软件调试学习笔记

CPU的操作模式：

1. 保护模式：段的保护机制和页的保护机制。
2. 实模式：可以直接访问物理内存和I/O空间。
3. 虚拟8086模式：保护模式下用于执行8086程序的准模式，通过该模式可以将8086程序当做保护模式的一项任务来执行。
4. 系统管理模式：用于系统固件指向电源管理，安全检查和平台相关的特殊任务，当CPU的系统管理终端管脚(SMI#)被激活，处理器将上下文保存，然后切换到另一个单独的地址空间中执行专门的SMM例程，SMM例程通过RSM指令是处理器退出SMM模式并恢复到系统管理中断前的状态。感觉与VT技术有点像。
5. IA-32e模式：应用程序可以是32位也可以是64位，内核和驱动程序必须是64位，也就是说如果为32位应用程序，在进入系统内核时会从32位切换到64位，从内核返回时再从64位切换回32位。

CPU模式之前的转换：

//todo

处理器通电开始运行或复位后处于实模式，CR0控制寄存器的PE(Protection Enable)标志用来控制处理器是处于实模式还是保护模式。EFLAGS标志寄存器的VM标志用来控制是在虚拟8086模式还是保护模式(之前在逆向函数过程中, 会经常看见对应标志位)，EFER寄存器(Extend Feature Register)的LME(Long Mode Enable)用来启用IA-32e模式.

进程资源：

1. 虚拟地址空间
2. 进程ID
3. 可执行文件镜像
4. EPROCESS结构体
5. 一个或者多个线程
6. 进程句柄表
7. CR3页目录基地址
8. PEB
9. 访问令牌

使用命令：

.process PID(代表所有进程) 0(代表要显示进程的属性) + [过滤条件]

ex: .process 0 0 notepad.exe

进程Session ID, 是指该进程所在的windows会话(session id)的ID号，当有多个用户同时登录时，windows会为每个登录用户建立一个会话，每个会话有自己的Work Station和 Desktop。

windows XP, 当只有一个用户登录时，用户启动的程序和系统服务程序都运行在session 0。当切换到另一个用户后，系统会建立session 1。从Vista之后(server 2003, windows 7及之后)，

系统服务运行在session 0，当用户登录后，会创建另一个会话session 1, 在任务管理器可以看到两个CSRSS进程在运行。特殊的系统进程不属于任何会话。

CR3(页目录基地址), X86(XP, 7, 8, 10等)下使用2-9-9-12(32位)分页模式，X64下使用9-9-9-9-12分页模式(48位)，如图所示：

WoW进程：

64位系统内核和驱动程序代码都是64位的，但用户空间的代码既存在32位又存在64位，针对运行在64位内核上的32位进程专门有一个名称，成为WoW进程。

32位程序需要使用老的32位WIN32API和一些库函数，所以在64位windows系统的目录里，存在SysWow64目录，里面专门用于存放32位版本的程序文件和动态链接库。而system32中存放的是64位的程序文件。

32位调试器无法调试64位程序，但64位调试器即可以调试32位又可以调试64位程序，windbg可以使用.effmach amd64|x86命令在两种模式下进行切换。

通过调试我们可以发现，在WoW进程中会加载两个NTDLL模块，一个是64位的，一个是32位的。WoW进程中的32位版本NTDLL.DLL在执行系统调用时，会调用特殊的Wow64SystemServiceCall函数，切换到64位的WoW转接层，从而进入64位内核调用具体实现。

注册表的重定向：

在调试WoW进程时，如果需要查看或者修改注册表，处于多种原因，注册表有可能会被重定向到Wow6432Node。

文件重定向：

在WoW进程中，存在两个NTDLL.DLL，当32位的WoW进程访问系统文件目录时，会被自动重定位到SysWOW64目录。

注册表的反射：

有些COM组件既有32位版本，又有64位版本。来自任何一方的修改都会自动更新到另一边。这样的表键如下：

HKLM\SoftWare\Classes;

HKLM\SoftWare\Ole;

HKLM\SoftWare\Rpc;

HKLM\SoftWare\Com3;

HKLM\SoftWare\EventSystem;

HKLM\SoftWare\CLSID;(只适用于进程外组件)

创建进程的过程：

1. 在父进程的用户空间打开要执行的可执行文件镜像，确定其名称，类型和系统的设置选项。
2. 进入父进程的内核空间位新进程创建EPROCESS结构体，进程地址空间，KPROCESS结构体以及PEB。
3. 以挂起方式创建线程
4. 通知子系统服务程序，对于windows应用程序，通知子系统服务进程(CSRSS)
5. 初始线程开始在内核空间执行。
6. 通过APC机制，在新进程自己的用户空间中执行初始化动作，通过NTDLL.DLL的加载器加载进程所依赖的DLL.

最小进程：

最小进程是指在创建进程时，指定特殊标志，只创建进程空间，无需向进程空间添加内容。

注册表进程，其主要作用是缓存注册表数据，以便提高注册表的访问效率，降低与注册表有关的内存开销。

任务管理器：

系统空闲进程（IDLE）进程的进程ID总是为0，该进程的线程数等于系统中的总的CPU的个数。

用户空间重要进程：

1. 会话管理进程(SMSS.EXE), 他是系统中第一个根据镜像文件创建的进程，是在系统启动后期由执行体的初始化函数创建的。它运行后，会加载和初始化win32子系统的内核模块Win32k.sys，创建win32子系统服务进程（CSRSS.EXE），以及登录进程（WINLOGON.EXE）。
2. Windows子系统服务器进程(CSRSS)，为子系统提供服务。例如登记进程和线程，管理控制台窗口，管理DOS程序虚拟机(VDM)进程。
3. 登录进程（WINLOGON.EXE）,负责用户登录和安全有关事务。该进程启动后，会创建LSASS进程和系统服务管理进程（SERVICE.EXE）
4. 本地安全和认证进程(LSASS.EXE),负责用户身份验证。
5. 服务管理进程（SERVICE.EXE），负责启动和管理系统服务程序。Ex, SpoolSv.exe是打印脱机服务, WmiPrvSE.EXE是WMI提供器管理服务, SVCHOST是一个通用的服务宿主程序。

内核文件：

1. 单处理器版本：AMD64: NTOSKRNL.EXE

X86: 支持PAE NTKRNLPA.EXE

不支持PAE NTOSKRNL.EXE

1. 多处理器版本：AMD64: NTKRNLMP.EXE

X86: 支持PAE NTKRPAMP.EXE

不支持PAE NTKRNLMP.EXE

从Vista开始即使系统只有一个CPU，也会使用多处理器版本。

系统特殊进程：

空闲进程(IDLE PROCESS)和系统进程(SYSTEM PROCESS)

共同的特征：

1. 没有对应的可执行文件镜像。
2. 不存在用户空间，只有内核空间。
3. 具有固定的进程ID, IDLE PORCESS进程ID为0，SYSTEM PROCESS进程ID为4（XP之后）。

空闲进程：

CPU通电后就需要不断的取指令并且执行，如果没有事情，也需要给它几条指令，组成一个循环，让它在哪里空转。在现今的操作系统中，被设计成空闲线程，当CPU没有其他线程需要执行时，让其执行空闲线程。由于系统中的每个CPU都需要空闲线程，因此系统空闲进程中的线程数与处理器的个数是一致的。在执行体阶段0初始化时，进程管理器的初始化函数便会创建空闲进程和第一个空闲线程。由于系统空闲进程时系统启动时创建的第一个进程，而且一直存在，所以可以通过观察该进程的运行时间来推测系统启动的时间。

系统进程：

系统进程是操作系统内核和所有系统线程的宿主，其作用是为操作系统系统独立的进程空间和进程对象。在系统启动阶段，进程管理器创建空闲进程后，便创建系统进程，即系统进程是系统创建的第二个进程。

NTDLL:

系统在启动阶段便会将NTDLL加载到内存中，并把它映射到所有用户进程的进程空间中，并且映射在相同的虚拟地址。

镜像文件的加载：

编译好的程序是存储在磁盘上的，运行是需要先加载到内存中，NTDLL中的LDR负责执行该任务。