$BitMap	System (4)	339		2.803683...	30.88779...	119,291.400			2,367,488	🟨
16	C:\Windows\WinSxS\Catalogs		364		18.81305...	30.88840...	300,597.000			1,839,104	🟪
17	C:\Windows\System32\PerfStringBackup.INI		5		2.962210...	9.863099...	830.500			1,676,734	🟩
18	C:\Windows\SoftwareDistribution\Download\caa4...	System (4)	1	172	0.071117...	0.100170...	29,053.700	0xFFFFE000F19478C8	0xFF	1,355,776	🟫
19	C:\Windows\CbsTemp\30591536_2650514309\Pack...		12		9.703426...	20.69356...	476.700			1,244,281	🟧
20	C:\Windows\System32\perfc009.dat		9		0.275578...	11.78723...	1,688.100			1,190,920	🟪
21	C:\Windows\WinSxS\Temp\PendingRenames		243		9.860489...	30.90953...	1,773,970.600			1,171,456	🟪
22	C:\Users\Sasha\AppData\Local\Temp\WPR_initiate...	WPRUI.exe (3352)	1	4,024	32.02125...	32.02417...	2,922.700	0xFFFFE001303830E8	0xFF	1,048,576	🟦
23	C:\Windows\WinSxS\Temp\InFlight\dbd9b3b830ca...	TiWorker.exe (3056)	1	3,000	30.96905...	30.96927...	211.900	0xFFFFE000F19CECD8	0xFF	1,013,837	🟩

Tracing File Accesses in Real-Time

Linux
fileslower from BCC

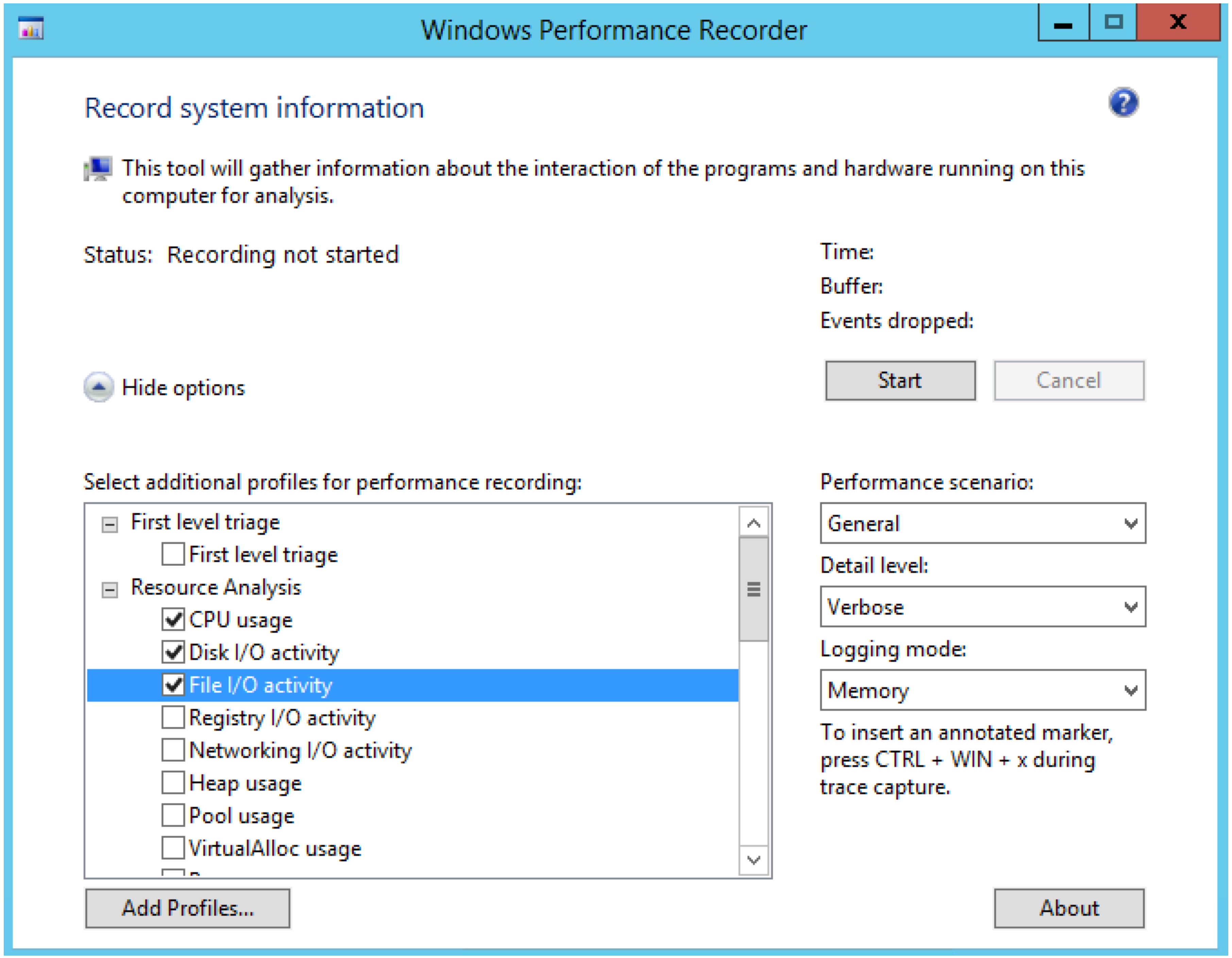
```
Tracing sync read/writes slower than 1 ms
TIME(s)  COMM           TID    D BYTES    LAT(ms) FILENAME
0.560    cksum          3699   R 65536       1.18 RecordMySQLQuery
15.990   bash           3700   R 128         3.15 dd
16.077   dd             3700   W 1048576     1.15 file.out
16.260   dd             3700   W 1048576     1.99 file.out
16.276   dd             3700   W 1048576     1.94 file.out
16.295   dd             3700   W 1048576     2.20 file.out
16.315   dd             3700   W 1048576     2.25 file.out
16.337   dd             3700   W 1048576     2.75 file.out
16.356   dd             3700   W 1048576     2.51 file.out
16.382   dd             3700   W 1048576     3.83 file.out
16.392   dd             3700   W 1048576     1.22 file.out
16.408   dd             3700   W 1048576     2.53 file.out
```

Windows
etrace



```
C:\Dev\etrace\bin\Release>etrace --kernel Process,Thread,FileIO,FileIOInit --event FileIO/Read --field FileName,IoSize
Processing start time: 5/11/2017 8:45:44 AM
FileName                         IoSize
-------------------------------------------------------------
                                 32768
C:\Windows\system32\Logfile...   72
C:\Windows\system32\Logfile...   64960
                                 4096
                                 4096
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   783
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   4096
C:\Packages\Plugins\Microso...   4096
                                 4096
C:\WindowsAzure\Logs\Aggreg...   8704
                                 16384
                                 20480
                                 24576
                                 32768
                                 32768
                                 8192
                                 8192
                                 8192
                                 8192
                                 8192
                                 8192
```

Demo:
Summarizing I/O Operations


Windows Performance Recorder

Record system information

This tool will gather information about the interaction of the programs and hardware running on this computer for analysis.

Status: Recording not started

Time:
Buffer:
Events dropped:

Select additional profiles for performance recording:

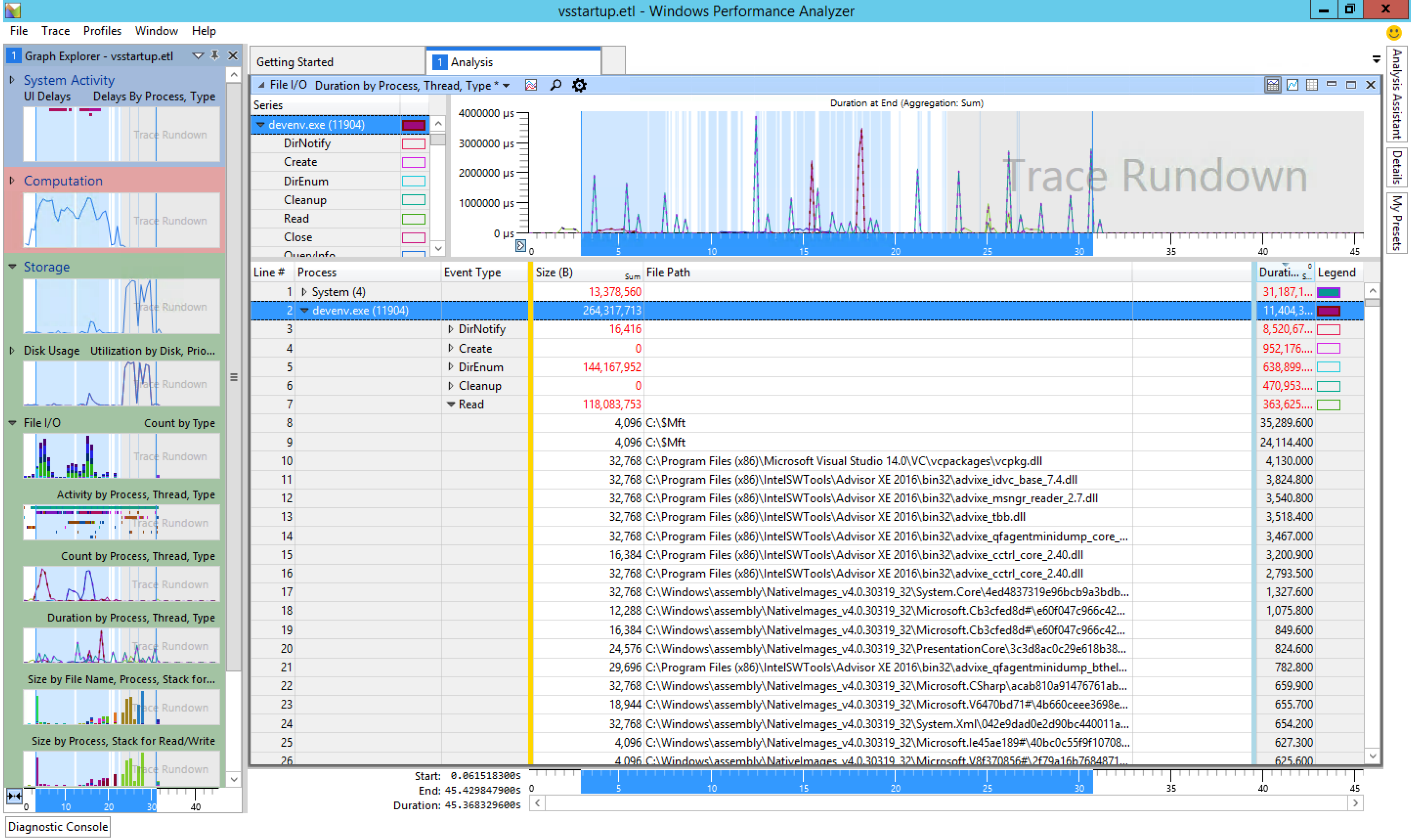
- First level triage
  - [ ] First level triage
- Resource Analysis
  - [✓] CPU usage
  - [✓] Disk I/O activity
  - [✓] File I/O activity
  - [ ] Registry I/O activity
  - [ ] Networking I/O activity
  - [ ] Heap usage
  - [ ] Pool usage
  - [ ] VirtualAlloc usage

Performance scenario: General

Detail level: Verbose

Logging mode: Memory

To insert an annotated marker, press CTRL + WIN + x during trace capture.



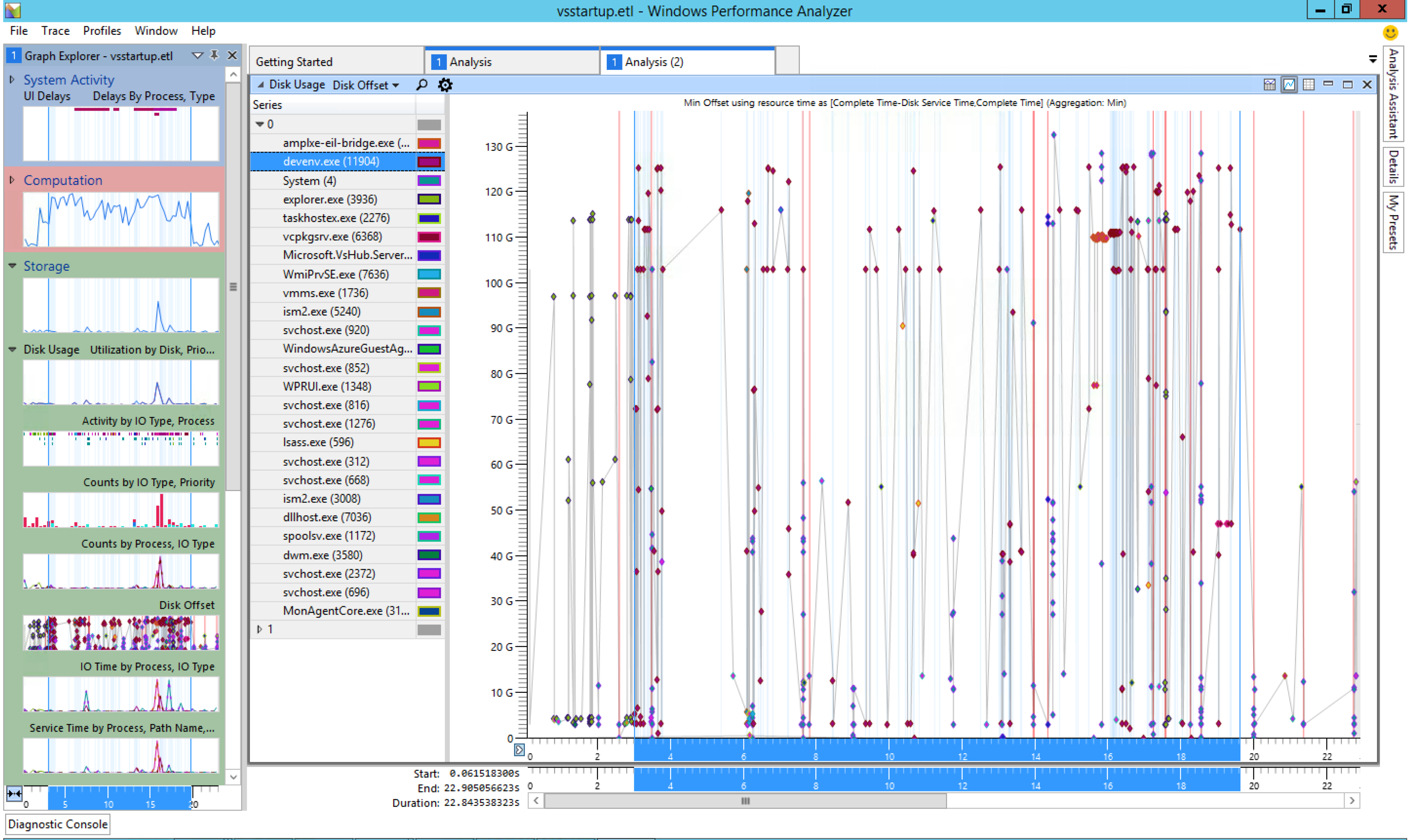
vsstartup.etl - Windows Performance Analyzer

File Trace Profiles Window Help

1 Graph Explorer - vsstartup.etl
System Activity
UI Delays Delays By Process, Type
Computation
Storage
Disk Usage Utilization by Disk, Prio...
File I/O Count by Type
Activity by Process, Thread, Type
Count by Process, Thread, Type
Duration by Process, Thread, Type
Size by File Name, Process, Stack for...
Size by Process, Stack for Read/Write
Trace Rundown
Diagnostic Console
Getting Started
1 Analysis
File I/O Duration by Process, Thread, Type *
Series
devenv.exe (11904)
DirNotify
Create
DirEnum
Cleanup
Read
Close
QueryInfo
Duration at End (Aggregation: Sum)
4000000 µs
3000000 µs
2000000 µs
1000000 µs
0 µs
0 5 10 15 20 25 30 35 40 45
Trace Rundown
Analysis Assistant
Details
My Presets

Line #	Process	Event Type	Size (B) Sum	File Path	Durati...	Legend
1	System (4)		13,378,560		31,187,1...	🟩
2	devenv.exe (11904)		264,317,713		11,404,3...	🟥
3		DirNotify	16,416		8,520,67...	□
4		Create	0		952,176....	□
5		DirEnum	144,167,952		638,899....	□
6		Cleanup	0		470,953....	□
7		Read	118,083,753		363,625....	□
8			4,096	C:\$Mft	35,289.600	
9			4,096	C:\$Mft	24,114.400	
10			32,768	C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\vcpackages\vcpkg.dll	4,130.000	
11			32,768	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_idvc_base_7.4.dll	3,824.800	
12			32,768	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_msngr_reader_2.7.dll	3,540.800	
13			32,768	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_tbb.dll	3,518.400	
14			32,768	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_qfagentminidump_core_...	3,467.000	
15			16,384	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_cctrl_core_2.40.dll	3,200.900	
16			32,768	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_cctrl_core_2.40.dll	2,793.500	
17			32,768	C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Core\4ed4837319e96bcb9a3bdb...	1,327.600	
18			12,288	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.Cb3cfed8d#\e60f047c966c42...	1,075.800	
19			16,384	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.Cb3cfed8d#\e60f047c966c42...	849.600	
20			24,576	C:\Windows\assembly\NativeImages_v4.0.30319_32\PresentationCore\3c3d8ac0c29e618b38...	824.600	
21			29,696	C:\Program Files (x86)\IntelSWTools\Advisor XE 2016\bin32\advixe_qfagentminidump_bthel...	782.800	
22			32,768	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.CSharp\acab810a91476761ab...	659.900	
23			18,944	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.V6470bd71#\4b660ceee3698e...	655.700	
24			32,768	C:\Windows\assembly\NativeImages_v4.0.30319_32\System.Xml\042e9dad0e2d90bc440011a...	654.200	
25			4,096	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.Ie45ae189#\40bc0c55f9f10708...	627.300	
26			4,096	C:\Windows\assembly\NativeImages_v4.0.30319_32\Microsoft.V8f370856#\2f79a16b7684871...	625.600	

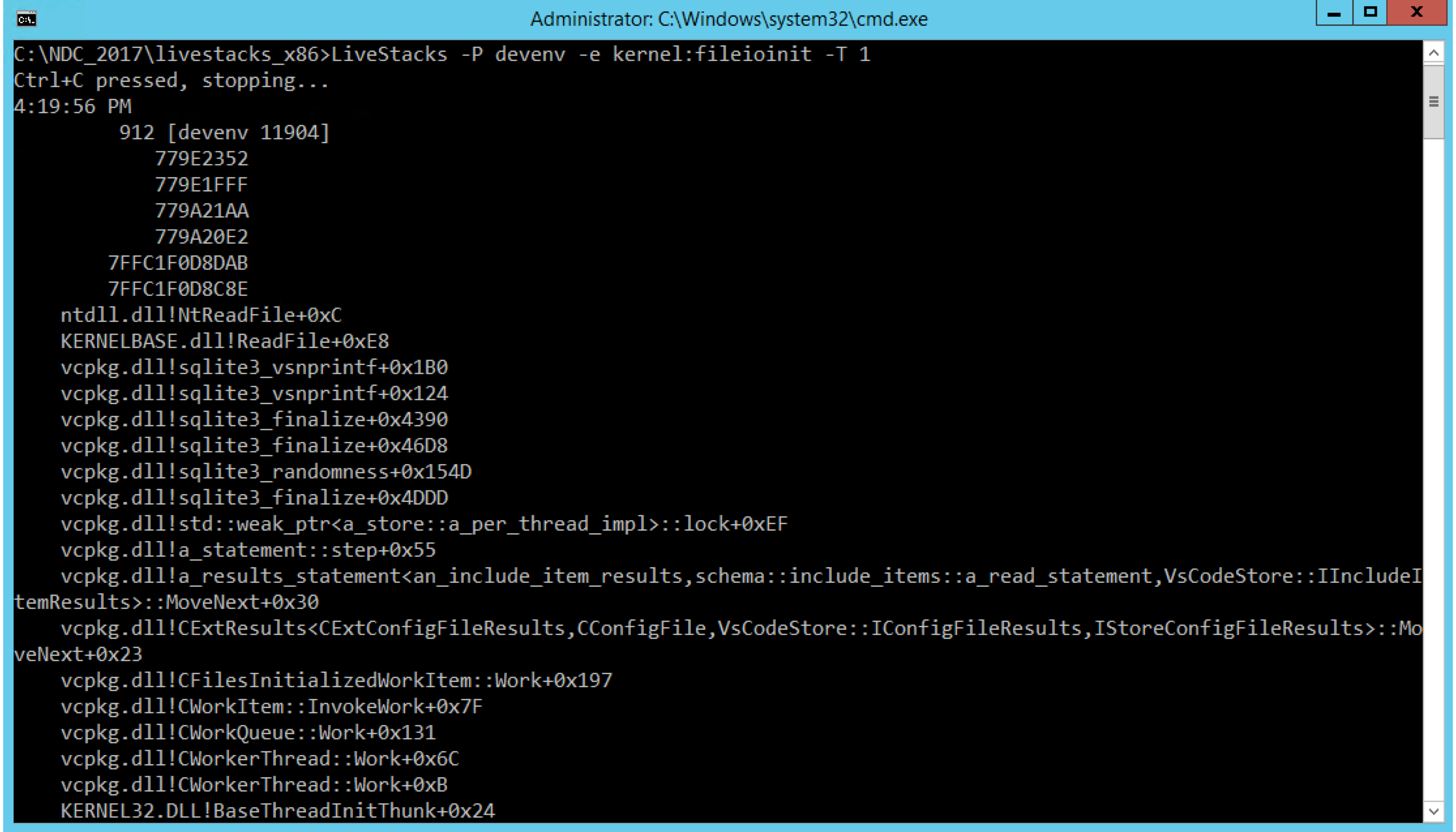
```
Start:    0.061518300s
End:     45.429847900s
Duration: 45.368329600s
```


vsstartup.etl - Windows Performance Analyzer
File
Trace
Profiles
Window
Help
Graph Explorer - vsstartup.etl
System Activity
UI Delays
Delays By Process, Type
Computation
Storage
Disk Usage
Utilization by Disk, Prio...
Activity by IO Type, Process
Counts by IO Type, Priority
Counts by Process, IO Type
Disk Offset
IO Time by Process, IO Type
Service Time by Process, Path Name,...
Getting Started
Analysis
Analysis (2)
Disk Usage
Disk Offset
Analysis Assistant
Details
My Presets
Diagnostic Console

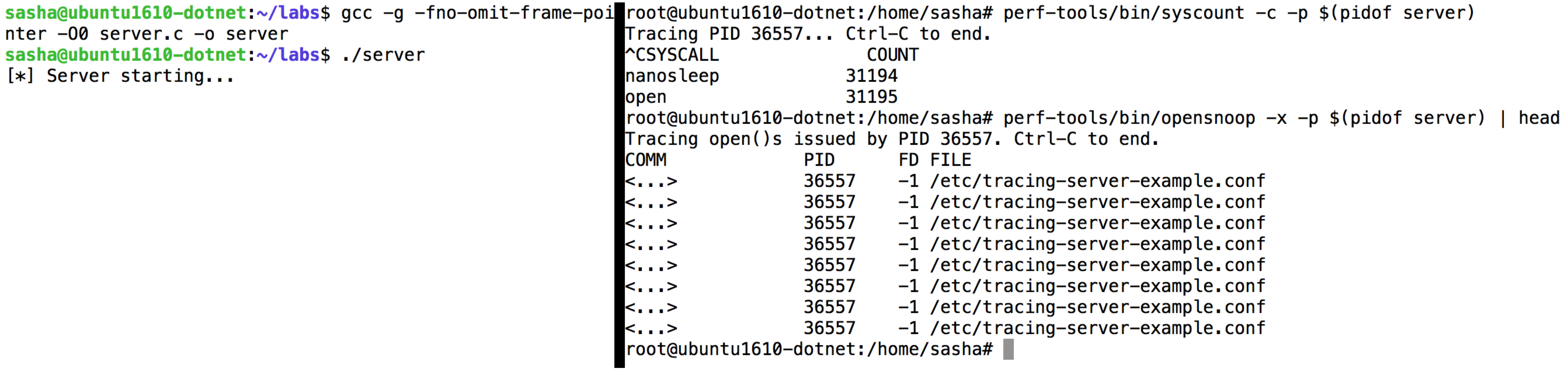
Series
- 0
  - amplxe-eil-bridge.exe (...
  - devenv.exe (11904)
  - System (4)
  - explorer.exe (3936)
  - taskhostex.exe (2276)
  - vcpkgsrv.exe (6368)
  - Microsoft.VsHub.Server...
  - WmiPrvSE.exe (7636)
  - vmms.exe (1736)
  - ism2.exe (5240)
  - svchost.exe (920)
  - WindowsAzureGuestAg...
  - svchost.exe (852)
  - WPRUI.exe (1348)
  - svchost.exe (816)
  - svchost.exe (1276)
  - lsass.exe (596)
  - svchost.exe (312)
  - svchost.exe (668)
  - ism2.exe (3008)
  - dllhost.exe (7036)
  - spoolsv.exe (1172)
  - dwm.exe (3580)
  - svchost.exe (2372)
  - svchost.exe (696)
  - MonAgentCore.exe (31...
- 1

Min Offset using resource time as [Complete Time-Disk Service Time,Complete Time] (Aggregation: Min)

Start: 0.061518300s
End: 22.905056623s
Duration: 22.843538323s



```
C:\NDC_2017\livestacks_x86>LiveStacks -P devenv -e kernel:fileioinit -T 1
Ctrl+C pressed, stopping...
4:19:56 PM
         912 [devenv 11904]
            779E2352
            779E1FFF
            779A21AA
            779A20E2
        7FFC1F0D8DAB
        7FFC1F0D8C8E
    ntdll.dll!NtReadFile+0xC
    KERNELBASE.dll!ReadFile+0xE8
    vcpkg.dll!sqlite3_vsnprintf+0x1B0
    vcpkg.dll!sqlite3_vsnprintf+0x124
    vcpkg.dll!sqlite3_finalize+0x4390
    vcpkg.dll!sqlite3_finalize+0x46D8
    vcpkg.dll!sqlite3_randomness+0x154D
    vcpkg.dll!sqlite3_finalize+0x4DDD
    vcpkg.dll!std::weak_ptr<a_store::a_per_thread_impl>::lock+0xEF
    vcpkg.dll!a_statement::step+0x55
    vcpkg.dll!a_results_statement<an_include_item_results,schema::include_items::a_read_statement,VsCodeStore::IIncludeI
temResults>::MoveNext+0x30
    vcpkg.dll!CExtResults<CExtConfigFileResults,CConfigFile,VsCodeStore::IConfigFileResults,IStoreConfigFileResults>::Mo
veNext+0x23
    vcpkg.dll!CFilesInitializedWorkItem::Work+0x197
    vcpkg.dll!CWorkItem::InvokeWork+0x7F
    vcpkg.dll!CWorkQueue::Work+0x131
    vcpkg.dll!CWorkerThread::Work+0x6C
    vcpkg.dll!CWorkerThread::Work+0xB
    KERNEL32.DLL!BaseThreadInitThunk+0x24
```



```
sasha@ubuntu1610-dotnet:~/labs$ gcc -g -fno-omit-frame-poi
nter -O0 server.c -o server
sasha@ubuntu1610-dotnet:~/labs$ ./server
[*] Server starting...
```

```
root@ubuntu1610-dotnet:/home/sasha# perf-tools/bin/syscount -c -p $(pidof server)
Tracing PID 36557... Ctrl-C to end.
^CSYSCALL              COUNT
nanosleep            31194
open                 31195
root@ubuntu1610-dotnet:/home/sasha# perf-tools/bin/opensnoop -x -p $(pidof server) | head
Tracing open()s issued by PID 36557. Ctrl-C to end.
COMM             PID      FD FILE
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
<...>            36557    -1 /etc/tracing-server-example.conf
root@ubuntu1610-dotnet:/home/sasha# 
```

Summary

- We have learned:

✓To obtain and analyze dumps of C++ apps

✓Which production-ready tracing tools can be used with C++ apps

✓To obtain CPU profiles and flame graphs

✓To identify memory leaking call stacks

References

- `perf` and flame graphs
  - https://perf.wiki.kernel.org/index.php/Main_Page
  - http://www.brendangregg.com/perf.html
  - https://github.com/brendangregg/perf-tools
- Event Tracing for Windows
  - https://msdn.microsoft.com/en-us/windows/hardware/commercialize/test/wpt/index
  - https://github.com/goldshtn/etrace
  - https://github.com/goldshtn/LiveStacks
  - https://github.com/Microsoft/perfview
- Dump analysis
  - https://msdn.microsoft.com/en-us/library/windows/hardware/ff551063(v=vs.85).aspx
  - http://dumpanalysis.org/
  - http://windbg.org
- BCC tutorials
  - https://github.com/iovisor/bcc/blob/master/docs/tutorial.md
  - https://github.com/iovisor/bcc/blob/master/docs/tutorial_bcc_python_developer.md
  - https://github.com/iovisor/bcc/blob/master/docs/reference_guide.md

Thank You!

Sasha Goldshtein

Google Research

goldshtn

goldshtn