计算机安全与维护

Windows系统崩溃与恢复

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

Windows为什么会崩溃

- 页面错误
  - —0xA-IRQL_NOT_LESS_OR_EQUAL
  - —0xD1-DRIVER_IRQL_NOT_LESS_OR_EQUAL
- 电源管理
  - —0x9F-DRIVER_POWER_STATE_FAILURE
- 显示
  - —0x116 - VIDEO_TDR_FAILURE

Windows为什么会崩溃

- **异常和缺陷**
  - —0x1E - KMODE_EXCEPTION_NOT_HANDLED
  - —0x3B - SYSTEM_SERVICE_EXCEPTION
  - —0x7E-SYSTEM_THREAD_EXCEPTION_NOT_HANDLED
  - —0x7F - UNEXPECTED_KERNEL_MODE_TRAP

Windows为什么会崩溃

- 硬件
  - —0x7A - KERNEL_DATA_INPAGE_ERROR
  - —0x124 - WHEA_UNCORRECTABLE_ERROR
- 关键部件
  - —0xF4 - CRITICAL_OBJECT_TERMINATION
- NTFS文件系统
  - —0x24 - NTFS_FILE_SYSTEM

Windows为什么会崩溃

- 内核池
  - 0x19 - BAD_POOL_HEADER
  - 0xC2 - BAD_POOL_CALLER
  - 0xC5 - DRIVER_CORRUPTED_EXPOOL
- 内存管理
  - 0x1A - MEMORY_MANAGEMENT
  - 0x4E - PFN_LIST_CORRUPT
- USB
  - 0xFE - BUGCODE_USB_DRIVER

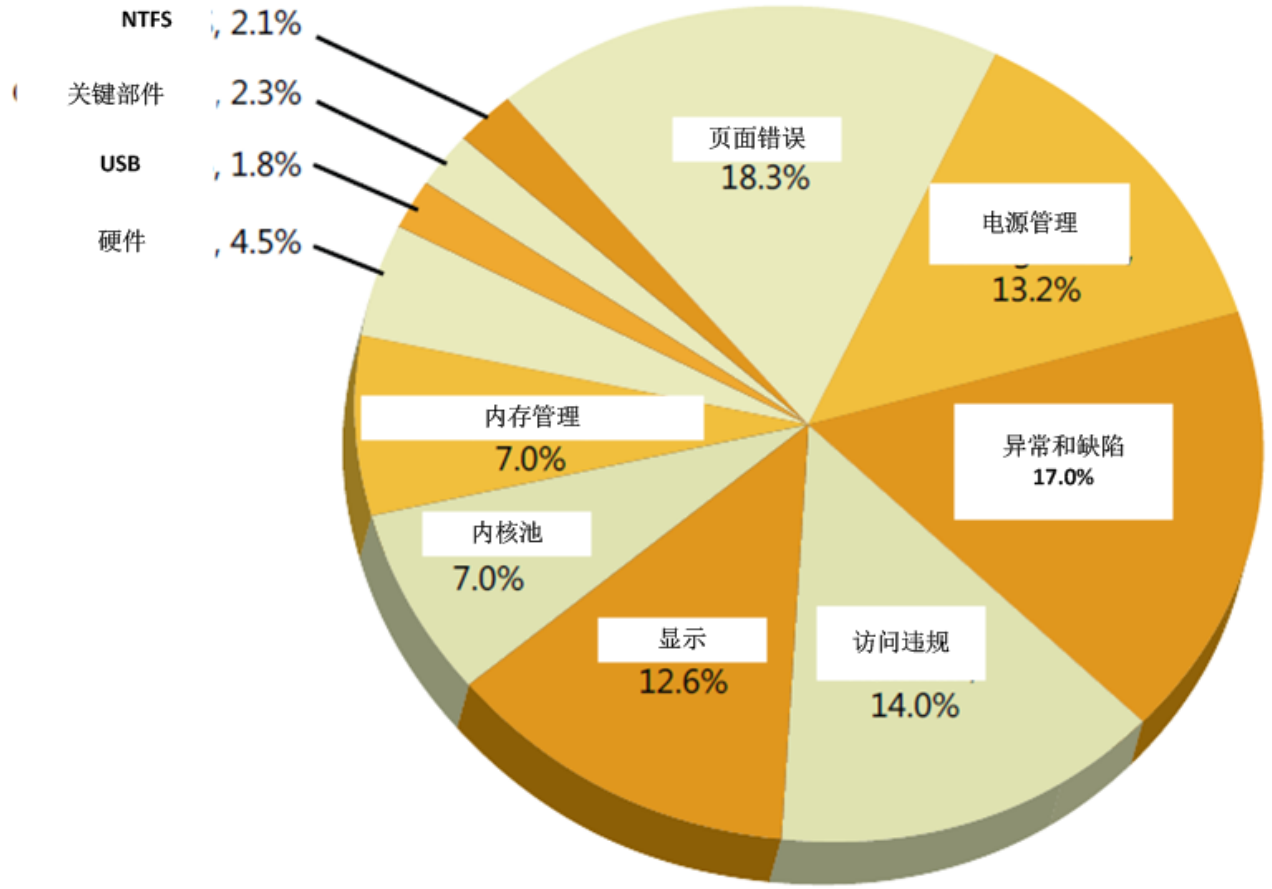
Windows为什么会崩溃

- 访问违例
  - —0x50 - PAGE_FAULT_IN_NONPAGED_AREA
  - —0x8E - KERNEL_MODE_EXCEPTION_NOT_HANDLED with P1 = 0xC0000005

STATUS_ACCESS_VIOLATION

Windows为什么会崩溃

- Windows 7引起蓝屏的主要因素



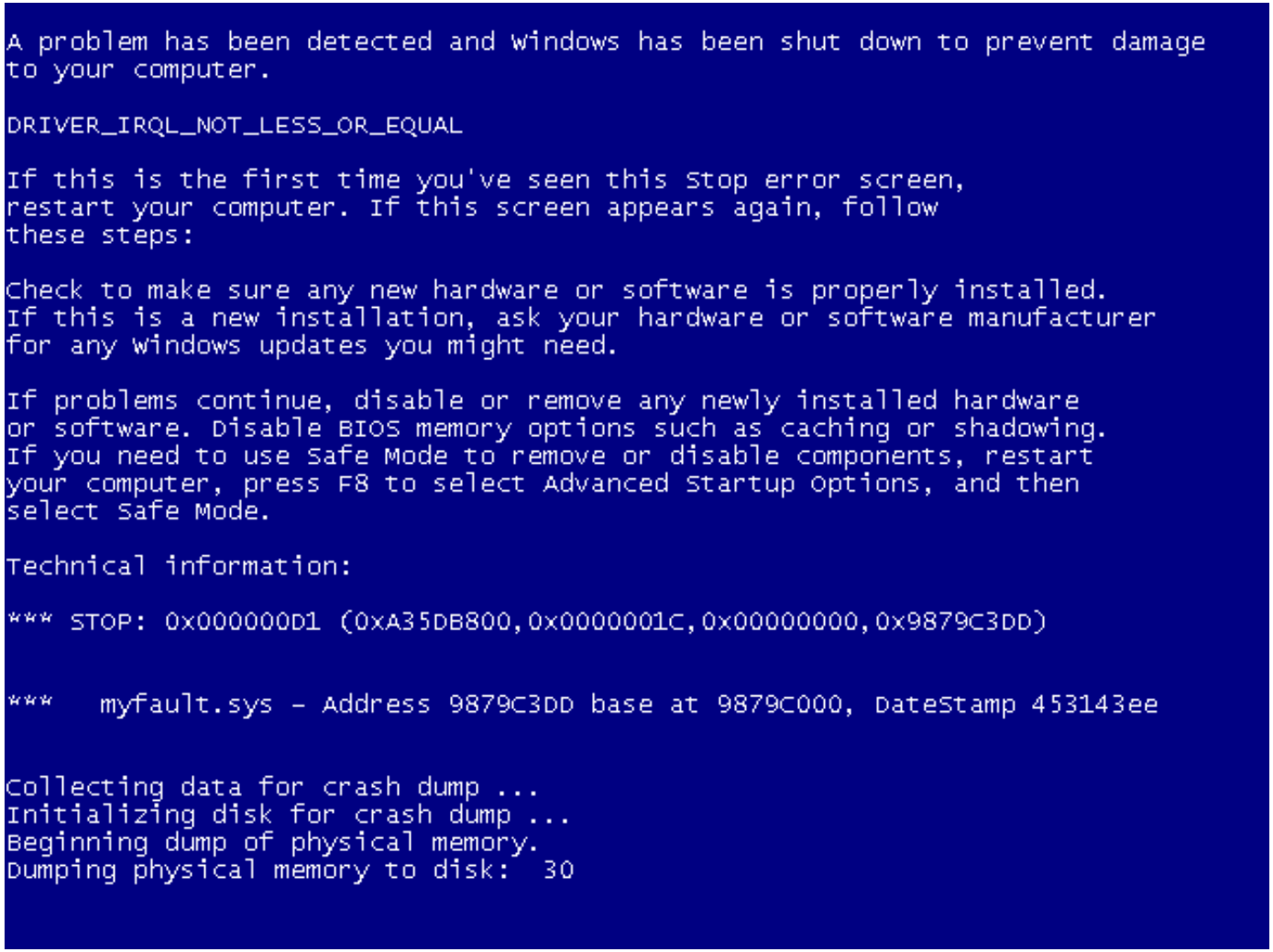
Windows为什么会崩溃

- Windows可以避免崩溃吗？
- 是否可以忽略该异常，让设备驱动程序或者子系统继续执行
- 该错误被隔离和该组件被回复的可能性是存在的
- 更有可能检测到的异常来源于更深层的问题，系统继续执行可能导致更多的异常，存储数据也可能会被破坏，招致的风险太高

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

蓝屏


A problem has been detected and Windows has been shut down to prevent damage
to your computer.
DRIVER_IRQL_NOT_LESS_OR_EQUAL
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:
Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.
If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.
Technical information:
*** STOP: 0x000000D1 (0xA35DB800,0x0000001C,0x00000000,0x9879C3DD)
*** myfault.sys - Address 9879C3DD base at 9879C000, DateStamp 453143ee
Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk: 30

蓝屏

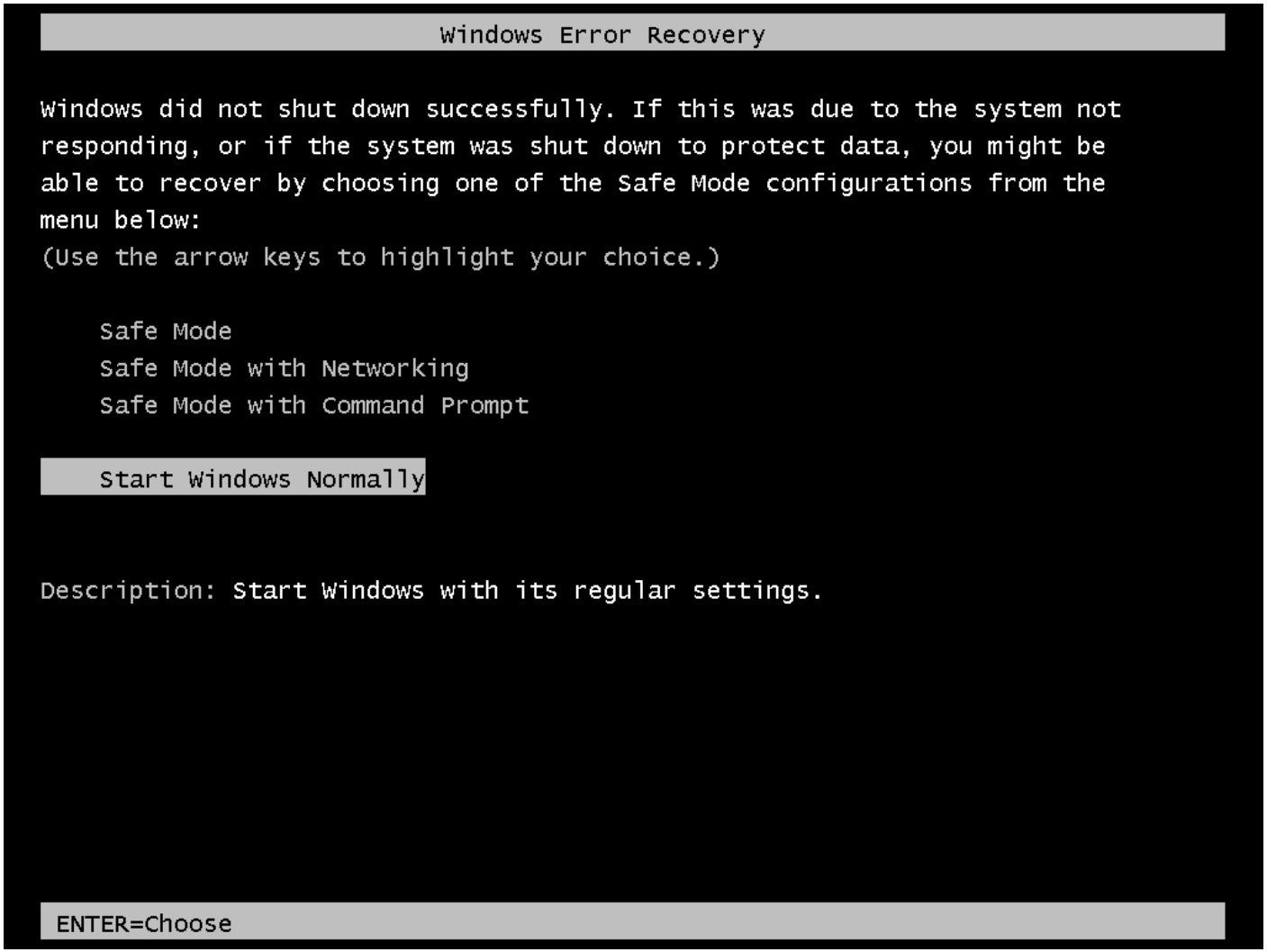
- 不管系统崩溃的原因是什么，真正执行崩溃的函数是KeBugCheckEx，该函数接收一个停止代码，以及四个根据停止代码来解释的参数
- KeBugCheckEx屏蔽了该系统所有处理器上的所有中断以后，将显示器切换到低分辨率的VGA图形模式，绘制蓝色背景
- 显示停止代码，紧跟一些文本建议用户怎么做

蓝屏

- KeBugCheckEx调用任何已注册的设备驱动程序错误检查回调函数，让驱动程序有机会停止他们的设备
- KeBugCheckEx在蓝屏显示停止代码的文本表示，第一行列出了传递给KeBugCheckEx的停止代码和四个参数
- 其中参数包含操作系统或设备驱动程序代码地址时，显示该地址所处的模块的基地址，日期戳，以及设备驱动程序的文件名

蓝屏

- Windows的错误信息恢复界面



蓝屏

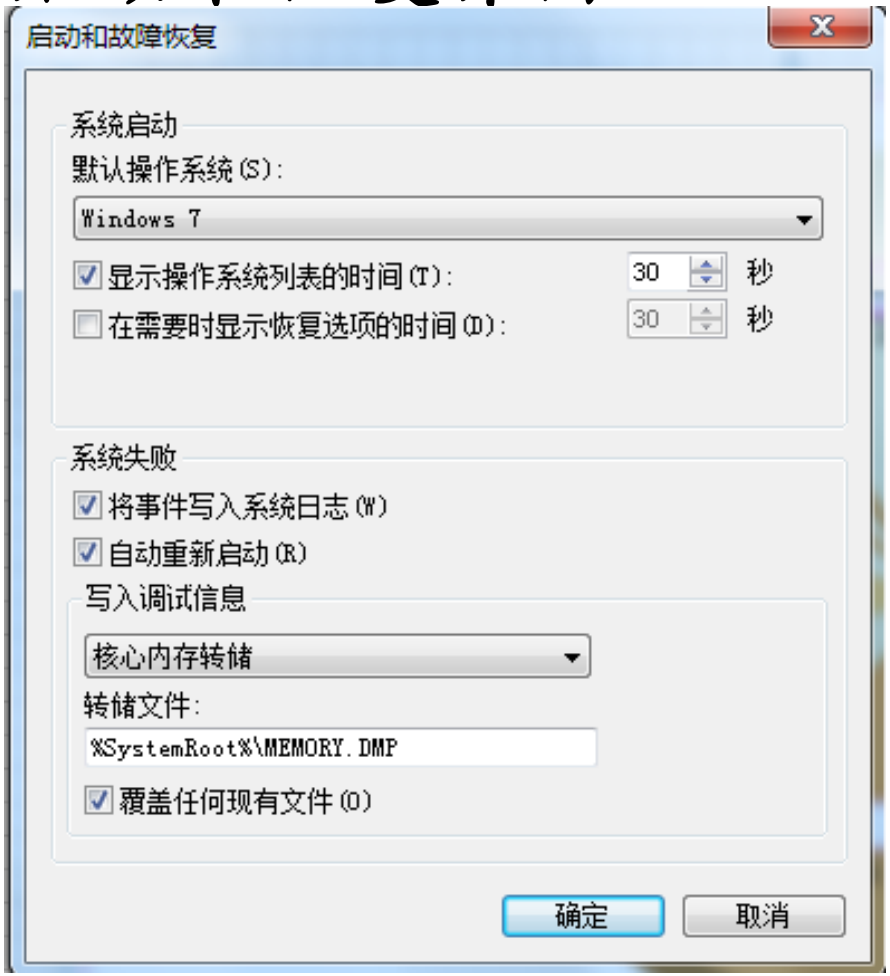
- 通常是在安装了一个新的软件产品或者硬件设备以后才开始看到蓝屏的，可以让windows将注册表的设备驱动注册键从最近一次成功引导的配置中恢复一份拷贝
- 如果仍然看到蓝屏，一种显而易见的方法是，卸掉在你第一次看到蓝屏之前刚刚加入的那些组件

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

崩溃转储文件

- Windows 7启动和恢复界面



崩溃转储文件

- 对于一次系统崩溃，有三种层次的信息可以被记录下来
  - —完全内存转储
  - —内核内存转储
  - —小内存转储

崩溃转储文件

- 完全内存转储
  - —包含了在崩溃时刻所有的物理内存
  - —这种转储类型要求页面文件至少是物理内存大小再加上1MB（用于记录头信息）
  - —由于在大内存系统上，它需要非常大的页面文件，所以，这种类型的转储文件是最少见的设置

崩溃转储文件

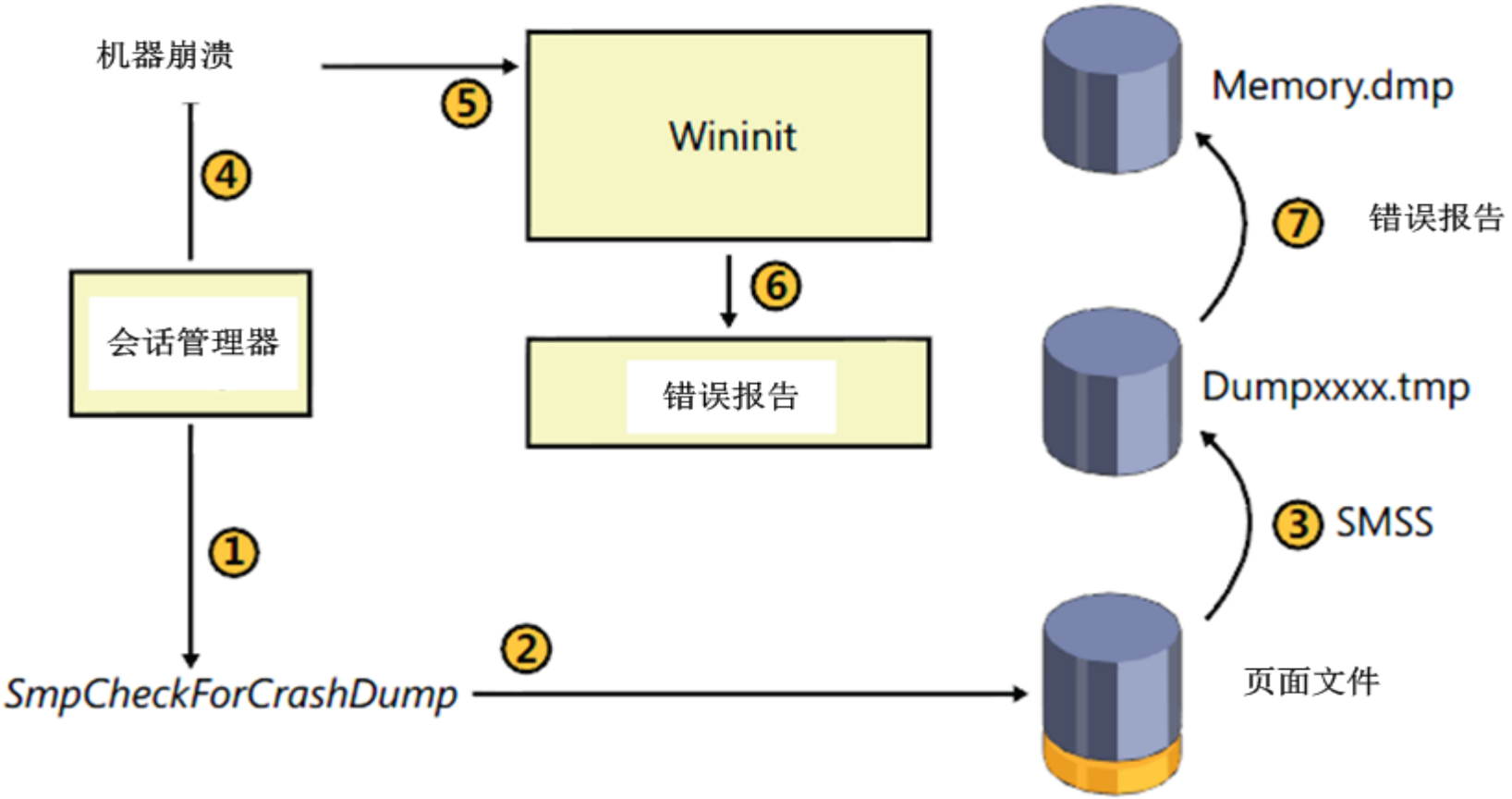
- 内核内存转储
  - —只包含了在崩溃时刻位于物理内存中的内核模式读/写页面
  - —因为只有内核模式的代码才可以直接导致windows崩溃，所以，用户进程页面通常对于崩溃调试来说是不必要的
  - —所有与崩溃转储分析有关的数据结构也被记录在内核内存的转储中
  - —没有很好的办法来预测内核内存转储的大小，取决于操作系统和活动驱动程序分配的内核模式内存数量

崩溃转储文件

- 小内存存储
  - —也被成为小转储（minidump）或分类优先转储（triage dump），大小为64KB（在64位系统上为128KB）
  - —包含停止代码和参数，已加载的设备驱动程序列表描述当前进程和线程的数据结构，以及引起此次崩溃的线程的内核栈

崩溃转储文件

- 崩溃转储的生成



崩溃转储文件

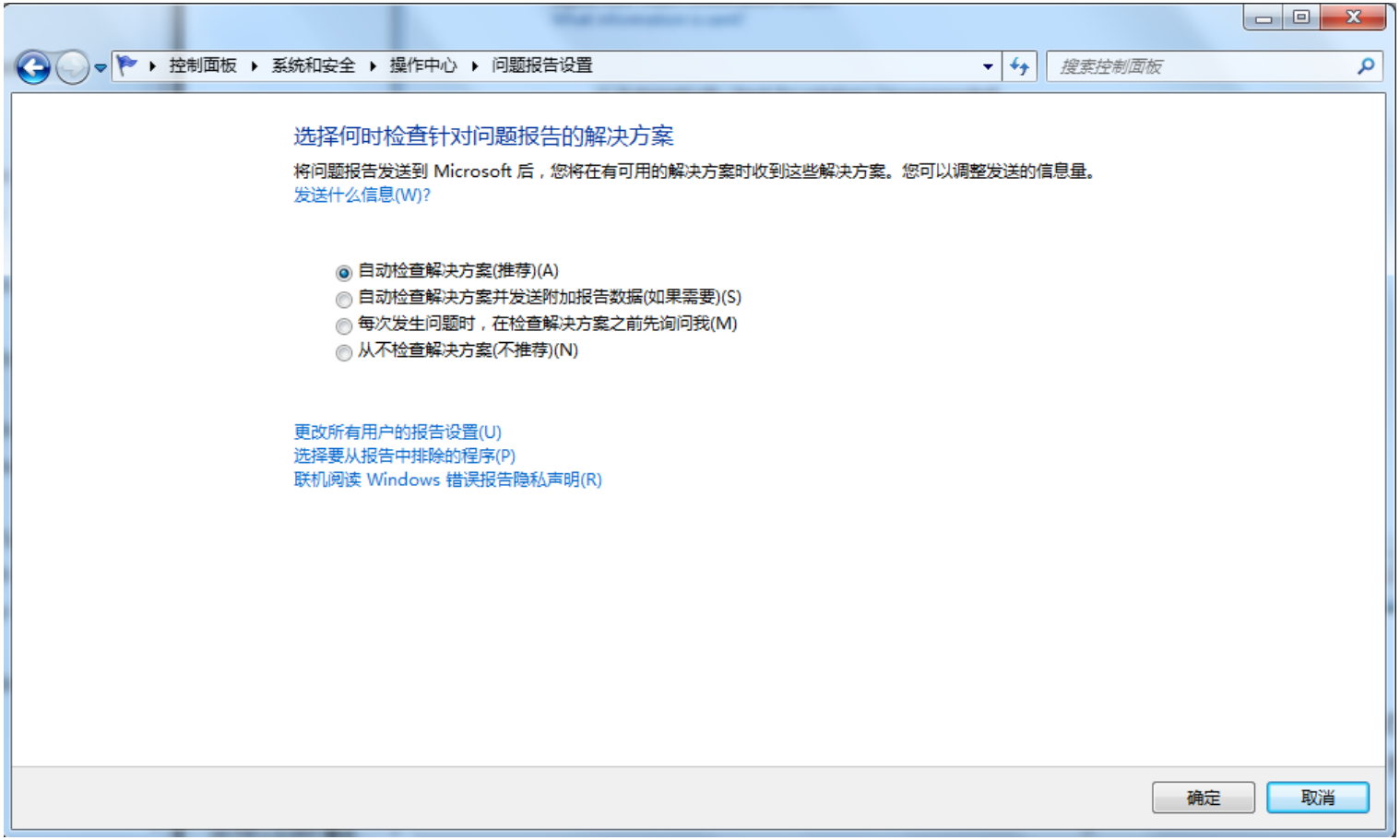
- 当系统引导时，读取注册表值HKLM\CurrentControlSet\Control\CrashControl，以检查当前配置的崩溃转储选项
- 写崩溃转储过程中涉及的组件，系统计算校验和
- 当KeBugCheckEx执行时，计算这些组件的校验和，匹配时，KeBugCheckEx直接将转储信息写到由页面文件占据的磁盘扇区中，绕过文件系统驱动程序

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

Windows错误报告

- 问题报告配置页面



Windows错误报告

- Windows错误报告可以自动将进程和系统崩溃提交给Microsoft进行分析
- 相关的配置信息保存在注册表的HKLM\Software\Microsoft\PCHealth\ErrorReporting中
- 在崩溃之后系统重新引导起来，检查ErrorReporting的值，包括Showui，DoReport和IncludeKernelFaults,如果这三个值都是true，则发送一个崩溃转储报告

Windows错误报告

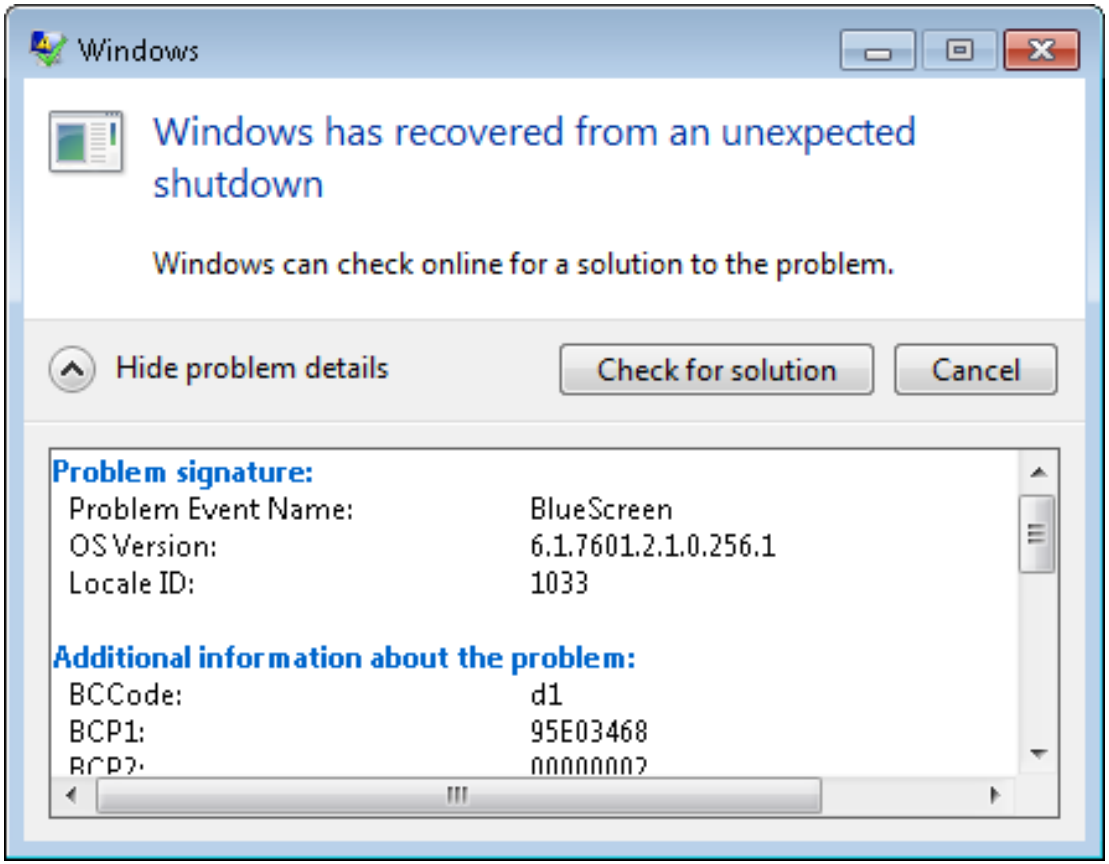
- 如果它生成的转储类型不是小转储，则提取一个小转储，并将其保存在\windows\minidumps的默认位置处
- 将小转储文件的名称写到HKLM\software\microsoft\windows\windows Error Reporting\KernelFaults\Queue中
- 增加指令到HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce执行WerFault.exe,这样当第一个用户登录时， WerFault.exe被执行

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- **在线崩溃分析**
- 基本的崩溃转储分析
- 高级的崩溃转储分析

在线崩溃分析

- **崩溃转储错误报告**



在线崩溃分析

- WerFault在用户登陆时执行，新启动的WerFault注册表HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\KernelFaults\Queue的键值
- 如果有之前的崩溃转储数据和没有被发送的崩溃报告，WerFault生成两种XML格式的数据文件

在线崩溃分析

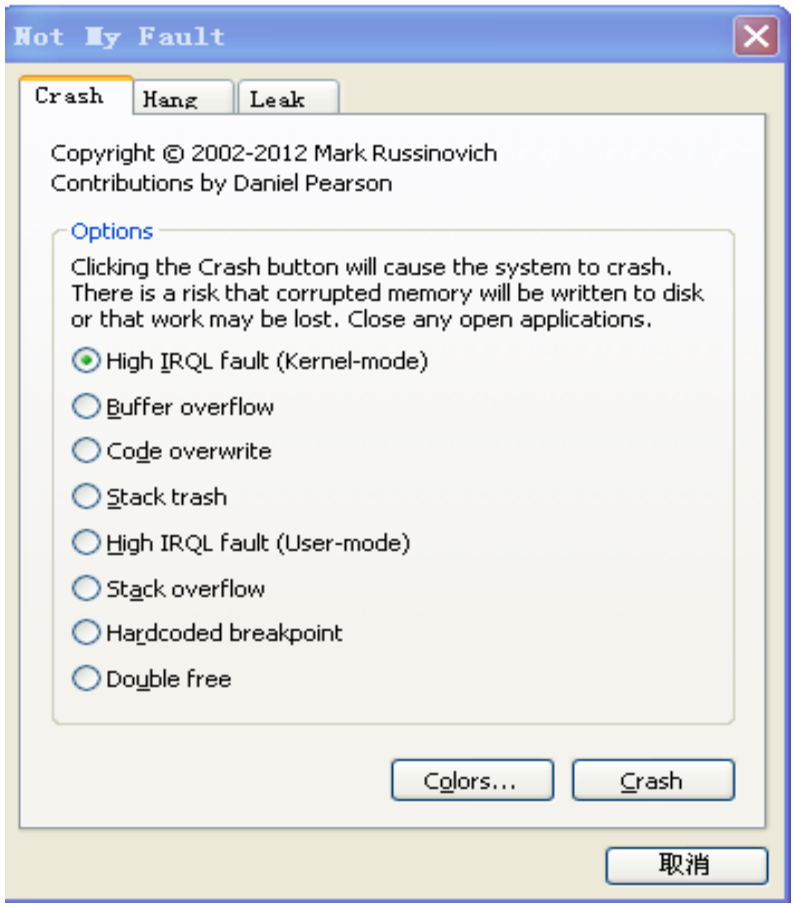
- 第一份XML文件包含了基本的系统描述，包括系统版本，在系统中安装的驱动文件列表，和当前系统中存在的设备列表
- 第二份XML文件包含了供OCA服务器使用的元数据，包括触发错误报告的事件种类和额外的配置信息
- WerFault发送两份XML文件和小转储数据到https://oca.microsoft.com

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

基本的崩溃转储分析

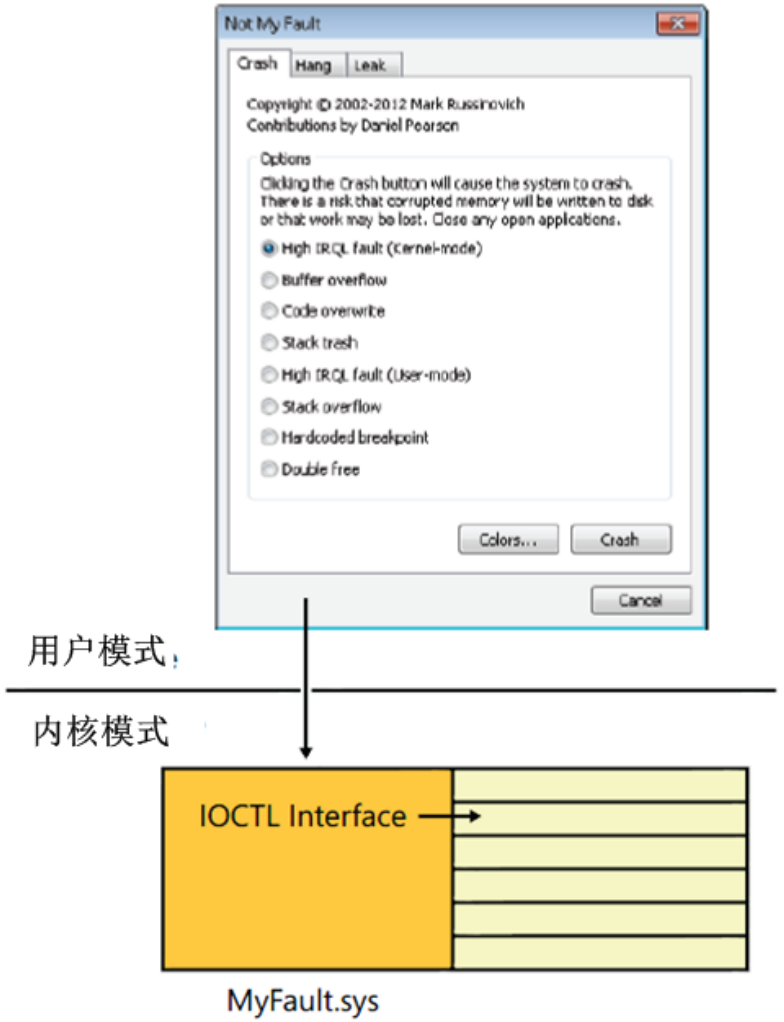
- Notmyfault



下载地址：http://technet.microsoft.com/en-us/sysinternals/bb963901

基本的崩溃转储分析

- Notmyfault结构



基本的崩溃转储分析

- Notmyfault是由一个名为Notmyfault.exe的可执行文件和一个名为Myfault.sys的驱动程序组成的。当你运行NotMyfault可执行程序时，它加载该驱动程序
- 该工具允许你以各种方式来崩溃系统，或者让驱动程序泄露换页内存池
- 工具中提供的崩溃类型代表了Microsoft产品支持服务组看到最常见的系统崩溃

基本的崩溃转储分析

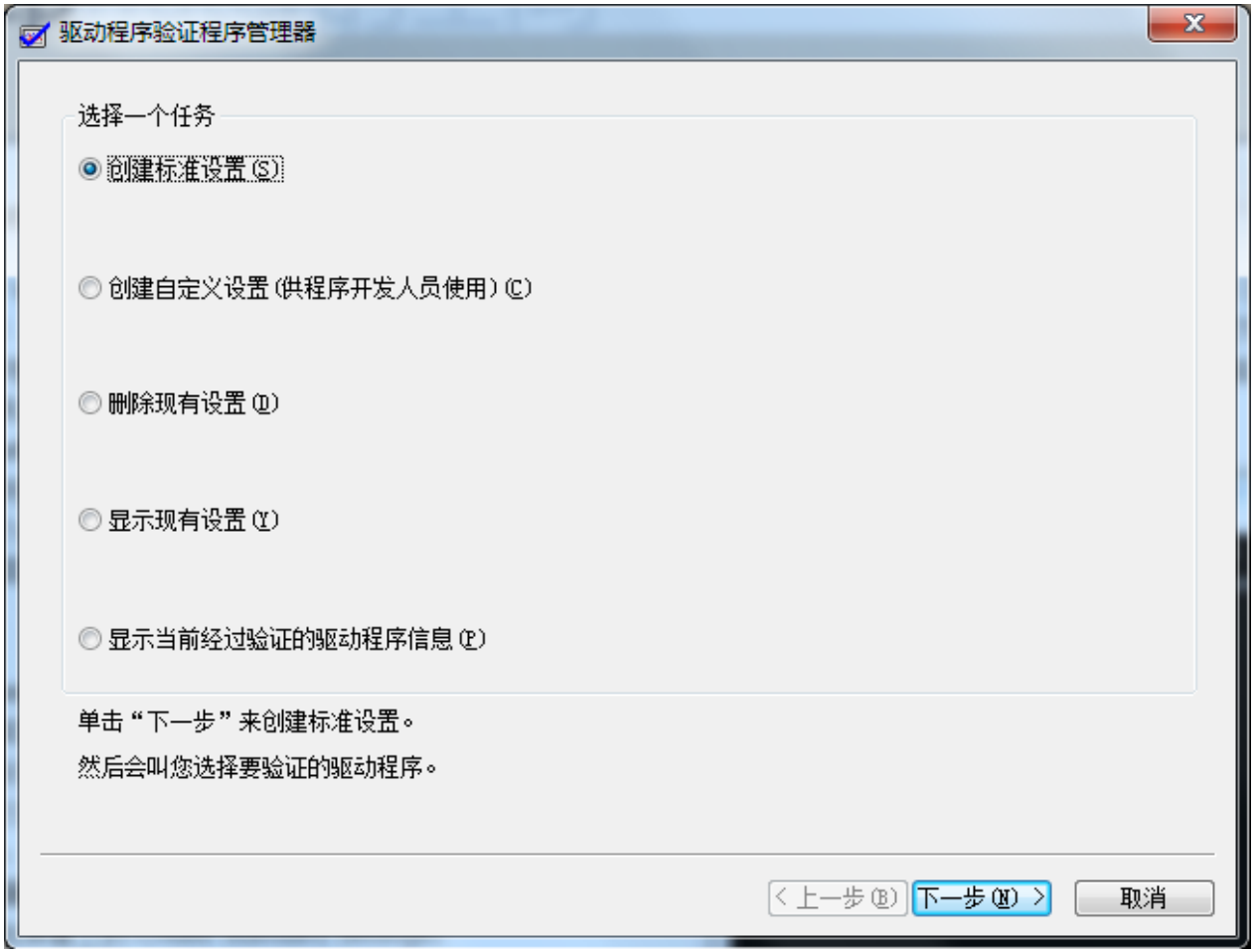
- 最直接的Notmyfault崩溃时通过选择High IRQL Fault（Kernel mode）选项，这使得驱动程序从换页池中申请一个页面，再释放该页面
- 将IRQL提升到DPC/Dispatch级别上，访问刚刚被释放的页面，若不会引起系统崩溃，则进程仍然往下执行
- 越过该页面的尾部继续读内存，直至由于访问无效页面而引起系统崩溃

基本的崩溃转储分析

- 该驱动程序执行了以下的非法操作
  - —引用了不再属于它的内存
  - —在DPC/Dispatch级别或更高级别上引用了换页内存池，这是非法的
  - —当它越过了它所申请的内存尾部时，试图引用一段可能无效的内存

基本的崩溃转储分析

- 使用崩溃诊断工具



基本的崩溃转储分析

- 如果怀疑最近加入的驱动程序可能是崩溃的源头，对这些驱动程序使用驱动程序检验器进行检验，除了低资源模拟选项外，检查所有其他的选项
- 对于系统中未签名的驱动程序，启用第一级别中的检验选项
- 对于系统中给的所有驱动程序，启用第一级别中的检验选项，为了维持合理的性能，可以将驱动程序分成组，每次引导时，只针对一个组启用驱动程序检验器

基本的崩溃转储分析

- 缓冲区溢出和特殊内存池

—在windows上，最常见的崩溃源头是内存池被破坏，内存池破坏通常发生于驱动程序遭受了缓冲区上溢或缓冲区下溢的错误

—这些错误会破坏池跟踪数据结构或者破坏其他驱动程序的缓冲区，这种崩溃本质上是不可能调试的

—运行Notmyfault并选择“Buffer Overflow”错误，可以产生一个破坏内存池类型的崩溃，使得Myfault申请一个缓冲区，改写该缓冲区之后的40字节

基本的崩溃转储分析

- 代码改写和系统代码写保护
  - —如果驱动程序有一个可导致破坏或者不正确的解释其自身数据结构的错误，当它把已破坏的数据解释成一个内存指针值的时候，他可能引用到不属于他自己的内存
  - —运行Notmyfault并选择了“code overwrite”选项时，myfault驱动程序破坏掉NtReadFile内核函数的入口点

本章内容提要

- Windows为什么会崩溃
- 蓝屏
- 崩溃转储文件
- Windows错误报告
- 在线崩溃分析
- 基本的崩溃转储分析
- 高级的崩溃转储分析

高级的崩溃转储分析

- 使用“!cpuinfo”指令显示系统在使用的处理器列表
- 使用进程ID加上k指令系统中每一个处理器的栈
- 使用!thread指令显示每一个处理器当前的进程信息
- 使用.time指令显示系统时间信息，包括系统何时崩溃和它已经运行了多久

高级的崩溃转储分析

- 使用lm指令和k,t选项显示加载的内核驱动列表
- 使用!vm指令查看是否系统耗尽了虚拟内存，分页池或者非分页池
- 使用!process 0 0调试器查看哪些进程正在运行，确保了解每一个进程的用途，试着禁止或者卸载不必要的应用程序和windows服务

高级的崩溃转储分析

- 栈破坏
  - —栈溢出（stack overrun）或栈破坏（stack trash）是由于缓冲区的上溢或下溢错误造成的
  - —目标缓冲区并不像在notmyfault的缓冲区溢出区错误中看到的那样位于内存池中，而是位于执行此错误的线程的栈中
  - —当运行Notmyfault并选择了stack trash时，myfault驱动程序溢出一个缓冲区，该缓冲区是它在当期执行线程的内核栈中分配的

高级的崩溃转储分析

- 当Myfault试图将控制权返回给发起此调用的Ntoskrnl函数时，它从栈中读取返回地址，这是该线程应该继续往下执行的地址，该地址已经被栈缓冲区溢出破坏了
- 该地址处没有包含代码，执行到一条非法CPU指令或者引用了无效内存时，就会引发一个非法异常并崩溃

高级的崩溃转储分析

- 挂起的系统或无响应的系统
- 以下的情况可能导致系统挂起
  - —一个设备驱动程序没有从它的中断服务（ISR）例程或者延迟过程调用（DPC）例程中返回
  - —一个高优先级的实时线程抢占了窗口系统驱动程序的输入线程
  - —在内核模式中发生了死锁

高级的崩溃转储分析

- 可以利用驱动程序检验器中的称为“死锁检测”的选项
- 可以监视自旋锁，快速互斥体和互斥体的使用情况查找可能会导致死锁的模式
- 如果找到了死锁的模式，则驱动程序检验器使当前系统崩溃，并且指明哪个驱动程序引起了死锁

高级的崩溃转储分析

- 有两种方法可以允许进入到一个挂起的系统中，使用手工崩溃诊断技术来确定哪个驱动程序或者组件导致了系统挂起
- 第一种方法：让一个已挂起的系统崩溃，希望得到一个能够进行分析的崩溃转储
- 第二种方法：用内核调试器进入该系统，并分析系统的行为

高级的崩溃转储分析

- 为了手工使一个已挂起的系统崩溃，首先加入一个DWORD注册表值HKLM\SYSTEM\CurrentControlSet\Services\i8042prt\Parameters\CrashOnCtrlScroll，将其设置为1，重新引导后，i8042端口驱动程序监视两次ScrollLock键击，同时右Ctrl键也被按下
- 这时调用KeBugCheckEx,并且停止代码为MANUALLY_INITIATED_CRASH（0xE2），触发一个由用户手工引起的崩溃

高级的崩溃转储分析

- 另一种触发系统崩溃的方法，如果你的硬件有内置的“崩溃”按钮，通过给系统主板的不可屏蔽中断（NMI）阵脚发送一个信号，可以触发崩溃
- 将DWORD注册表值HKLM\SYSTEM\CurrentControlSet\Control\CrashControl\NMICrashDump设置为1
- 当按下转储开关时，系统接到一个NMI，内核的NMI中断处理器调用KeBugCheckEx，这种方式的使用机会更高

课后实验

- 实验一 制造一个蓝屏事件
  - —使用的工具有：Notmyfault，该工具包括一个名为Notmyfault.exe的可执行文件和一个名为Myfault.sys的驱动程序
  - —该程序允许用户以各种方式来崩溃系统
  - —尽量在虚拟机中测试，可能会对文件或者硬盘造成伤害

课后实验

- 选择Notmyfault工具的High IRQL Fault（Kernel mode）选项并引发崩溃，执行非法操作
  - —引用了不再属于它的内存
  - —在DPC/Dispatch级别或更高级别上引用了换页内存池
  - —它越过了它申请的内存的尾部，试图引用一段可能无效的内存
- 在指定的路径生成崩溃转储文件

课后实验

- 实验二 对转储文件进行分析
  - —使用的工具为winDbg，该工具是用来调试windows内核程序和分析内核问题的
  - —下载地址
    http://msdl.microsoft.com/download/symbols/debuggers/dbg_x86_6.11.1.404.msi

课后实验

—安装winDbg软件工具

—导入之前系统崩溃生成的转储文件

—截图说明引起崩溃的原因是myfault.sys驱动程序

—输入命令！ analyze －v 查看详细分析

```
*    using the _NT_SYMBOL_PATH environment variable.
*    using the -y <symbol_path> argument when starting tl
*    using .sympath and .sympath+
**********************************************************
**********************************************************
* Symbols can not be loaded because symbol path is not
*
* The Symbol Path can be set by:
*    using the _NT_SYMBOL_PATH environment variable.
*    using the -y <symbol_path> argument when starting tl
*    using .sympath and .sympath+
**********************************************************
Probably caused by : myfault.sys ( myfault+5ab )

Followup: MachineOwner
---------

kd> !analyze -v
**********************************************************
*
*                        Bugcheck Analysis
*
**********************************************************

DRIVER_IRQL_NOT_LESS_OR_EQUAL (d1)
```

课后实验

- 详细分析的内容说明了关于停止代码和相关参数的说明

```
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

DRIVER_IRQL_NOT_LESS_OR_EQUAL (d1)
An attempt was made to access a pageable (or completely invalid) address at an
interrupt request level (IRQL) that is too high.  This is usually
caused by drivers using improper addresses.
If kernel debugger is available get stack backtrace.
Arguments:
Arg1: e12d7008, memory referenced
Arg2: 00000002, IRQL
Arg3: 00000000, value 0 = read operation, 1 = write operation
Arg4: f8c2a5ab, address which referenced memory

Debugging Details:
------------------
```

课后实验

- 可能出现问题的栈调用的过程

```
DEBUG_FLR_IMAGE_TIMESTAMP:  4f806ca0

READ_ADDRESS: unable to get nt!MmSpecialPoolStart
unable to get nt!MmSpecialPoolEnd
unable to get nt!MmPoolCodeStart
unable to get nt!MmPoolCodeEnd
 e12d7008

CURRENT_IRQL:  2

FAULTING_IP:
myfault+5ab
f8c2a5ab 8b08            mov     ecx,dword ptr [eax]

DEFAULT_BUCKET_ID:  DRIVER_FAULT

BUGCHECK_STR:  0xD1

LAST_CONTROL_TRANSFER:  from f8c2a5ab to 80541683

STACK_TEXT:
WARNING: Stack unwind information not available. Following frames may be wrong.
b24e7b74 f8c2a5ab badb0d00 00000002 b24e7c00 nt!Kei386EoiHelper+0x27db
b24e7bf0 f8c2a9db 82133230 b24e7c34 f8c2ab26 myfault+0x5ab
b24e7bfc f8c2ab26 81f13680 00000001 00000000 myfault+0x9db
b24e7c34 804ef119 82032458 82133230 806d32d0 myfault+0xb26
b24e7c58 80576bff 82032458 82133230 81f13680 nt!IoBuildPartialMdl+0xed
b24e7d00 8056f46c 0000009c 00000000 00000000 nt!NtWriteFile+0x39b7
b24e7d34 8053e638 0000009c 00000000 00000000 nt!NtDeviceIoControlFile+0x2a
b24e7ddc 80542dd2 b20561f0 81e2a1e8 00000000 nt!KeReleaseInStackQueuedSpinLockFromDpcLevel+0xb14
b24e7e90 bf800b92 bf80eee4 b24e76f4 0006faf8 nt!KiDispatchInterrupt+0x5a2
b24e7e94 bf80eee4 b24e76f4 0006faf8 b24e7a38 win32k+0xb92
b24e7eb0 8053e638 00020210 00000020 00020210 win32k!EngFreeUserMem+0x5579
b24e7f08 805206ce 00000000 00000000 00000000 nt!KeReleaseInStackQueuedSpinLockFromDpcLevel+0xb14
b24e7f0c 00000000 00000000 00000000 027fd824 nt!MmTrimAllSystemPagableMemory+0x929e
```

课后实验

- 查看引起问题的myfault.sys驱动程序的详细信息

```
kd> lm kv m myfault.sys
start    end        module name
kd> lm kv m myfault*
start    end        module name
f8c2a000 f8c2b880   myfault    (no symbols)
    Loaded symbol image file: myfault.sys
    Image path: \??\C:\WINDOWS\system32\drivers\myfault.sys
    Image name: myfault.sys
    Timestamp:        Sun Apr 08 00:34:40 2012 (4F806CA0)
    CheckSum:         00003871
    ImageSize:        00001880
    File version:     4.0.0.0
    Product version:  4.0.0.0
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:        3.7 Driver
    File date:        00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Sysinternals
    ProductName:      Sysinternals Myfault
    InternalName:     myfault.sys
    OriginalFilename: myfault.sys
    ProductVersion:   4.0
    FileVersion:      4.0 (sysinternals.com)
    FileDescription:  Crash Test Driver
    LegalCopyright:   Copyright © 2002-2012 Mark Russinovich
```