

Index

NUMBERS

16 bit applications, 29
32 bit application interrupt, 201-202
80286 microprocessor, 34
80386 microprocessor
 address translation mechanism, 37
 addressing, 35-38
 page address translation, 37
 switching memory context, 78-79
 Windows 3.1 usage, 34

A

ActualMessageLength, 151
address spaces. *See also* memory management
 flat addressing, 34-35, 38
 process isolation, 35, 38-39
 shared spaces, 26-27, 39-48
 Win32 API allocation, 26-27
 Windows 95/NT differences, 26-27
AddressOfCallBacks, 247
AddressOfIntrinyPoint, 232
AddressOfFunctions, 237-238
AddressOfIndex, 247
AddressOfNameOrdinals, 238-239
AddressOfNames, 238
AddressOfRawData, 245
ADVAPI32 system ID mappings, 124
API calls. *See* application programming interface (API) calls
AppImt_DLL, 27
application programming interface (API) calls
 calls added by NT, 30-31
 hooking, 62-78
 native API, 145
 standard API calls, 27
 translation to local procedure calls (LPC), 8
 .ASM files, 19
 assembly language patterns, 99-100

B

Base
 ImageRvaToVa parameter, 226
 PE export directory field, 237
BaseOfCode, 232
BaseOfData, 232
binary compatibility, 4
binary search trees, 51
BIND_ALL_IMAGES, 242
BindImageO, 241
BindImageExO, 242
BIND_NO_BOUND_IMPORTS, 242
BIND_NO_UPDATE, 242
bitness, 29
BUILD.EXE, 18

C

Cache Manager, 12-13
CALLGATE.DLL, 209
callgates
 32-bit compiler involvement, 209
 creating using CreateCallGate, 43-44
 paging issues, 220-221
 privilege level, 208
 privileged code execution, 207-209
 selector handling, 208-209
 stack selection, 208
 usage example, 40-41, 209-220
CallGateSelector, 73
CALLGATE.SYS, 209
calling conventions, 87-88
cdecl calling convention, 87
CfuncGetPageDirectoryO, 41
CfuncGetPhysicalAddressAndPageAttri
 butesO, 73, 74
CGATEAPP.EXE, 209
CGATEDLL.DLL, 209
Characteristics
 COFF header field, 230
 MapAndLoad parameter, 228
 PE debug directory field, 245

 PE export directory field, 237
 PE resource directory field, 243
 PE section table member, 248
 thread local storage value, 247
CheckImageSize, 233
Checksum, 235
client(), 164, 174, 183
CMD.EXE, 18
CmLogLevel, 96-97
CmLogSelect, 96-97
CodePage, 244
CodeView debugger, 245
COFF files, 225
COFF style headers, 230-232
compatibility, 4
compiler code generation patterns, 89-90
connection information length, 148
core
 defined, 9
 HAL layer, 10
 kernel layer, 10
 NT Executive layer, 10-15
CPL (Current Privilege Level), 36
CR3 register, 37, 79
CreateCallGateO, 43
CreateFileMapping(), 71, 147
 usage example, 174
CreateProcess, 27-28
CreateRingOCallGateO, 72, 75
CreateToolHelp32Snapshot, 28
CSRSRV.DBG, 86
CSRSS subsystem, 111, 125, 155
 interrupt processing, 191
CSRSS.DBG, 86
CSRSS.EXE, 8, 25, 92-93
Current Privilege Level (CPL), 36

D
DataDirectory[IMAGE_NUMBEROF_DI
 RECTORY_ENTRIES], 236
.DBG files, 85

- DDK. *See* Device Driver Kit (DDK)
- debugging
 - hooking system services, 112
 - Portable Executable (PE) files, 245
- debugging flags
 - CmLogLevel, 96-97
 - CmLogSelect, 96-97
 - ExpEchoPoolCalls**, 91
 - LpcptTraceMessages, 91-92
 - MmDebug, 92-93
 - NtGlobalFlag, 94-95
 - ObDebugFlags, 94
 - ObpShowAllocAndFree, 91
 - SepDumpSD, 95-96
 - TokenGlobalFlag, 96
- DecideTheInstanceAndActO, 72
- DecideTheInstanceAndDoTheThings()**, 72
- Device Driver Kit (DDK), 17
 - Start menu shortcuts, 18
- device drivers
 - kernel extension use, 141
 - kernel-mode device drivers, 30
 - layering, 12
 - MyDnverEntry macro, 20-21
 - pseudo device drivers, 19-20, 141
 - structure, 19-22
 - virtual device drivers (VXD), 30
 - Windows 95/NT differences, 29-30
 - writing, 17-22
- DeviceIoControl, 141
- dirty bits, 37-38
- dirty pages, 49, 66
- DisplayPageDirectoryO, 42
- DisplayPageDirectoryForAllProcesses(), 81
- DisplayVirtualAndPhysicalAddresses()**, 68
- DllCharacteristics, 235
- DllPath, 228, 241
- DLLs. *See* dynamic link libraries (DLLs)
- DOS system services, 109-110
 - hooking, 113
- DotDll**, 228
- downward compatibility, 4
- Dr. Watson
- DnverDispatch, 21
- DriverEntry, 19
- DnverObject, 21
- DriverUnload, 21, 22
- DUMPBIN utility, 97
- DumpVad()**, 59
- DWORD, 41, 208
 - DWORD Table, 141
 - dynamic link libraries (DLLs)
 - hooking DLL API calls, 62-78
 - loading, 62-63
 - Portable Executable (PE) files, 240
 - relocation, 62, 78
 - shared code pages, 35, 39-48, 49, 69-70
 - shared code pages, writing to, 62
 - service descriptor shadow table
 - usage, 127-128
 - system ID mappings, 124
 - thunking**, 29
 - USER32.DLL similarity, 9
- GDI.EXE, 24
- GOT. *See* Global Descriptor Table (GOT)
- GetPageAttributesString()**, 66
- GetPageDirectory()**, 43
- GetPhysicalAddressAndPageAttributes(), 68, 74
- Global Descriptor Table (GOT), 36
 - selectors, 208-209
- H**
- HAL (hardware abstraction layer), 10
- HalGetInterruptVectorO, 192
- handle maintenance, 11
- handlers, 125-126
- hardware abstraction layer (HAL), 10
- hFile, 228
- hooking API calls, 62-78
- hooking software interrupts, 191-198
- hooking system services
 - DOS, 113
 - example using NtCreateFileJ, 118-122
 - hacking, relation to, 112
 - kernel data structures involved, 116-117
 - kernel level hooking, 113
 - need for, 111-112
 - user level hooking, 113
 - Windows 3.x, 113-114
 - Windows 95/98, 114
- I**
- I/O Manager
 - cache management, 12
 - device driver model, 12
 - network drivers, 13
- IDTR (Interrupt Descriptor Table Register), 189, 190
- IDTs (Interrupt Descriptor Tables), 117, 190
- ImageBase, 227, 232
- MAGE_DIRECTORY_ENTRY_EXPORT, 237
- ImageDirectoryEntryToDataO, 236
- IMAGE_EXPORT_DIRECTORY, 237
- E**
- EAX register, 103, 117, 126, 183, 203
- ECX register, 104
- EDI register, 104
- EDX register, 103, 117, 126, 203
- ejfanew, 229
- EndAddressOfRawData, 247
- environment subsystems, 144-145
- error handling code, 98
- event trapping, 111
- ExAllocatePoolWithTag, 91
- ExFreePool, 91
- ExpEchoPoolCalls, 91
- extensibility features, 3-4
- F**
- fastcall, 87-88
- fDOSImage, 228
- fGoingDown, 233
- FIFO (First-In-First-Out), 49
- file systems layering, 12
- FileAlignment, 234
- FileHeader, 228
- First-In-First-Out (FIFO), 49
- FirstInstanceO, 69
- FirstPage, 51
- FirstThunk, 240
- fork system calls, 63
- ForwarderChain**, 239
- forwarders, 237
- fReBase
- fRebaseSysfileOk, 233
- FreeDll()**, 65-66
- fSystemImage, 228
- G**
- GDI32.DBG, 86
- GDI32.DLL
 - functionality in Windows 2000/NT, 125

IMAGE_FILE_AGGRESSIVE_WS_TRIM, 231
IMAGE_FILE_32BIT_MACHINE, 231
IMAGE_FILE_BYTES_REVERSED_HI, 231
IMAGE_FILE_BYTES_REVERSED_LO, 231
IMAGE_FILE_DEBUG_STRIPPED, 231
IMAGE_FILE_DLL, 230
IMAGE_FILE_EXECUTABLE_IMAGE, 230
IMAGE_FILE_LINE_NUMS_STRIPPED, 231
IMAGE_FILE_LOCAL_SYMS_STRIPPED, 231
IMAGE_FILE_NET_RUN_FROM_SWAP, 231
IMAGE_FILE_RELOCS_STRIPPED, 231
IMAGE_FILE_REMOVABLE_RUN_FROM_SWAP, 231
IMAGE_FILE_SYSTEM, 230
IMAGE_FILE_UP_SYSTEM_ONLY, 231
IMAGEHLP.DLL, 226, 227, 233
IMAGE_IMPORT_DESCRIPTOR, 239
ImageName, 228, 241
ImageNtHeaderO, 227
IMAGE_NT_HEADERS, 227, 229-230
IMAGE_REL_BASED_ABSOLUTE, 245
IMAGE_REL_BASED_HIGHLOW, 245
IMAGE_RESOURCE_DIRECTORY_ENTRY, 243
ImageRvaToVaO, 226-227
IMAGE_SCN_CNT_CODE, 248
IMAGE_SCN_CNT_INITIALIZED_DATA, 248
IMAGE_SCN_CNT_UNINITIALIZED_DATA, 248
IMAGE_SCN_LNK_REMOVE, 248
IMAGE_SCN_MEM_DISCARDABLE, 248
IMAGE_SCN_MEM_EXECUTE, 249
IMAGE_SCN_MEM_NOT_CACHED, 248
IMAGE_SCN_MEM_NOT_PAGED, 248
IMAGE_SCN_MEM_READ, 249
IMAGE_SCN_MEM_SHARED, 248
IMAGE_SCN_MEM_WRITE, 249
IMAGE_SNAP_BY_ORDINAL, 239
IMAGE_SUBSYSTEM_NATIVE, 235
IMAGE_SUBSYSTEM_OS2_CUI, 235
IMAGE_SUBSYSTEM_POSIX_CUI, 235
IMAGE_SUBSYSTEM_WINDOWS_CUI, 235

IMAGE_SUBSYSTEM_WINDOWS_GUI, 235
 implicitly linked functions, 28
 integral subsystems, 144
 Intel 80286 microprocessor, 34
 Intel 80386 microprocessor
 address translation mechanism, 37
 addressing, 35-38
 page address translation, 37
 switching memory context, 78-79
 Windows 3.1 usage, 34
 Interrupt Descriptor Table (IDT), 117, 190
 Interrupt Descriptor Table Register (IDTR), 189, 190
 interrupt gates, 190
 structure, 202
 interrupt handlers, 125-126
 Interrupt Service Routines (ISRs), 109, 189
 Interrupt Vector Tables (IVTs), 189
InterruptGate_t, 202-203
 interrupts
 32 bit application execution, 201-202
 adding to kernel, 202-207
 defined, 189
 hooking, 191-198
 INTOX2B, 185-186
 INTOX2C, 185-186
 INT Ox2E, 186
 INT 2Ah, 191
 INT 2Bh, 191
 INT 2Ch, 186, 191
 INT 2Dh, 191
 INT2Eh, 117, 118, 191, 192-198, 220
 INT 2I, 190
 protected mode processing, 190
 unused interrupts, 202
 V86 mode processing, 190
IoConnectInterrupt(), 192
IoCreateDevice, 19
IoCreateSymbolicLink, 19
IRP_MJ_CLOSE, 21
IRP_MJ_CREATE, 21
IRP_MJ_DEVICE_CONTROL, 21
 ISRs (Interrupt Service Routines), 109, 189
 IVTs (Interrupt Vector Tables), 189

K

KeAddSystemServiceTable, 126, 129, 141-142
KeAttachProcess(), 79-80
KeDetachProcessO, 79-80
Ke1386AllocateGdtSelectors(), 101-105, 208
 kernel
 address space sharing, 40
 assembly language patterns, 99-100
 customizing, 123
 debugging flags, 91-97
 extending using device drivers, 141
 interrupts, adding, 202-207
 introduced, 10
 system services, adding, 128-140
 kernel level hooking, 113
 data structures involved, 116-117
 kernel-mode device drivers, 30
KERNEL32.DBG, 86
KERNEL32.DLL, 9, 15, 83, 111
 service ID mappings, 124, 145
 toolhelp functions, 28
KeServiceDescriptorTableO, 118, 126, 129
KeServiceDescriptorTableShadow, 126, 127-128, 142
 address, 129
KiBBTEndUnexpectedRange, 127-128
KiKiEndUnexpectedRange, 126-127
JcErrorMode, 127-128
KiSetLowWaitHighThreadO, 176-177, 186
KiSystemServiceO, 126
KRNL386, 24

L

language support, 6
LastRvaSection, 226-227, 228
 LDTs (Local Descriptor Tables), 36
 Least Significant Bit (LSB), 42
 Length (NtConnectPort field), 149
 linkers, 224
 version, 232
 Links (MapAndLoad parameter), 228
LoadDllAndInitializeVirtualAddressesO, 65, 68
LoadedImage, 228, 229
LOADEDIMAGE, 227
LoaderFlags, 235

- Local Descriptor Tables (LDTs), 36
 - local procedure calls (LPCs)
 - client-server communication channel, 152-153
 - client-subsystem communication, 145-147
 - connecting with server, 148-150
 - introduced, 14
 - long message communication, 147-148
 - port objects, communication via, 146-147
 - Quick LPC, 145, 175-186
 - receiving client requests, 150-152
 - replying to client requests, 150-152
 - shared memory, 147-148
 - shared memory sample program, 166-175
 - short message communication, 146-147
 - short message sample program, 156-166
 - types, 146
 - Local Security Authority (LSA), 144
 - LPC_CLIENT_DIED, 151, 155, 162
 - LPC_CONNECTION_REQUEST, 151, 162, 167, 171
 - LPC_DATAGRAM, 151, 154, 162
 - LPC_HANDLE_CLOSED, 185
 - LPC_PORT_CLOSED, 151, 155, 162
 - LpcptTraceMessages, 91-92
 - LPC_REPLY, 151
 - LPC_REQUEST, 151, 162, 171
 - LPCs. *See* local procedure calls (LPCs)
 - LPCSECTIONINFO, 149
 - LPCSECTIONMAPINFO, 169
 - LSA (Local Security Authority), 144
 - lsass.exe, 144
- ## M
- MACH operating system, 7, 143-144
 - Machine (PE file header field), 230
 - Magic (PE header field), 232
 - main(), 43, 72, 165
 - maintainability, 4-5
 - MajorFunction field, 22
 - MajorImageVersion, 234
 - MajorLinkerVersion, 232
 - MajorOperatingSystemVersion, 234
 - MajorSubsystemVersion, 234
 - MajorVersion
 - PE debug directory field, 245
 - PE export directory field, 237
 - PE resource directory field, 243
 - MapAndLoad(), 227-228
 - mapInfo, 149
 - MappedAddress, 228
 - MaxConnectInfoLength, 148, 161
 - MaxDataLength, 148, 161
 - MAX_VAD_ENTRIES, 53
 - memory management
 - access permissions, 37-38
 - address translation mechanism, 37
 - allocation/deallocation, 11
 - callgate, 40-41, 43-44
 - context switching, 78-82
 - dirty pages, 49, 66
 - DLL loading, 62-63
 - flat address spaces, 34-35, 38
 - local procedure call shared memory, 147-148
 - microprocessors, relation to, 33-34
 - page attributes, retrieving, 66
 - page frame databases (PFDs), 49-51
 - page table directories, 37, 38, 40, 47
 - page table entries (PTEs), 39, 73-74
 - paging, 37, 47-48, 62
 - Portable Executable (PE) format, 225-226
 - process environment blocks (PEBs), 51, 53
 - process isolation, 35, 38-39
 - running two program instances, 63
 - segment descriptors, 36
 - segment selectors, 36
 - segment tables, 36
 - segmented model, 33-34
 - shared memory sample program, 166-175
 - shared pages, 35, 39-48, 49, 62, 69-70
 - Thread Environment Blocks (TEBs), 53
 - VAD tree node structure, 53
 - virtual address descriptors (VADs), 51-61
 - virtual memory management, 49-51
 - Virtual Memory Manager, 13-14
 - Windows 95/98/NT differences, 82-83
 - MessageData, 151
 - MessageType, 151
 - microprocessors. *See also* 80286 microprocessor; 80386 microprocessor
 - relation to memory management, 33-34
 - Microsoft linker, 224
 - MinorImageVersion, 234
 - MinorLinkerVersion, 232
 - MinorOperatingSystemVersion, 234
 - MinorSubsystemVersion, 234
 - MmVersion
 - PE debug directory field, 245
 - PE export directory field, 237
 - PE resource directory field, 243
 - ML.EXE, 19
 - MmBadPageListHead, 50
 - MmCreateProcessAddressSpaceJ, 48
 - MmDebug, 92-93
 - MmFreePageListHead, 50
 - MmModifiedNoWritePageListHead, 50
 - MmModifiedPageListHead, 50
 - MmStandbyPageListHead, 50
 - MmZeroedPageListHead, 50
 - ModuleName, 228
 - Most Significant Bit (MSB), 42
 - MSPDB60.DLL, 86
 - multiprocessing
 - asymmetric, 6
 - features of NT, 5-6
 - symmetric, 6
 - multiprogramming, 6-7
 - multitasking
 - Windows 95/NT differences, 28-29
 - MYDRIVERENTRY macro, 20-21
- ## N
- Name
 - PE export directory field, 237
 - PE import directory field, 240
 - PE section table member, 247
 - native API, 145
 - network driver interface specification (NDIS) standard, 13
 - NewImageBase, 234
 - NewImageSize, 234
 - .NMS files, 86
 - NonFirstInstanceO, 70-71

- NonRelocatableFunctionO, 71
 - normal operation mode, 9
 - NT 3.51
 - GDI32.DLL functionality, 25
 - `_KiEndUnexpectedRange`, 126-127
 - NT 4.0 compared, 142
 - Quick LPC use, 175
 - service ID mapping, 124
 - system services provision, 111
 - Telnet capabilities, 6-7
 - user interface, 143
 - USER32.DLL functionality, 25
 - Windows 2000 compared, 142
 - NT 4.0
 - GDI32.DLL functionality, 25, 125
 - INT2Chuse, 186
 - INTOx2Buse, 185-186
 - JCiErrormode, 127-128
 - service ID mapping, 124-125
 - system services provision, 111
 - Telnet capabilities, 6-7
 - user interface, 143
 - USER32.DLL functionality, 25, 125
 - WIN32K.SYS, 111
 - NtExecutive, 110
 - NtAcceptConnectPortO, 146, 152-153
 - usage example, 159
 - NtAllocateVirtualMemoryO, 98, 100
 - NtCompleteConnectPortfJ, 153
 - NtConnectPortO, 146, 148-149
 - usage example, 174
 - NtCreateFileO, 118-122
 - NtCreatePortO, 146, 148
 - usage example, 161
 - NTDLLDBG, 86
 - NTDLL.DLL, 15, 25, 26, 94, 116, 202
 - NtGlobalFlag, 94-95
 - NtHeaders, 226
 - NtImpersonateClientOfPortO, 156
 - NTKRNLMP.EXE, 9
 - NtListenPortO, 146, 153-154
 - NtLockVirtualMemoryO, 100
 - `NtOpenThread()`, 182
 - NTOSKRNLDBG, 86
 - NTOSKRNL.EXE
 - export dumps, 101
 - selector handling, 208
 - System Service Dispatch Table (SSDT) creation, 117
 - System Service Dispatch Table (SSDT) locating, 118-122
 - system services counters, 192
 - system services provided by, 9, 111
 - NtQueryMutantQ, 98-99
 - NtRaiseHardErrorO, 155
 - NtRegisterThreadTerminatePortO, 155
 - NtReplyPortO, 155
 - NtReplyWaitReceivefJ, 146
 - usage example, 171
 - NtReplyWaitReceivePortO, 146, 150
 - message types, 151-152
 - usage example, 157, 162, 171
 - NtRequestPortO, 146, 154
 - NtRequestWaitReply(), 165
 - NtRequestWaitReplyPortO, 146, 153
 - NtSetHighWaitLowEventPairO, 176
 - NtSetLowWaitHighEventPairO, 176
 - `NtSetLowWaitHighThread()`, 186
 - service ID, 186
 - NTVTJM subsystem, 8
 - NumberOfFunctions, 237
 - NumberOfEntries, 243
 - NumberOfLinenumbers, 248
 - NumberOfNamedEntries, 243
 - NumberOfNames, 237
 - NumberOfRelocations, 248
 - NumberOfRvaAndSizes, 235
 - NumberOfSections, 228, 230
 - NumberOfServices, 118
 - NumberOfSymbols, 230
- O**
- ObDebugFlags, 94
 - Object Manager, 10-11
 - ObjectAttributes, 148
 - OBJECT_ATTRIBUTES, 166
 - ObpShowAllocAndFree, 91
 - OffsetToData, 244
 - OldImageBase, 234
 - OldImageSize, 234
 - OriginalFirstThunk, 239
 - OS/2 subsystem, 8
- P**
- Page Directory Base Register (PDBR), 37
 - page frame databases (PFDs), 49-51
 - page table directories, 37, 38, 40, 47, 100
 - page table entries (PTE), 39, 73-74, 100
 - PageDirectory, 41
 - paging, 47-48, 62
 - address translation mechanism, 3i
 - callgates accessing paged data, 220-221
 - dirty pages, 49, 66
 - page attributes, retrieving, 66
 - page table linear addressing, 47-48
 - parameters passed to undocumented functions, 97-99
 - ParamTable, 117
 - ParamTableBase, 118
 - .PDB files, 86
 - PDBR (Page Directory Base Register), 37
 - PE format. See Portable Executable (PE) format
 - PEBs (process environment blocks), 51, 53, 99
 - performance data
 - obtaining via system services hooking, 112
 - PFDs (page frame databases), 49-51
 - pLpcMessageIn, 153
 - PointerToLinenumbers, 248
 - PointerToRawData, 245, 248
 - PointerToRelocations, 248
 - PointerToSymbolTable, 230
 - portability features, 3
 - Portable Executable (PE) format
 - binding, 240-241
 - COFF style header, 230-232
 - data directory, 225
 - data directory indices, 237-239
 - debug directory, 245
 - DOS header/executable stub, 224, 229
 - dynamic linking, 240
 - export directory, 237-239
 - file structure, 224-225
 - header section, 224-225
 - import directory, 239-240
 - introduced, 223
 - loading PE files, 249
 - memory management, 225-227
 - optional header, 232
 - Relative Virtual Address (RVA) use, 225-226
 - relocation block format, 244
 - relocation table, 244-245
 - resource directory, 242-244
 - section alignment requirements, 225
 - section table, 225, 247-249
 - thread local storage, 246-247
 - unmapping, 229

- PortName, 149
 - ports
 - client-subsystem communication, 145-147
 - connecting, 148-150
 - creating, 148
 - handle pointers, 152
 - introduced, 146-147
 - receiving client requests, 150-152
 - related functions, 148-156
 - replying to client requests, 150-152
 - POSIX subsystem, 8, 63, 115
 - privilege levels, communication across, 207-209
 - privileged code, executing using callgates, 207-209
 - privileged operation mode, 9, 15
 - process environment blocks (PEBs), 51, 53, 99
 - Process Manager, 14
 - process startup
 - Windows 95/NT differences, 27-28
 - Process32First, 28
 - Process32Next, 28
 - ProcessConnectionRequest()**, 159, 169
 - ProcessLpcRequest()**, 160
 - ProcessMessageData()**, 159, 160, 167, 171
 - protected mode interrupt processing, 190
 - PROTPTE data structure, 48
 - PSAPI.DLL, 28
 - PtConvertToGuiThread, 127
 - pseudo device drivers, 19-20, 141
 - PspW2ProcessCallout, 142
 - PspW32Process, 127
 - PTEs (page table entries), 39, 73-74, 100
- Q**
- Quick LPC, 145, 175-176
 - communication steps, 177-178
 - program sample, 178-186
 - Win32 subsystem use, 176-177
- R**
- read ahead, 12
 - Readonly (MapAndLoad parameter), 228
 - ReadProcessMemoryO, 29
 - real mode interrupt processing, 189
 - ReBaseImageO, 233-236
 - reference count maintenance, 11
 - RegRestoreKey, 30
 - RegSaveKey, 30
 - Relative Virtual Addresses (RVAs)
 - export-functions arrays, 237-238
 - memory-mapped file address computing, 226
 - parameters, 226-227
 - Portable Executable (PE) file use, 225-226
 - relocating DLLs, 62, 78
 - remote login support, 6-7
 - remote procedure calls (RPCs), 145-146
 - Requested Privilege Level (RPL), 36
 - reverse engineering
 - breakout point, 97-98
 - calling convention basics, 87-88
 - compiler code generation patterns, 89-90
 - debugging flag use, 90-97
 - error handling code, 98
 - parameters passed to
 - undocumented functions, 97-99
 - SoftICE use, 85-86
 - undocumented function
 - deciphering example, 101-105
 - validation code checking, 99
 - RINGO.ASM, 43, 58
 - rings, 207-208
 - RPCs (remote procedure calls), 145-146
 - RPL (Requested Privilege Level), 36
 - RVA (ImageRvaToVa parameter), 226
- S**
- SAM (Security Accounts Manager), 144
 - SDK (Win32 Software Development Kit), 18
 - SectionAlignment, 234
 - Sections (MapAndLoad parameter), 228
 - security
 - overview, 5
 - Windows 95/98/NT differences, 30
 - Security Accounts Manager (SAM), 144
 - Security Reference Monitor, 13
 - segment descriptors, 36
 - segment selectors, 36
 - segment tables, 36
 - SelectorArray, 208
 - selectors, 208-209
 - SepDumpSD, 95-96
 - serverO, 171, 182
 - server thread, 176-178
 - Service Descriptor Table, 118, 126
 - shadow table, 126, 127-128
 - ServiceCounterTable, 118
 - ServiceTableBase, 118
 - Session Manager, 144
 - SetDataStructureOffsetsO, 56
 - SETENV.BAT, 18
 - shadow table, 126, 127-128, 142
 - address, 129
 - SharedSectionSize, 167
 - Size (PE resource directory entry), 244
 - SizeOfCode, 232
 - SizeOfData, 245
 - SizeOfHeaders, 235
 - SizeOfHeapCommit, 235
 - SizeOfHeapReserve, 235
 - SizeOfImage, 228, 234
 - SizeOfInitializedData, 232
 - SizeOfOptionalHeader, 230
 - SizeOfRawData, 248
 - SizeOfStackCommit, 235
 - SizeOfStackReserve, 235
 - SizeOfUninitializedData, 232
 - SizeOfZeroFill, 247
 - smss.exe, 144
 - SoftICE, 85-86
 - software interrupts. *See* interrupts
 - source compatibility, 4
 - SSDT. *See* System Service Dispatch Table (SSDT)
 - SSPT. *See* System Service Parameter Table (SSPT)
 - standard API calls, 27
 - StartAddressOfRawData, 246
 - STATUS_NO_EVENT_PAIR, 186
 - StatusRoutine, 242
 - stdcall, 87-88
 - Subsystem (PE header value), 235
 - subsystems
 - architecture, 7-9
 - environment subsystems, 144-145
 - extensibility, 3-4
 - integral subsystems, 144
 - introduced, 143-144
 - port creation, 146
 - types, 144
 - supervisor-mode, 66
 - SymbolPath, 233, 241

system call interface, 15

System Service Dispatch Table (SSDT)

- creation by NTOSKRNL, 117
- expanding, 123-124, 128-129
- locating in NTOSKRNL, 118-122
- service ID mappings, 124-125
- service ID validity checking, 126

System Service Parameter Table (SSPT)

- expanding, 128-129
- introduced, 117

system services

- adding, 128-140
- calling using INT 2Eh, 192-198
- defined, 109
- DOS system services, 109-110
- ID mappings, 124-125
- implementation, 126-127
- interrupt handlers, 125-126
- modifying, 111-112
- Windows NT, 115-116
- Windows 3x/95/98 system services, 110

system services, hooking

- DOS hooking, 113
- example using `NtCreateFile()`, 118-122
- hacking, relation to, 112
- kernel data structures involved, 116-117
- kernel level hooking, 113
- need for, 111-112
- user level hooking, 113
- Windows 95/98, 114
- Windows 3.x, 113-114

T

Task State Segments (TSSs), 202

Telnet support, 6-7

Thread Environment Blocks (TEBs), 53, 99, 126, 177

Thread Local Storage (TLS), 246-247

`ThreadInformationClass`, 182

thunking, 29, 239, 240

TimeDateStamp

- PE debug directory field, 245
- PE export directory field, 237
- PE header field, 230
- PE import directory field, 239
- PE resource directory field, 243

TimeStamp, 234

TLS (Thread Local Storage), 246-247

TokenGlobalFlag, 96

toolhelp functions, 28

TSSs (Task State Segments), 202

Type (PE debug directory field), 245

U

Unicode character set, 157

UNICODE_STRING, 166

UNIX, 143-144

UnMapAndLoadO, 229

user interface, 143

user mode, 66

USER32.DBG, 86

USER32.DLL

- functionality in Windows 2000/1> 125
- introduced, 9
- service descriptor shadow table usage, 127-128
- system ID mappings, 124
- thunking, 29
- Win32 subsystem interaction, 25

USER.EXE, 24

V

V86 mode interrupt processing, 190

VadInfoArray, 55

VadInfoArrayIndex, 53, 55

VadRoot, 51

VADs. *See* virtual address descriptor; (VADs)

VadTreeDisplayO, 56

VadTreeRoot, 53

VadTreeWalk(), 55

validation code, checking, 99

virtual address descriptors (VADs), 51-61

- global variables, 53
- tree node structure, 53

virtual device drivers (VXD), 30

virtual memory management, 49-51

Virtual Memory Manager, 13-14

`VirtualAddress`, 247

`VirtualAlloc()`, 98

`VirtualAllocEx()`, 79

`VirtualLock()`, 65

`VirtualProtectO`, 62

`VirtualSize`, 247

`VirtualUnlockO`, 66

VXD (virtual device drivers), 30

W

Win16Mutex, 28-29

Win32 API

- Windows 95 implementation, 24
- Windows 95/NT differences, 26-31

Win32 subsystem, 7-9, 15

- INTOX2Buse, 186
- INT Ox2C use, 186
- Quick LPC use, 176-177

WIN32K.DBG, 86

WIN32K.SYS, 15, 25, 111, 126, 141

Win32s, 23

Win32VersionValue, 234

Windows 3.x

- Intel 80386 microprocessor use, 34
- system services, 110
- system services hooking, 113-114

Windows 95/98

- system services, 110
- system services hooking, 114
- user interface, 143
- Win32.API implementation, 24

Windows 95/98/NT differences

- device drivers, 29-30
- memory management, 82-83
- multitasking, 28-29
- process startup, 27-28
- toolhelp functions, 28
- Win32.API implementation, 26-31

Windows 2000

- GDI32.DLL functionality, 125
- KeAddSystemServiceTable, 141
- KiBBTEndUnexpectedRange, 127-128
- NT compared, 142
- service ID mapping, 125
- shadow table, 129
- Telnet capabilities, 6-7
- user interface, 143
- USER32.DLL functionality, 125

Windows CE, 23

WINNT.H file, 229

WORDS, 208

WOW subsystem, 8, 25

write requests

- delayed write, 12

WriteProcessMemoryO, 79

Z

ZwSetInformationThreadO, 182