**附录 E WinDBG 命令索引**

| 分类 | 命令 | 描述 | 示例（$$后为注释） | 章号 |
|---|---|---|---|---|
| 帮助 | ? | 显示所有标准命令 | ? | |
| | .help | 显示所有元命令 | .help<br>.help /D　　$$显示带有 DML 导航链接的帮助<br>.help /D r* | |
| | .hh | 打开帮助文件 | .hh .reload | |
| | !help | 显示扩展命令 | !ext.help | 22 |
| | | | !help dumpheap | 22 |
| | .chain | 显示加载的插件 | .chain /D | 22 |
| | !error | 显示错误码含义 | !error c0000034 | 2 |
| | | | !error 6 | 12 |
| | | | !error　80004005h | 17 |
| | version | 显示版本信息 | version | |
| 表达式 | ? | 评估宏汇编表达式 | ? a27-65e | 10 |
| | ?? | 评估 C++表达式 | ?? (unsigned int )-1073741819<br>?? &(this->m_Button1) | |
| | ; | 命令分隔符 | r;kv;gc | |
| | $$ | 注释符号 | | |
| | poi | 取地址内容 | dd poi(@ebp+8) | |
| | .expr | 显示或改变表达式类型 | .expr /s c++ | |
| | n | 设置数制 | n 10 | |
| | .formats | 以多种格式显示数 | .formats 80000001 | 27 |
| 调试会话 | .create | 创建新的进程并调试 | .create notepad.exe | |
| | .attach | 附加到指定进程 | .attach 2568　　$$ 2568 是进程 ID | |
| | .restart | 让调试目标重新运行（用户态） | .restart<br>.restart /F | 12 |
| | .reboot | 重启启动目标系统（内核态） | .reboot | |
| | .crash | 强制目标系统崩溃（内核态） | .crash | |
| | .detach | 分离调试目标 | .detach | |
| | q | 终止调试 | q | |
| 模块和符号 | .symfix | 设置符号服务器 | .symfix c:\symbols | 6, 15 |
| | .reload | 刷新模块和符号 | .reload | 6 |

| | | | | |
|---|---|---|---|---|
| | | | .reload /f ntfs.sys | |
| | | | .reload /user | 7 |
| | ld | 加载符号文件 | ld kernel32 | |
| | .sympath | 显示或设置符号路径 | .sympath<br>.sympath+ d:\work\debug | 10 |
| | !sym | 设置符号选项 | !sym noisy<br>!sym quiet | |
| | .symopt | 设置符号选项 | .symopt+ 40 $$ 加载不严格匹配的 PDB | |
| | x | 显示符号 | x winmine!* | 29 |
| | | | x ole32!*BindToObject | 8 |
| | ln | 搜索符号 | ln aebc6eac | 10 |
| | | | ln poi(0000001`400520b8) | 14 |
| | lm | 列模块或显示模块详情 | lm vm urlmon<br>lmD | 4 |
| | !lmi | 模块和 PDB 文件详情 | !lmi nt | |
| | !dlls | 动态库信息 | !dlls -a | |
| | !imgreloc | 模块的重定位信息 | !imgreloc | |
| | !dh | 观察 PE 文件头 | !dh 76930000 -a | 30 |
| 转储文件 | .dump | 产生转储文件 | .dump /mfh c:\dumps\pdf\acrobat.dmp | 9 |
| | ! analyze | 自动分析 | ! analyze -v | 10, 15 |
| | .writemem | 将内存数据写到文件 | .writemem c:\dumps\blog.txt 07288600 L2000 | 1 |
| | .bugcheck | 显示蓝屏错误码 | .bugcheck | |
| | adplus （脚本） | 监视进程和自动转储 | Adplus -pn powerpnt.exe -pn wincmd32.exe -hang -o c:\test | 8 |
| 进程 | \| | 显示或切换进程（用户态） | \|*<br>\|2s | |
| | !process | 显示进程（内核态） | !process 0 0<br>!process 0 0 powerpnt.exe<br>!process 8 1<br>!process 88bfbb80 2 | 2, 10, 8 |
| | !dml_proc | 观察进程信息 | !dml_proc $$ 以 DML 方式显示进程信息 | |
| | .process | 显示或切换进程（内核态） | .PROCESS /i 9382a530 | 7 |
| | .kill | 杀进程 | .kill 8adc85f0 $$ 参数为 EPROCESS 地址 | |
| | !peb | 观察进程环境块 | !peb | |
| | dt | 观察数据结构 | dt nt!_EPROCESS<br>dt ntdll!_PEB @$peb -r | |

| | | | | |
|---|---|---|---|---|
| | .tlist | 列进程 | .tlsit $$列的是调试器所在系统的进程 | |
| 线程 | ~ | 显示线程 | ~<br>~* | |
| | | 切换线程 | ~0s $$0 是线程序号 | 8, 9 |
| | | | ~~[14d8]s $$ 14d8 为线程 ID | 13 |
| | | 针对 1 或多个线程执行命令 | ~* kv1 | 9 |
| | | | ~*e ? @$tid;!gle | 13 |
| | | | ~1 k | 9 |
| | !teb | 显示线程环境块 | !teb | 29 |
| | .thread | 显示或切换线程（内核态） | .thread 80551d20 | 15 |
| | | | .thread /p 87474da8 | 16 |
| | !thread | 观察线程（内核态） | !thread | 10 |
| | | | !thread 9383a030 | 7, 8 |
| | dt | 观察数据结构 | dt nt!_ETHREAD | |
| | | | dt _TEB -y Last | 13 |
| | | | dt _CONTEXT | 29 |
| | | | dt _CONTEXT 0007fd30 | 33 |
| | !wow64exts.info | 运行在 64 位系统的 32 位线程信息 | !wow64exts.info | |
| 内存 | !address | 观察内存空间 | !address | 1 |
| | | 观察内存块属性 | !address 0728988a | 1 |
| | s | 搜索内存数据 | s -u 10000 L8000000 "当年在交大" | 1 |
| | d | 显示内存数据 | db e1c52ce0 | 2 |
| | | | da 000000000a4a9640 | 19 |
| | | | dd @ebx l1 | 9 |
| | | | dd bc74ed08 l1 | 8 |
| | | | dd 805490c4+8 l1 | 15 |
| | | | dU /c 50 poi(0423ed44+8) | 12 |
| | | | dS fc8d3528 | 30 |
| | | | du nt!NtInitialUserProcessBuffer | 30 |
| | | | db 0xe17734ac+14 | 30 |
| | dt | 按类型显示 | dt _GUID 01f1b434 | 4 |
| | | | dt _EXCEPTION_POINTERS 74c97038 -r | 6 |
| | e | 编辑内存数据 | ew 02c9ffcc $$ 按字(Word)编辑 02c9ffcc 开始的内存 | 1 |
| | | | eb 001b5942-8 ff fe | 1 |
| | | | ed f655cfb4 f655d7dc | 10 |
| | !dd | 读物理地址 | !dd ffffffe0 | |
| | !ed | 写物理地址 | !ed ffffffe0 80000000 | |
| | !vtop | 虚拟地址转换到 | !vtop 0 bafde064 | 15 |

| | | | | |
|---|---|---|---|---|
| | | 物理地址 | | |
| | !pte | 显示页表项 | !pte f655cfb4 | 10 |
| | !memusage | 显示物理内存使用情况 | !memusage | |
| | !pool | 显示内核池信息 | !pool e326c000 | |
| | !vad | 观察进程的地址空间 | !vad 8a760500 | |
| | !sdbgext.hwnd | 观察窗口句柄 | !sdbgext.hwnd 001506c4 | 8 |
| 堆 | !heap | 显示堆信息 | !heap 01670000 -A | 34 |
| | dt | 观察数据结构 | dt _heap_entry 0167fc10 | 34 |
| | !gflag | 观察或者启用堆的调试支持 | !gflag<br>!gflag +hpa | |
| 栈 | k | 函数调用序列 | kn 100 | 4, 8 |
| | | | kn | 14 |
| | | | k =f655d7dc aebc3e5a aebc3e5a 99 | 10 |
| | .frame | 切换当前栈帧 | .frame /c c | 4 |
| | dds | 显示数据和符号 | dds 80a056e0 | 27 |
| | dv | 显示局部变量 | dv -V | |
| 寄存器 | r | 读写寄存器 | r cr3 | 10 |
| | | | r cs, ds, es, fs, gs, ss | 29 |
| | | | r cr2 | 10 |
| | | | r eip=0042D69E | 29 |
| | | | r ecx=poi(ecx);r ecx; z(ecx!=0) | 17 |
| | rdmsr | 读 MSR 寄存器 | rdmsr 19c | 29 |
| 上下文 | .ecxr | 切换到异常上下文 | .ecxr | 14 |
| | .tss | 切换到指定 TSS | .tss 28 | 10 |
| | .trap | 切换到陷阱帧 | .trap 8089a570 | |
| | .effmach | 切换 32/64 位上下文 | .effmach x86<br>.effmach amd64 | |
| 断点 | bp | 软件断点 | bp KERNELBASE!CreateFileW+0x5 "dU /c 50 poi(@ebp+8);gu;r eax;.if(@eax<0){.echo hit}.else{gc}" | 12 |
| | | | bp KERNELBASE!CreateFileW+0x5 "dU /c 50 poi(@ebp+8);gc" | 12 |
| | | | bp `d4dtest!d4dtestdlg.cpp:196` | |
| | ba | 硬件断点 | ba w4 0006fc74 | 17 |
| | bm | 成批设置断点 | bm /a nt!Dbgk* | |
| | bd/be/bc/bl | 管理断点 | bl | |
| 执行和跟 | g | 恢复执行 | g<br>gu $$ 返回到父函数 | 12 |

| | | | | |
|---|---|---|---|---|
| 踪 | | | | |
| | p | 单步执行 | p<br>p “dv”　　$$ 单步后自动执行 dv 命令<br>pc　　$$ 单步到下一条 CALL 指令 | |
| | t | 单步跟踪 | t<br>tc　$$ 跟踪到下一条 CALL 指令<br>tb　$$ 跟踪到下一条分支指令 | |
| | wt | 自动追踪 | wt –l 3 | 28 |
| | | | | |
| 反汇编 | u | 反汇编 | u nt!PsGetCurrentProcessId | 10 |
| | uf | 反汇编整个函数 | uf RtlLeaveCriticalSection | 14 |
| | ub | 反向反汇编 | ub 773c78e9 | 14 |
| | a | 汇编 | a <地址> | |
| 死锁 | !locks | 扫描关键区（用户态）或 ERESOURCE（内核态） | !locks | 9 |
| | !cs | 观察关键区 | !cs -l | 9 |
| | !alpc | 观察 ALPC 端口 | !alpc /m 9322a230 | 7 |
| | dt | 观察数据结构 | dt _RTL_CRITICAL_SECTION | 9 |
| | | | dt _ERESOURCE<br>dt _KEVENT -r | |
| 处理器 | !pcr | 管处处理器控制区 | !pcr | 15 |
| | dg | 观察段描述符 | dg @fs | 15 |
| | !idt | 观察 IDT 表 | !idt -a | 29 |
| | !cpuid | 显示 CPU 型号 | !cpuid | |
| | !cpuinfo | 显示 CPU 属性(内核态) | !cpuinfo | |
| | !irql | 显示保存的 IRQL | !irql | |
| | dt | 观察数据结构 | dt _KPCR | |
| 驱动程序 | !drvobj | 显示和观察驱动对象 | !drvobj ser2pl | 16 |
| | !devobj | 观察设备对象 | !devobj 85163500 | 16 |
| | !devstack | 观察设备栈 | !devstack 85163500 | 16 |
| | !devnode | 观察设备节点 | !devnode 0 1 | 16 |
| | !irp | 观察 IRP | !irp 85163e70 | 16 |
| | dt | 观察数据结构 | dt _IO_STACK_LOCATION 85879d60<br>dt _IRP<br>dt _DRIVER_OBJECT<br>dt _DEVICE_OBJECT | 16 |
| | !arbiter | 显示资源分配情 | !arbiter 2 $$ 物理地址的分配情况 | 15 |

| | | | | |
|---|---|---|---|---|
| | | 况 | | |
| | !pci | 观察 PCI 空间 | !pci ff 0 2 0 0 ff | 28 |
| | !pcitree | 显示 PCI 设备 | !pcitree | |
| | ib/iw/id | 读 I/O 端口 | ib 510 | |
| | ob/ow/od | 写 I/O 端口 | ob 510 0a | |
| | !amli | AMLI 调试器 | !amli dns /s _sb_.pci0.bat0 | 24 |
| 托管程序 | .loadby | 加载扩展命令模块 | .loadby sos mscorwks | 13 |
| | .load | 加载扩展命令模块 | .load clr10\sos.dll | 22 |
| | !name2ee | 显示名字对应的对象信息 | !name2ee clihello CliHello.CliHello.Main | 22 |
| | !ip2md | 显示程序地址对应的方法描述 | !ip2md 0x7ff`001c0dbd | 13 |
| | !threads | 观察线程的托管 | !threads | 13 |
| | !clrstack | 托管栈回溯 | !clrstack | 13 |
| | !do | 显示托管对象 | !do 0000000002901750 | 13 |
| | !bpmd | 设置断点 | !bpmd CliHello CliHello.CliHello.Main | 22 |
| 注册表 | !reg | 操作注册表 | !reg openkeys "hklm\software\microsoft\windows nt\currentversion\winlogon" | 30 |
| | | | !reg kcb e101a7a0 | 30 |
| | | | !reg kvalue 0xe17734ac | 30 |
| | | | !reg cellindex 0xe1442920 721380 | 30 |
| 调试事件 | sxe | 配置调试事件 | sxe ld:portabledeviceapi<br>sxd ld | 35 |
| | gn | 调试器不处理异常 | gn | |
| | gh | 调试器处理异常 | gh | |
| | .lastevent | 显示上一个调试事件的信息 | .lastevent | |
| 远程调试 | .server | 将调试器升级为调试服务器 | .server tcp:port=2000 | 7 |

**注：**本表摘自《格蠹汇编——软件调试案例集锦》一书，最右一列为该书的章号。欢迎访问高端调试网站（http://www.advdbg.org）获取更多软件调试有关的资源或者参与技术讨论。

**附录 F 常用的汇编指令（x86）**

| 指令 | 功能 | 示例 | 机器码（十六进制） |
|---|---|---|---|
| INT 3 | 软件断点 | int 3 | CC |
| NOP | 空操作 | nop | 90 |
| PUSH | 压栈 | push ebp | 55 |
| POP | 从栈中弹出 | pop ecx | 59 |
| ADD | 加法 | add ecx,eax | 2BC1 |
| SUB | 减法 | sub esp,4Ch | 83EC4C |
| IDIV | 整数除法 | idiv eax,ecx | F7F9 |
| CALL | 调用函数 | call dword ptr [winmine!_imp__LoadIconW (010010ac)] | FF15AC100001 |
| RET | 返回到父函数 | ret | C3 |
| SYSCALL | 调用系统服务 | syscall | 0F05 |
| INC/DEC | 递增/递减 | inc eax | 40 |
| MOV | 赋值 | mov dword ptr [winmine!cBombLeft (01005194)],eax | A394510001 |
| JMP | 无条件跳转 | jmp winmine!DrawBombCount+0x43 | EB08 |
| TEST | 逻辑比较 | test eax,eax<br>; EAX AND EAX, 结果为 0 则置 ZF 位(Zero Flag)，最高位为 1 则置 SF 位（符号位） | 85C0 |
| CMP | 数学比较 | cmp eax,edi<br>; EAX - EDI,结果为 0 则置标志寄存器的 ZF 位，为负时则置 SF 位 | 3BC7 |
| JE/JNE | 条件跳转（等/不等） | jne winmine!WinMain+0x135 (01002325) | 7507 |
| JB/JNB | 条件跳转（小于/不小于） | jb WINSTA!WinStationSendWindowMessage+0xa4 | 0f826accffff |
| JL/JGE | 条件跳转（） | jge winmine!DrawBombCount+0x3b (010027c0) | 7d0d |
| XOR | 异或 | xor eax,ebp | 33C5 |
| LEA | 取有效地址 | lea eax,[ebp-30h] | 8D45D0 |
| MOVS | 串赋值 | rep movs byte ptr es:[edi],byte ptr [esi]<br>; ESI 指向的内存区循环赋值到 EDI 指向的内存区，循环次数在 ECX 中 | F3A4 |
| STOS | 串存储 | rep stos dword ptr es:[edi]<br>; 将 EAX 寄存器的值循坏写到 EDI 开始的内存中，循环次数在 ECX 中 | F3AB |