

-
- .NET
 - advantages, **Vol 4:** 9
 - class library, **Vol 4:** 12
 - framework, **Vol 4:** 10–12
 - interoperate,with, **Vol 4:** 24–27, 24
 - overview, **Vol 4:** 7–9, 7
- .NET framework
 - redevelopment, **Vol 4:** 22–24, 22
- .NET,from SFU, **Vol 2:** 107

6

- 64-bit Programming, **Vol 3:** 8–11, 8, 37–41
 - scalable data types, **Vol 3:** 38
- 64-bit Windows
 - overview, **Vol 3:** 7–11, 7

A

- Access Control Entries (ACE), **Vol 3:** 19
- Access Control Lists (ACL), **Vol 3:** 18, 123, 237, 242, 243
- access token, **Vol 3:** 19
- active directory, **Vol 5:** 8
- Active Template Library (ATL), **Vol 3:** 31
- Active X Template Library (ATL), **Vol 3:** 31, 36, 37, 150, 151, 153, 156, 158, 160, 162, 163, 169, 173, 180, 182, 201, 202
- ActivePerl, **Vol 2:** 99
- address space, **Vol 3:** 228
- ADO.NET
 - connections, **Vol 4:** 29
 - dataadapter, **Vol 4:** 31
 - datareaders, **Vol 4:** 29
 - introduction, **Vol 4:** 29–31, 29
- AIX, **Vol 1:** 8
- alert view, **Vol 5:** 25
- application
 - monitoring process, **Vol 5:** 24
- application architectures, **Vol 1:** 20–22
- application domains
 - advantages, **Vol 4:** 54
 - comparison, UNIX process, **Vol 4:** 53
 - comparison,threads, **Vol 4:** 54
 - introduction, **Vol 4:** 16, 53–55, 53
 - programming, **Vol 4:** 54–55
 - programming,example, **Vol 4:** 55
- Application Programming Interface(API),
 - Vol 1:** 36, 37
 - Vol 5:** 26
- application security
 - role based, **Vol 4:** 15
- application servers, **Vol 5:** 6

- application types
 - device drivers, **Vol 1:** 22
 - distributed, **Vol 1:** 21
 - business tier, **Vol 1:** 21
 - data storage tier, **Vol 1:** 21
 - view tier, **Vol 1:** 21
 - graphics intensive, **Vol 1:** 22
 - web, **Vol 1:** 21
- Application Window, **Vol 4:** 138
- applications
 - rich client,controls, **Vol 4:** 28
 - rich client,events, **Vol 4:** 28
 - rich client,introduction, **Vol 4:** 28
 - web,introduction, **Vol 4:** 28–29, 28
- architectural differences,UNIX and Windows, **Vol 3:** 11–25, 11
- archiving tools, **Vol 2:** 102
- assemblies, **Vol 4:** 18, 201–202, 201, 202
- assembly
 - introduction, **Vol 4:** 201
 - private, **Vol 4:** 202
 - shared, **Vol 4:** 202
- assessing, **Vol 3:** 221
- auditing, **Vol 2:** 125
 - Vol 3:** 253
 - Vol 4:** 210
- authentication, **Vol 2:** 125
 - Vol 3:** 252
 - Vol 4:** 210
 - Vol 5:** 21
- authorization, **Vol 2:** 125
 - Vol 3:** 252
 - Vol 4:** 210
 - Vol 5:** 21
- automount, **Vol 3:** 16

B

- BackOffice components, **Vol 5:** 25, 26
- basetsd.h., **Vol 3:** 37
- Berkeley, **Vol 5:** 22
- Berkeley r commands, **Vol 2:** 118
- Berkeley remote shell commands, **Vol 2:** 98
- bitmap, **Vol 3:** 211
- Bourne shell (sh), **Vol 3:** 141
- BSD 4.3, **Vol 2:** 13
- budget and purchasing plan, **Vol 1:** 48
- building application, **Vol 2:** 29
- building the application, **Vol 3:** 241
- business goals, **Vol 1:** 27–29

C

- C function declarations, **Vol 3:** 218
- C shell, **Vol 2:** 99

C shell (csh), Vol 3: 141
 C++ interoperability (IJW)
 advantages, Vol 4: 26
 calling Win32 API, example, Vol 4: 47
 disadvantages, Vol 4: 26
 introduction, Vol 4: 26
 marshalling, Vol 4: 48,
 performance consideration, Vol 4: 27
 callback, Vol 3: 208
 capturing keyboard events, Vol 3: 168–172,
 168
 capturing mouse events, Vol 3: 165–167, 165
 CGI script migration, Vol 3: 239
 Citrix, Vol 3: 24
 Citrix MetaFrame, Vol 5: 23
 client window, Vol 4: 174
 CLR
 advantages, Vol 4: 11
 cluster configuration, Vol 2: 126
 Vol 3: 253
 Vol 4: 211
 code component testing, Vol 2: 123
 Vol 3: 250
 Vol 4: 208
 color
 management, Vol 4: 166
 color management, Vol 3: 177–178, 177
 COM
 Interix, encapsulating, Vol 2: 110
 COM interop services
 introduction, Vol 4: 26
 COM Interop services
 advantages, Vol 4: 27
 Command Debugger (CMD), Vol 3: 33
 command line
 UNIX development servers, Vol 5: 21
 command line, SFU 3.5 Vol 2: 10
 command shell, Vol 3: 238
 Command window, Vol 3: 209
 Command Window, Vol 4: 174
 command-line access tools, Vol 5: 21
 command-line shell, Vol 3: 141
 command-line tools
 rcp, Vol 5: 21
 rexec, Vol 5: 21
 rsh, Vol 5: 21
 ssh, Vol 5: 21
 telnet, Vol 5: 21
 Common Internet File System (CIFS), Vol 1:
 15, 70
 Common Internet File System(CIFS), Vol 3:
 16
 Common Language Runtime(CLR), Vol 5: 26
 communicating with a control, Vol 3: 161
 communications plan, Vol 1: 13, 17, 44, 46,
 47, 51
 compatibility layer, Vol 3: 218
 Component based development, Vol 3: 144–
 145, 144
 Component Object Model, Vol 3: 144
 Component Object Model (COM), Vol 5: 7,
 26, 27, 28
 component, COM, middleware, Vol 1: 18
 computer information, Vol 2: 93
 Vol 3: 229–231, 229
 Vol 4: 195–196, 195
 hardware platform name, Vol 3: 229
 host name, Vol 3: 229
 network name, Vol 3: 229
 operating system name, Vol 3: 229
 operating system release level, Vol 3: 229
 operating system version number, Vol 3:
 229
 configuration, Vol 2: 104
 Vol 3: 241
 inetd, Vol 2: 105
 logon/logoff scripts, Vol 2: 105
 configuration file, Vol 4: 175, 202–203, 202
 application, Vol 4: 203
 machine, Vol 4: 203
 security, Vol 4: 203
 connectionless protocols, Vol 3: 139
 connectivity
 command-line, Vol 5: 18, 19, 21
 terminal emulation, Vol 5: 18, 19, 21
 Windows to UNIX, Vol 5: 21
 console mode, Vol 3: 209
 Console Mode, Vol 4: 174
 control, Vol 3: 19, 24, 108, 124, 149, 151,
 157, 158, 160–164, 160, 161, 202, 211,
 212, 241
 communicating, with, Vol 4: 148–149, 148
 creating, Vol 4: 145–148, 145
 identifying, Vol 4: 148
 CPrintDialog, Vol 3: 202, 203
 CreateDialog, Vol 3: 156–160, 156, 157
 CreateSolidBrush, Vol 3: 183, 193
 CreateThread, Vol 3: 52, 59–62
 CreateWindow, Vol 3: 49, 149, 152, 154–
 156, 155, 160, 161, 164, 195
 creating a control, Vol 3: 160
 creating the functional specification for the
 solution, Vol 1: 43, 45, 47
 creating threads, Vol 3: 59
 critical section, Vol 3: 77
 C-Runtime (CRT) library, Vol 3: 31
 Curses, Vol 3: 24

D

daemons, Vol 2: 16–17, 17, 58, 73–78, 74
 Vol 3: 13, 121
 Vol 4: 4, 16, 18, 20, 109, 130–131, 130
 converting code, Vol 2: 78
 cron service, Vol 2: 75
 porting, Interix, Vol 2: 77–78, 77

- porting, Interix service, **Vol 2:** 77
- daemons and services, **Vol 1:** 13
- data model
 - UNIX/64, **Vol 3:** 8–11, 9
 - Win64, **Vol 3:** 8–11, 9
- data protection, **Vol 2:** 125
 - Vol 3:** 252
 - Vol 4:** 210
- data sharing, **Vol 5:** 23
- database
 - accessing, SFU 3.5 **Vol 2:** 96
 - connectivity, **Vol 2:** 95–98, 95
 - Vol 4:** 131
 - data source, **Vol 2:** 95
 - driver, **Vol 2:** 95
 - odbc, **Vol 2:** 95
 - unixODBC, **Vol 2:** 96
- database connectivity, **Vol 3:** 239–240, 239
 - ActiveX® Data Objects, **Vol 3:** 239
 - Microsoft Data Access Components, **Vol 3:** 239
 - OLE DB, **Vol 3:** 239
 - Open Database Connectivity, **Vol 3:** 240
- database testing, **Vol 2:** 124
 - Vol 3:** 252
 - Vol 4:** 209
- debugging application, **Vol 2:** 30
- debugging FORTRAN, **Vol 3:** 223
- DeleteObject, **Vol 3:** 165, 174
- DeletePen, **Vol 3:** 165
- demons, **Vol 3:** 142
- deploy operate phase
 - introduction, **Vol 5:** 5, 8, 11
 - tasks, output, **Vol 1:** 24–25
 - Vol 5:** 5–6
- deployment
 - packaging tools, **Vol 3:** 242
- deploying
 - archiving tools, **Vol 2:** 102–104, 102
 - code modification, **Vol 2:** 101
 - compression, **Vol 2:** 103
 - configuration, **Vol 2:** 104
 - Interix, **Vol 2:** 99, 118–121, 118
 - packaging tools, **Vol 2:** 102
 - pushing, desktop, **Vol 2:** 99
 - remote management, **Vol 2:** 99
 - remote, Interix, **Vol 2:** 119
 - scripts, **Vol 2:** 98
 - tools, **Vol 2:** 98
- deploying application, **Vol 4:** 201–206, 201
- deploying phase
 - creating a live environment, application deployment, **Vol 5:** 6
 - creating a live environment, application servers, **Vol 5:** 6
 - creating a live environment, interoperability, **Vol 5:** 6
 - creating a live environment, networked file systems, **Vol 5:** 6
- Deploying Phase
 - creating a live environment, **Vol 5:** 6
 - creating a live environment, interoperability, **Vol 5:** 6
- deploying Win32 applications, **Vol 3:** 247
 - knife-edge installation, **Vol 3:** 247
 - two-phase deployment, **Vol 3:** 247
- deploying, application, **Vol 2:** 98–102
- deployment, **Vol 2:** 92
 - Vol 3:** 241
 - Vol 4:** 7, 9, 11, 18, 20
 - Vol 5:** 7
 - configuration, **Vol 3:** 241
 - group policy objects, **Vol 3:** 246
 - installation, **Vol 3:** 242
 - Installation Rollback, **Vol 3:** 243
 - Install-on-Demand, **Vol 3:** 242
 - packaging tools, **Vol 3:** 242
 - Systems Management Server, **Vol 3:** 246
 - tools, **Vol 5:** 7
- Deployment, **Vol 5:** 7
- deployment, installation, **Vol 5:** 8
- deployment:, **Vol 3:** 241–247
- deposing
 - libraries, **Vol 2:** 104
- design, **Vol 3:** 221
- Desktop Window, **Vol 4:** 138, 176
- detachstate, **Vol 3:** 84
- developing phase, **Vol 2:** 20, 22
- deliverables, **Vol 3:** 27
 - end, **Vol 2:** 127
 - Vol 3:** 254
 - Vol 4:** 212
 - goals, **Vol 2:** 8, 20
 - Vol 3:** 27
 - introduction, **Vol 2:** 20
 - major tasks, **Vol 3:** 27
 - milestone, **Vol 2:** 127
 - Vol 3:** 254
 - Vol 4:** 212
 - proof of concept, **Vol 4:** 35
 - risk, **Vol 4:** 34–35
 - tasks, **Vol 2:** 20–21, 21
- developing Pphase, **Vol 2:** 20, 21, 22
- development environment, **Vol 1:** 53
 - Vol 3:** 30–37
 - data population, **Vol 1:** 68–71
- development tools, **Vol 3:** 220, 223–224
- Device Context (DC), **Vol 3:** 172–177, 172
- dialog boxes, **Vol 3:** 31, 149, 150, 156, 164, 211, 263
 - Vol 4:** 221
- Dialog Boxes, **Vol 4:** 138
- DialogBox, **Vol 3:** 156, 157
- differences

- architectural differences between UNIX and Windows, **Vol 3:** 11–25, 11
- file system differences between UNIX and Windows, **Vol 3:** 17
- POSIX and Windows environment
 - variables, **Vol 3:** 228
- UNIX/64 vs. Win64, **Vol 3:** 8
- user interface differences, **Vol 3:** 23–24, 23
- Win32 vs. Win64 API, **Vol 3:** 10–11
- directory
 - access.UNIX, **Vol 4:** 104–106
 - operations, **Vol 2:** 51–52, 51
 - working directory, **Vol 2:** 51
 - working,UNIX, **Vol 4:** 103–108
- directory operations, **Vol 3:** 107, 113–117
- Discretionary Access Control List (DACL), **Vol 3:** 19, 20, 124
- display
 - management, **Vol 4:** 166
- display management, **Vol 3:** 177–196, 177
- DLLs, **Vol 4:** 11, 12, 25, 45, 204
- DPI, **Vol 4:** 174
- drawing, **Vol 3:** 178–183
 - lines, **Vol 3:** 179
 - rectangles, **Vol 3:** 181
 - shapes, **Vol 4:** 166–170, 169–170, 169
 - two-dimensional lines, **Vol 4:** 167–168, 167

E

- EndDialog, **Vol 3:** 156, 157, 158
- enterprise services, **Vol 4:** 189
- environment
 - deployment, **Vol 2:** 92
 - live, **Vol 2:** 92
 - process, **Vol 2:** 92
 - temporary files, **Vol 2:** 93
 - variables, **Vol 2:** 92
- Environment
 - live, **Vol 5:** 6
- environment block, **Vol 3:** 228
- environment variables, **Vol 2:** 108
 - Vol 3:** 228–229, 228
 - Vol 4:** 193–195, 193
- error handler, **Vol 2:** 60
- error handling, **Vol 3:** 127
- event handling, **Vol 3:** 165–172, 165
- event log, **Vol 5:** 25
- event monitoring, **Vol 5:** 26
- event queue, **Vol 3:** 126
- events, **Vol 1:** 13
 - introduction, **Vol 4:** 109–110
 - keyboard, **Vol 4:** 151–153, 151
 - model, **Vol 4:** 110–112
 - mouse, **Vol 4:** 150
 - raise, **Vol 4:** 111
- Events, **Vol 3:** 21
- evolving application
 - migration strategies, **Vol 4:** 21–22, 22

- exception
 - handling,model, **Vol 4:** 118–121, 118
 - monitoring, **Vol 5:** 25
- exception class, **Vol 4:** 17, 118
- exception handling, **Vol 3:** 127
- exception monitoring, **Vol 5:** 25
- exceptions, **Vol 1:** 13
 - .NET, **Vol 4:** 118–121, 118
- Exceptions, **Vol 3:** 21
 - Vol 5:** 25
- Extensible Markup Language (XML), **Vol 1:** 21, 60, 65

F

- fifo
 - UNIX,example, **Vol 4:** 126–129, 126
- file, **Vol 1:** 1
 - Vol 2:** 15, 23, 46, 48
 - Vol 3:** 1, 107
 - Vol 4:** 1
- access mode, **Vol 4:** 101–102
- control, **Vol 2:** 48, 49
- create, **Vol 2:** 49, 50
- fcntl(), **Vol 3:** 110–113, 110
- file Nnames, **Vol 2:** 15
- Gdbm, **Vol 2:** 55
- group, **Vol 2:** 49
- mount entry, **Vol 2:** 54
- NTFS, **Vol 2:** 52
- open mode, **Vol 4:** 101
- operations, **Vol 2:** 52–54, 52
- path Nnames, **Vol 2:** 15, 29
- permissions,Interix, **Vol 2:** 49
- permissions,win32, **Vol 2:** 50
- security, **Vol 2:** 49
- systems,UNIX, **Vol 2:** 53

- file access, **Vol 3:** 107
- file control, **Vol 3:** 107
- file descriptor, **Vol 3:** 107
- file handling, **Vol 3:** 107
- file I/O differences, UNIX, **Vol 2:** 48
- file management, **Vol 2:** 11, 15–16, 15, 16, 47–57, 48, 51, 52, 53, 54, 64, 65, 95, 98, 108
- Vol 4:** 4, 16, 85, 96–108
- File management, **Vol 1:** 1
- Vol 2:** 15, 23, 46
- Vol 3:** 1
- Vol 4:** 1
- file system
- automount, **Vol 3:** 16
- Common Internet File System(CIFS), **Vol 3:** 16
- differences, **Vol 3:** 14–17, 14
- inode, **Vol 3:** 15
- interoperability, **Vol 1:** 15
- network file system, **Vol 3:** 16
- networked file systems, **Vol 1:** 14–15, 14

Server Message Block(SMB), **Vol 3**: 16
file system interaction, **Vol 3**: 236
file systems, networked file systems
 path names, **Vol 1**: 15
file systems, networked file systems, file
 names, **Vol 1**: 14
File Transfer Protocol (FTP), **Vol 5**: 23
fonts, **Vol 3**: 184
 creating, **Vol 4**: 158
 example, **Vol 4**: 159
 using, **Vol 4**: 157–160, 157
Fortran, **Vol 1**: 22
 using mixed languages, **Vol 3**: 216–220,
 216
 using third party libraries, **Vol 3**: 216–220,
 216
FORTRAN interface, **Vol 3**: 217
FORTRAN modules, **Vol 3**: 218
FreeTDS, **Vol 2**: 97
full duplex pipes, **Vol 3**: 23, 135

G

Garbage Collector (GC), **Vol 4**: 86–87, 86,
87, 90
Gdbm, **Vol 2**: 54–57, 55
GDI+, **Vol 3**: 150, 151, 172–183, 207
 Vol 4: 134, 172–183
GetOpenFileName, **Vol 3**: 164
getrlimit, **Vol 3**: 53
getrusage, **Vol 3**: 53
GetSysColor, **Vol 3**: 193
GID, **Vol 1**: 15
 Vol 3: 18, 123
globbing, **Vol 3**: 238
Graphics Class object, **Vol 4**: 175
Graphics Context, **Vol 4**: 166, 175
Graphics Device Interface (GDI), **Vol 3**: 172
 Vol 4: 154
GrayString, **Vol 3**: 193, 194
group policy objects, **Vol 3**: 246
group security, **Vol 3**: 121
GUI access tool, **Vol 5**: 21

H

half duplex pipes, **Vol 3**: 23, 135
handles, **Vol 1**: 13, 62
 Vol 3: 125–127
 socket handles, **Vol 3**: 21, 125
header files, **Vol 2**: 24, 27, 28, 29, 82, 102,
104
heap, **Vol 4**: 85
heap memory, **Vol 3**: 101
Hummingbird, **Vol 5**: 23
 Exceed, **Vol 5**: 23
HyperTerminal, **Vol 5**: 22
Hypertext Transfer Protocol (HTTP), **Vol 1**:
21

I

I/O Completion Ports, **Vol 3**: 98
identifying a control, **Vol 3**: 161
imaging, **Vol 3**: 205–208, 205
 UNIX, **Vol 3**: 205
 Windows, **Vol 3**: 206
Impersonation, **Vol 3**: 20, 124
include files, **Vol 3**: 164
inetd, **Vol 2**: 105
infrastructure services, **Vol 2**: 16, 23, 58
 Vol 3: 1
 Vol 4: 1
Infrastructure services
 differences, **Vol 3**: 17
inode, **Vol 3**: 15, 126
input,output,error, **Vol 2**: 109
installation, **Vol 3**: 242–245, 242
Installation Rollback, **Vol 3**: 243
Install-on-Demand, **Vol 3**: 242
InstallShield, **Vol 3**: 244
integrating
 FORTRAN with POSIX, **Vol 3**: 220
integration plan, **Vol 1**: 49
integration testing, **Vol 2**: 124
 Vol 3: 251
 Vol 4: 209
intended audience, **Vol 3**: 2
 Vol 4: 2
Interix, **Vol 1**: 31
 Vol 5: 6, 7, 21, 22, 23, 24, 26, 27
 building application, **Vol 2**: 29–30, 29
 COM,encapsulating, **Vol 2**: 110
 debugging application, **Vol 2**: 30–31, 30
 development environment, **Vol 2**: 20, 24,
 29, 30, 32, 40, 48, 82, 98, 107, 110
 Vol 3: 43
 Vol 4: 38
 differences,UNIX, **Vol 2**: 10–18, 10–18
 environment, **Vol 2**: 20, 24, 29, 30, 32, 40,
 48, 49–50, 82, 98, 107, 110
 Vol 3: 43
 Vol 4: 38
 functions, **Vol 2**: 86
 header files, **Vol 2**: 24, 27, 28, 29
 introduction, **Vol 2**: 8, 9
 MSI, **Vol 2**: 101
 remote,deployment, **Vol 2**: 119
 SDK, **Vol 2**: 17, 18, 29, 30, 68, 82, 83, 84,
 90, 106
 sub-system, **Vol 2**: 9
Interix,functions
 BSD string and bit, **Vol 2**: 89
 command line api, **Vol 2**: 88
 math, **Vol 2**: 86–87, 86
 regular expressions, **Vol 2**: 87
 shell api, **Vol 2**: 88
 string manipulation, **Vol 2**: 89–90, 89
Interlocked Exchange, **Vol 3**: 69

interop services, **Vol 5**: 27, 28
Interoperability mechanisms
 wrapping unmanaged C++ classes, **Vol 4**:
 25
 wrapping unmanaged C++
 classes, example, **Vol 4**: 41–44, 42
interprocess communication, **Vol 2**: 63–68,
64
 Vol 4: 4, 16, 17, 20, 57, 109, 124–130, 124,
 129
 message queues, **Vol 2**: 68
 shared memory, **Vol 2**: 47
 sockets, **Vol 4**: 19, 121
 unnamed pipes, **Vol 2**: 64
Interprocess communication, **Vol 3**: 135–139
 Pipes, **Vol 3**: 22, 23, 135–138, 135
 Shared Memory, **Vol 3**: 22, 138
 Sockets, **Vol 3**: 21
interprocess communication (IPC), **Vol 4**: 124
Interprocess communication (IPC), **Vol 3**: 21–
23, 21
inttypes.h, **Vol 3**: 37
Invoke .NET, from SFU, **Vol 2**: 107
Invoke win32/64, from SFU, **Vol 2**: 107
ioctl, **Vol 2**: 27, 48–49, 48, 49, 83, 84, 101
ioctl() Calls, **Vol 3**: 108–109, 108
ioctlsocket, **Vol 3**: 108
ipc
 shared memory, **Vol 2**: 47
IPC
 DLLs and shared libraries, **Vol 1**: 18
 mechanisms, **Vol 1**: 17
 UNIX versus Windows, operating system,
 Vol 1: 17
lpcm, **Vol 2**: 47, 65
lpcs, **Vol 2**: 14, 47, 65
IsDialogMessage, **Vol 3**: 163
Isolated Storage, **Vol 4**: 187
It Just Works (IJW), **Vol 5**: 26

J

Java Server Pages (JSP), **Vol 1**: 21
job objects, **Vol 3**: 53

K

kernel mode, **Vol 1**: 7
keyboard focus, **Vol 3**: 171
 Vol 4: 153–154, 153
knife-edge installation, **Vol 3**: 247
Korn, **Vol 2**: 99
Korn shell (ksh), **Vol 3**: 141

L

libraries, **Vol 2**: 104
 Vol 3: 164–165, 164
live environment, **Vol 2**: 92
LLP64, **Vol 3**: 8

logging system messages, **Vol 3**: 232–235
logon/logoff scripts, **Vol 2**: 105
low-level file access, **Vol 3**: 108
LP64, **Vol 3**: 8

M

maintenance, **Vol 1**: 30
managed heap, **Vol 4**: 85
management
 device, **Vol 4**: 150–156, 150
management testing, **Vol 2**: 126
 Vol 3: 253
 Vol 4: 211
managing multiple threads, **Vol 3**: 95
managing process, **Vol 3**: 57
Marshaling data, **Vol 4**: 44
master project schedule, **Vol 1**: 51
memory
 allocation, **Vol 4**: 85–86, 85
 functions, **Vol 2**: 46
 heap, **Vol 2**: 46
 Vol 3: 101
 management, **Vol 4**: 85–95, 85
 memory-mapped files, **Vol 3**: 105
 release, **Vol 4**: 86–93, 86
 shared, **Vol 2**: 47
 synchronize access, **Vol 2**: 47
 virtual address space, **Vol 3**: 11, 35, 101,
 138
memory management, **Vol 1**: 1
 Vol 2: 11, 14, 23, 46–47, 46
 Vol 3: 1, 14
 Vol 4: 1
 virtual memory, **Vol 3**: 14
memory mapped files, **Vol 4**: 129
memory-mapped files, **Vol 3**: 105, 106, 248
Message Box, **Vol 4**: 139
message queues, **Vol 1**: 17
Message Queues, **Vol 3**: 139
 Vol 4: 130
Microsoft Exchange, **Vol 5**: 25, 26
Microsoft Exchange Server, **Vol 5**: 26
Microsoft Foundation Classes (MFC), **Vol 3**:
 31, 36, 37, 150, 151, 197, 200, 201, 202,
 206
Microsoft Messaging Queue (MSMQ), **Vol 3**:
 22
Microsoft Solutions Framework, **Vol 2**: 20, 31,
 32
 Vol 3: 43
 Vol 4: 38
Microsoft Visual Studio IDE, **Vol 3**: 33
middleware, **Vol 3**: 143
migrating FORTRAN
 assessing, **Vol 3**: 221
 design, **Vol 3**: 221
 development tools, **Vol 3**: 220
 migration planning, **Vol 3**: 222

- mitigating risk, Vol 3: 221
- resources, Vol 3: 220
- scoping, Vol 3: 222
- sizing, Vol 3: 221
- validation, Vol 3: 221
- migrating FORTRAN code, Vol 3: 4, 215
- migrating scripts, Vol 2: 94–95, 94
Vol 3: 235–239, 235
 - device and network programming, Vol 3: 238
 - environment variables, Vol 3: 237
 - file system interaction, Vol 3: 236
 - process and thread execution, Vol 3: 238
 - signal handling, Vol 3: 238
 - user interface, Vol 3: 239
 - Windows registry, Vol 3: 237
- migration
 - alternatives, Vol 1: 32–34
 - build environment, to, Vol 1: 71–77
- migration planning, Vol 3: 222
- migration strategies
 - evolving application, Vol 4: 21–22, 22
 - static application, Vol 4: 21–22, 22
- migration using SFU 3.5, 32
 - network management, Vol 1: 32
 - optimizing investments, Vol 1: 32
 - reducing development time, Vol 1: 32
 - reusing scripts, Vol 1: 32
 - SDK, Vol 1: 32
- mitigating risk, Vol 3: 221
- MKS, Vol 1: 36
 - Vol 5: 23
 - Toolkit, Vol 1: 36
 - Vol 5: 23
- modal dialog box, Vol 3: 156
- mode
 - kernel, Vol 1: 7
 - protected, Vol 1: 7
 - supervisor, Vol 1: 7
 - user, Vol 1: 7
- modeless dialog box, Vol 3: 156
- Modeless Dialog Box, Vol 4: 139
- monitoring, Vol 2: 118
 - application, Vol 5: 24
 - exception, Vol 5: 25
 - perfmon.msc, Vol 5: 25
 - performance monitor, Vol 5: 25
 - system monitor, Vol 5: 25
- Motif, Vol 2: 84
 - Vol 3: 24
- mouse messages
 - client area, Vol 3: 165–167, 165, 166
- MSF, Vol 2: 20, 31, 32
 - Vol 3: 43
 - Vol 4: 38
- MSI, Vol 2: 101
- MSI, Interix rsh, Vol 2: 101

- multiprocessor considerations, Vol 2: 136
 - Vol 3: 262
 - Vol 4: 220
- multitasking, Vol 1: 8, 12
 - Vol 3: 11
- Mutex, Vol 3: 73

N

- named pipes, Vol 3: 23, 135
 - Vol 4: 126–129
- namespace, Vol 5: 27, 28
- native signals, Vol 3: 129
- NetBIOS, Vol 1: 15
- network
 - security testing, Vol 2: 125
 - Vol 3: 253
 - Vol 4: 210
- Network File System (NFS), Vol 1: 21, 68, 70
 - Vol 3: 16
- Network Information Service, Vol 2: 16, 70, 104
- network management, Vol 5: 25
- network security testing, Vol 2: 125
 - Vol 3: 253
 - Vol 4: 210
- network utilization, Vol 2: 136
 - Vol 3: 262
 - Vol 4: 220
- networking, Vol 3: 23, 139–140, 139
 - Vol 4: 121–124, 121
 - connectionless protocols, Vol 3: 139
 - host name, Vol 2: 69–70, 69, 70
 - Remote Procedure Call (RPC), Vol 2: 69
 - RPC, Vol 3: 139, 140
 - socket calls, Vol 2: 70
 - sockets, Vol 3: 23, 139
 - Sockets, Vol 2: 70
 - TCP/IP, Vol 3: 23, 139
 - Transport Level Interface (TLI), Vol 2: 72–73, 72
 - User Datagram Protocol (UDP), Vol 2: 72
 - Windows Open System Architecture (WOSA), Vol 3: 139
 - Winsock, Vol 3: 23, 139
 - X/Open Transport Interface (XTI), Vol 2: 72
- networking,sockets,winsock, Vol 1: 16
- New Technology File System, Vol 2: 52
- NIS, Vol 2: 16, 70, 104
- notification, Vol 5: 25
- nterix
 - header files, Vol 2: 24–29
- NTFS, Vol 2: 52

O

- Online Transaction Processing (OLTP), Vol 1: 19
 - Vol 3: 143

OpenGL, **Vol 1:** 21, 22, 46, 77
Vol 2: 84
Vol 4: 173
OpenGL Graphics Library Utility Kit (GLUT),
Vol 3: 212
operations, **Vol 5:** 21, 24–25
 exception monitoring, **Vol 5:** 25
Operations, **Vol 5:** 6
 exception monitoring, **Vol 5:** 25
output buffer, **Vol 3:** 126

P

P/Invoke
 advantages, **Vol 4:** 26
 disadvantages, **Vol 4:** 26
 introduction, **Vol 4:** 45–46, 45
 Win32 API, example, **Vol 4:** 47
packaging tools, **Vol 2:** 102
 Vol 3: 242–245, 242
pager notification, **Vol 5:** 26
patch files, **Vol 3:** 244
pathnames, **Vol 2:** 108
performance monitor, **Vol 5:** 25
Performance Monitor and Exchange
 Administrator, **Vol 5:** 26
performance tuning, **Vol 2:** 134
 Vol 3: 260
 Vol 4: 218
Perl, **Vol 2:** 97, 98
 Vol 3: 142
pilot deployment plan, **Vol 1:** 50
pipes, **Vol 3:** 23, 135
 back pressure, **Vol 3:** 138
 full duplex pipes, **Vol 3:** 23, 135
 half duplex pipes, **Vol 3:** 23, 135
 named pipes, **Vol 3:** 23, 135
 process pipes, **Vol 3:** 135
 unnamed, **Vol 2:** 64
 unnamed pipes, **Vol 3:** 23, 135
pixmap, **Vol 3:** 211
planning
 scope, content, **Vol 1:** 41
 solution, **Vol 1:** 41
Planning
 architecture, **Vol 1:** 41
Platform Invoke (P/Invoke), **Vol 5:** 26
Platform SDK, **Vol 2:** 24
 Vol 3: 31–33
 debugging tools, **Vol 3:** 263
 Vol 4: 222
 file management tools, **Vol 3:** 264
 Vol 4: 222
 performance tools, **Vol 3:** 264
 Vol 4: 222
 testing tools, **Vol 3:** 264
 Vol 5: 1
 tools, **Vol 3:** 263
 Vol 4: 222

WinDBG, **Vol 3:** 33
plotting
 plotters in UNIX, **Vol 3:** 204
 plotters in Windows, **Vol 3:** 204
port UNIX shell scripts, **Vol 3:** 235
porting
 Sigaction call, **Vol 3:** 134
 porting from Win32 to Win64, **Vol 3:** 11
 porting methodologies, **Vol 1:** 49
POSIX, **Vol 2:** 12, 13, 14, 16, 29, 35, 36, 47,
 50, 59, 60, 61, 63, 76, 82, 83–84, 83, 102,
 107, 108, 109, 110, 111
POSIX.1, **Vol 2:** 13, 83, 102
POSIX-compliant threads, **Vol 2:** 14
predictive event management, **Vol 5:** 24
PrintDlg, **Vol 3:** 201
printing, **Vol 3:** 198–204, 198
 using APIs, **Vol 3:** 198–201, 198
 using MFC and ATL, **Vol 3:** 202–204, 202
 using system commands, **Vol 3:** 198
process, **Vol 3:** 45–58
 create, **Vol 2:** 34
 Vol 4: 51–53, 51
 create, example, **Vol 4:** 52
 CreateThread, **Vol 3:** 52, 59–62
 differences, UNIX, **Vol 3:** 11–13, 11
 groups, **Vol 2:** 34, 38
 hierarchy, **Vol 2:** 12, 35
 managing, **Vol 3:** 57
 multitasking, **Vol 2:** 11, 12
 Vol 3: 11
 multithreading, **Vol 2:** 12
 multiuser, **Vol 2:** 12
 priority, **Vol 2:** 39
 process hierarchy, **Vol 3:** 13
 replace process image, **Vol 2:** 35
 resource limits, **Vol 2:** 36–38, 37
 scheduling, **Vol 2:** 12, 39, 44
 Vol 3: 57
 spawn, **Vol 3:** 46
 wait, **Vol 3:** 49
 WaitForMultipleObjects, **Vol 3:** 49
 waiting, **Vol 2:** 36
 Windows Service, **Vol 3:** 13
process accounting, **Vol 3:** 56
process environment, **Vol 2:** 92
 Vol 3: 227–235, 227
 address space, **Vol 3:** 228
 computer information, **Vol 2:** 93
 Vol 3: 229
 environment block, **Vol 3:** 228
 environment variables, **Vol 3:** 228
 system messages, **Vol 2:** 93
 temporary files, **Vol 2:** 93
 Vol 3: 229
 variables, **Vol 2:** 92
process hierarchy, **Vol 3:** 13
process information, **Vol 3:** 49

process management, **Vol 1:** 1, 12–13
Vol 2: 11, 23, 34–40
Vol 3: 1
Vol 4: 1
Vol 5: 1
multitasking, **Vol 1:** 12
multithreading, **Vol 1:** 13
process hierarchy, **Vol 1:** 13
process pipes, **Vol 3:** 135–136, 135
introduction, **Vol 4:** 124–126, 124
UNIX, example, **Vol 4:** 124
process resource limits
managing, **Vol 4:** 56–57
process security, **Vol 3:** 121
process versus threads, **Vol 4:** 51, 55–56
production deployment plan, **Vol 1:** 50
programming model
Windows, **Vol 3:** 8
programming model UNIX, **Vol 3:** 8
proof of Cconcept, **Vol 2:** 22, 23
Proof of Concept (POC), **Vol 1:** 45
property sheet, **Vol 3:** 196
property sheets, **Vol 3:** 196
protected mode, **Vol 1:** 7
pthread, **Vol 3:** 59
Pthread, **Vol 2:** 12
Python, **Vol 2:** 99

Q

queuing systems, **Vol 1:** 18
Queuing Systems, **Vol 3:** 144

R

redevelopment
advantages, **Vol 4:** 22
Remote Copy, **Vol 5:** 22
Remote Desktops, **Vol 5:** 23
Remote Execute, **Vol 5:** 22
remote management, **Vol 2:** 99
Remote Procedure Call (RPC), **Vol 3:** 139, 140
Remote Shell, **Vol 5:** 22
remote shell commands, **Vol 2:** 98
remote, deployment in Interix, **Vol 2:** 119
repackaging, **Vol 3:** 245
requirements
high-level, definition, **Vol 1:** 29–30
resource files, **Vol 4:** 175
resource sharing, **Vol 5:** 23
resources, **Vol 3:** 220
REXX, **Vol 3:** 142
risks, **Vol 1:** 38–39
Root Window, **Vol 4:** 176
Ruby, **Vol 2:** 99
running .NET, from SFU, **Vol 2:** 107
running win32/64, from SFU, **Vol 2:** 107
Runtime Callable Wrapper(RCW), **Vol 5:** 28

runwin32, **Vol 2:** 107, 109

S

Samba, **Vol 1:** 15
scalable data types, **Vol 3:** 38
scaling, **Vol 2:** 135
Vol 3: 261
Vol 4: 219
scheduling process, **Vol 3:** 57
scheduling thread, **Vol 3:** 58
scheduling threads, **Vol 3:** 88
scope
solution, project, **Vol 1:** 28
scoping, **Vol 3:** 222
screen resolution, **Vol 4:** 174
script migration tasks, **Vol 3:** 235–239, 236
scripting languages, **Vol 3:** 142
Perl, **Vol 3:** 142
REXX, **Vol 3:** 142
Tcl/Tk, **Vol 3:** 142
Windows Scripting Host, **Vol 3:** 142
scripts, **Vol 2:** 98
ActivePerl, **Vol 2:** 99
C shell, **Vol 2:** 99
Korn, **Vol 2:** 99
Perl, **Vol 2:** 98
Python, **Vol 2:** 99
Ruby, **Vol 2:** 99
Tcl/Tk, **Vol 2:** 99
scripts, migration, **Vol 2:** 94
secure code, **Vol 2:** 126
Vol 3: 253
Vol 4: 210
Secure Shell, **Vol 5:** 22
authentication, **Vol 5:** 22
encryption, **Vol 5:** 22
Secure Sockets Layer (SSL), **Vol 3:** 20
security, **Vol 2:** 34
Vol 3: 17–21, 17, 121–125
Access Control Entries (ACE), **Vol 3:** 19
access token, **Vol 3:** 19
active directory, **Vol 1:** 15, 16
application, code access, **Vol 4:** 15, 203
GID, **Vol 3:** 18, 123
group, **Vol 2:** 60
group security, **Vol 3:** 121
impersonation, **Vol 3:** 20, 124
process security, **Vol 3:** 121
Secure Sockets Layer (SSL), **Vol 3:** 20
security permissions, **Vol 3:** 18
smart cards, **Vol 3:** 17
System Access Control List (SACL), **Vol 3:** 19, 124
token, **Vol 2:** 59–60
UID, **Vol 3:** 18, 123
UNIX Security, **Vol 3:** 18, 121
UNIX versus Windows, security, **Vol 1:** 15
user, **Vol 2:** 60

- user authentication, **Vol 1:** 15
 - Vol 3:** 17
- user level security, **Vol 3:** 121
- Windows Security, **Vol 3:** 19–21, 19
- Windows versus UNIX, security, **Vol 1:** 15, 50, 65
- security descriptors, **Vol 3:** 19
- security model
 - unified, **Vol 1:** 15
- security plan, **Vol 1:** 50, 51
- security rights, **Vol 3:** 244
- security testing, **Vol 2:** 125
 - Vol 3:** 252
 - Vol 4:** 210
- SelectObject, **Vol 3:** 165, 174, 176, 177, 181, 183, 185, 187, 188
- SelectPen, **Vol 3:** 165
- self-healing, **Vol 3:** 242
- semaphores, **Vol 3:** 79
- serialization, **Vol 4:** 187
- server message block, **Vol 1:** 15
- Server Message Block (SMB), **Vol 3:** 16
- Service Control Manager (SCM), **Vol 3:** 13, 142
- Service monitoring
 - exception monitoring, **Vol 5:** 25
- services, **Vol 2:** 17, 74, 105
 - Vol 4:** 130–131, 130
- Services For UNIX 3.5
 - command- line, **Vol 2:** 10
 - installation, **Vol 2:** 10
 - overview, **Vol 2:** 8
- setrlimit, **Vol 3:** 53
- SetTextColors, **Vol 3:** 188, 190, 193
- SFU
 - development environment, **Vol 1:** 55
- SFU based application, leveraging to .NET, **Vol 5:** 27
- SFU 3.5
 - building application, **Vol 2:** 29
 - command- line, **Vol 2:** 10
 - debugging application, **Vol 2:** 30
 - debugging tools, **Vol 2:** 137
 - installation, **Vol 2:** 10
 - monitoring tools, **Vol 2:** 137
 - overview, **Vol 2:** 8
 - testing tools, **Vol 2:** 137
- shared memory, **Vol 2:** 47
 - Vol 3:** 138
 - Vol 4:** 129
- shared objects, **Vol 4:** 20
- sharing
 - data, **Vol 5:** 23
 - resource, **Vol 5:** 23
- shell scripting, **Vol 3:** 141
- shell scripts, migration, **Vol 2:** 94
- shells, **Vol 3:** 141
 - Bourne shell (sh), **Vol 3:** 141
 - C shell (csh), **Vol 3:** 141
 - command-line shells, **Vol 3:** 141
 - Korn shell (ksh), **Vol 3:** 141
- shells and scripting, **Vol 1:** 19
- side by side deployment, **Vol 3:** 247
- SIGINT, **Vol 4:** 112–116, 112, 113
- signals, **Vol 1:** 13
 - Vol 2:** 12–13, 13, 61–63, 61
 - Vol 3:** 127–134
 - .NET, **Vol 4:** 116–117
 - bsd_signal, **Vol 2:** 62, 63
- Signals, **Vol 3:** 21
- Simple Network Management Protocol (SNMP), **Vol 5:** 25
- sizing, **Vol 3:** 221
- smart cards, **Vol 3:** 17
- SNMP, **Vol 5:** 25
- socket
 - control, **Vol 2:** 48, 49
- socket handles, **Vol 3:** 21, 125
- sockets, **Vol 3:** 23, 139
 - Vol 4:** 121–124, 121
 - domain name, **Vol 2:** 69, 70
 - host name, **Vol 2:** 69, 70
 - socket calls, **Vol 2:** 70
 - Transport Level Interface (TLI), **Vol 2:** 72
 - User Datagram Protocol (UDP), **Vol 2:** 72
 - winsoc, **Vol 2:** 54, 68, 70
 - X/Open Transport Interface (XTI), **Vol 2:** 72
- solution
 - functional specification, **Vol 1:** 46
- solution design, architecture development, **Vol 1:** 43–45
 - conceptual design, **Vol 1:** 43–44, 43, 44
 - logical design, **Vol 1:** 43, 44
 - physical design, **Vol 1:** 44, 51
- solution development plan, **Vol 1:** 49
- solution feature set, **Vol 1:** 47
- solution testing plan, **Vol 1:** 49
- spawn, **Vol 3:** 46
- SpinLocks, **Vol 3:** 71
- SQL Server, **Vol 5:** 25
- stabilizing
 - goals, **Vol 3:** 255–256, 255
 - Vol 4:** 213–224, 213
- multiprocessor considerations, **Vol 2:** 136
 - Vol 3:** 262
 - Vol 4:** 220
- network utilization, **Vol 2:** 136
 - Vol 3:** 262
 - Vol 4:** 220
- scaling, **Vol 2:** 135
 - Vol 3:** 261
 - Vol 4:** 219
- testing, **Vol 2:** 122
 - Vol 3:** 249
 - Vol 4:** 207

- tuning, **Vol 2:** 134
 - Vol 3:** 260
 - Vol 4:** 218
- stabilizing goals, **Vol 2:** 128
- stabilizing phase, **Vol 2:** 128
 - Vol 3:** 255
 - Vol 4:** 4, 213
- standard (stream) file access, **Vol 3:** 108
- standard error stream, **Vol 3:** 107
- standard input stream, **Vol 3:** 107
- standard input,output,error, **Vol 2:** 109
- standard output stream, **Vol 3:** 107
- static application
 - introduction, **Vol 4:** 21
 - migration strategies, **Vol 4:** 21–22, 22
- status bars, **Vol 3:** 197
- streams
 - standard error, **Vol 3:** 107
 - standard input, **Vol 3:** 107
 - standard output, **Vol 3:** 107
- supervisor mode, **Vol 1:** 7
- support, **Vol 5:** 13
- Symmetric Multiprocessing (SMP), **Vol 3:** 12
- synchronization, **Vol 3:** 22
- synchronizing
 - shared resources, **Vol 3:** 106
 - thread, **Vol 3:** 65
- System Access Control List (SACL), **Vol 3:** 19, 124
- system configuration, **Vol 1:** 16
- system messages, **Vol 2:** 93
 - log, **Vol 4:** 197–200, 197
- System V, **Vol 2:** 13, 14, 30, 47, 64, 65, 68, 83, 84, 93, 94, 105
- Systems Management Server (SMS), **Vol 3:** 246
 - Vol 5:** 8, 24

T

- Tcl/Tk, **Vol 2:** 99
 - Vol 3:** 142
- TCP/IP, **Vol 3:** 23, 139
- Telnet
 - Windows, **Vol 5:** 22
- temporary files, **Vol 2:** 93
 - Vol 3:** 229
- Termcap, **Vol 3:** 24
- Terminal Services, **Vol 3:** 24
- TerminateThread, **Vol 3:** 62–65, 62
- Terminfo, **Vol 3:** 24
- test data, **Vol 2:** 122
 - Vol 3:** 250
 - Vol 4:** 207
- test environment, **Vol 2:** 122
 - Vol 3:** 249
 - Vol 4:** 207

- test network load balancing, **Vol 2:** 126
 - Vol 3:** 253
 - Vol 4:** 211
- test plan, **Vol 2:** 122, 128
 - Vol 3:** 249, 255
 - Vol 4:** 207, 213
- test report, **Vol 2:** 122
 - Vol 3:** 250
 - Vol 4:** 207
- test specification, **Vol 2:** 122
 - Vol 3:** 249
 - Vol 4:** 207
- testing, **Vol 2:** 122–126, 128, 122–126
 - Vol 3:** 255
 - Vol 4:** 4, 122–126, 213
- code component, **Vol 2:** 123
 - Vol 3:** 250
 - Vol 4:** 208
- code review, **Vol 2:** 31–32
 - Vol 4:** 37–38
- database, **Vol 2:** 124
 - Vol 3:** 252
 - Vol 4:** 209
- integration, **Vol 2:** 124
 - Vol 3:** 251
 - Vol 4:** 209
- management, **Vol 2:** 126
 - Vol 3:** 253
 - Vol 4:** 211
- network load balancing, **Vol 2:** 126
 - Vol 3:** 253
 - Vol 4:** 211
- security, **Vol 2:** 125
 - Vol 3:** 252
 - Vol 4:** 210
- tools, **Vol 3:** 263
 - Vol 4:** 221
- unit, **Vol 2:** 123
 - Vol 3:** 250
 - Vol 4:** 208
- text, **Vol 3:** 183–196
 - calculating metrics, **Vol 4:** 163
 - display, **Vol 4:** 156–160, 156
 - drawing, **Vol 3:** 187–189, 187
 - Vol 4:** 161–164
 - formatting, **Vol 3:** 190–192, 190
 - widgets and controls, **Vol 3:** 195
- Text Widgets, **Vol 4:** 164–166, 164
- thread, **Vol 3:** 58–98
 - attributes, **Vol 2:** 44
 - cancel, **Vol 3:** 62–65
 - creating, **Vol 3:** 59–62, 59
 - detaching, **Vol 2:** 42
 - detachstate, **Vol 3:** 84
 - managing multiple threads, **Vol 3:** 95–98, 95
- model, **Vol 2:** 40
- prioritizing, **Vol 3:** 88–95, 88

- priority, Vol 2: 34, 44, 46
- pthread, Vol 3: 59
- scheduling, Vol 2: 12, 34, 44, 46
 - Vol 3: 58, 88–95, 88
- synchronization, Vol 2: 42–43
 - Vol 3: 65
- synchronization, condition variables, Vol 2: 43
- synchronization, mutex, Vol 2: 43
- synchronization, semaphores, Vol 2: 43
- TerminateThread, Vol 3: 62
- terminating, Vol 2: 42
- thread attributes, Vol 3: 83–87, 83
- Thread Local Storage (TLS), Vol 3: 101–104
 - Vol 4: 93–95, 93, 94
- thread management, Vol 1: 1
 - Vol 2: 11, 14, 23, 34, 40–45
 - Vol 4: 1
- thread synchronization, Vol 3: 65–83, 65
 - critical section, Vol 3: 77–79, 77
 - Interlocked Exchange, Vol 3: 69–71, 69
 - mutex, Vol 3: 73
 - semaphores, Vol 3: 79–83, 79
 - SpinLocks, Vol 3: 71–73, 71
- threads
 - create, Vol 4: 58–60, 58
 - create,UNIX example, Vol 4: 58
 - CreateThread, Vol 3: 52
 - differences,UNIX, Vol 3: 11–13, 11
 - management, Vol 4: 57–83, 57
 - multiple, Vol 4: 80–83, 80
 - multiple,non-synchronized,UNIX, Vol 4: 64
 - multitasking, Vol 3: 11
 - scheduling,priorities, Vol 4: 77–80, 77
 - scheduling,priorities,example, Vol 4: 78
 - synchronization,techniques, Vol 4: 63–77, 66
 - synchronize, Vol 4: 63
 - synchronize,using Interlocked Compare Exchange, Vol 4: 66–68
 - synchronize,using mutexes, Vol 4: 74–77, 74
 - synchronize,using semaphores, Vol 4: 68–74, 68
 - terminate, Vol 4: 60–63, 60
 - terminate, .NET example, Vol 4: 62
- threads versus process, Vol 4: 51, 55–56
- Timeouts, Vol 4: 170–173, 170
- Timers, Vol 4: 170–173, 170
- Tokens, Vol 3: 123–125
- tool bars, Vol 3: 197
- tools, Vol 2: 9, 10, 18, 21, 22, 29, 30, 40, 50, 54, 57, 75, 81, 82, 90, 96, 98, 99, 101, 103, 104, 106, 118, 128, 138
 - Vol 3: 263
- debugging, Vol 2: 17, 18, 137
- development, Vol 2: 8, 17

- monitoring, Vol 2: 137, 138
 - Vol 3: 264
 - Vol 5: 1
- packaging, Vol 4: 204–206, 204
- Platform SDK, Vol 3: 263
 - Vol 4: 222
- source test, Vol 2: 139
 - Vol 3: 265
 - Vol 5: 2
- testing, Vol 2: 137, 138
 - Vol 3: 263, 265
 - Vol 4: 221
 - Vol 5: 1
- Visual Studio tools, Vol 3: 263
 - Vol 4: 221
- win64, Vol 2: 139
 - Vol 3: 265
 - Vol 5: 2
- transaction processing, Vol 5: 24
- tuning, Vol 2: 134–137
 - Vol 3: 260
 - Vol 4: 218
- performance, Vol 2: 134
 - Vol 3: 260
 - Vol 4: 218
- two-phase deployment, Vol 3: 247
- type-safe code, Vol 4: 53

U

- UI
 - character-based UI, Vol 2: 82
 - curses, Vol 2: 84
 - Motif, Vol 2: 84
 - ncurses, Vol 2: 84
 - OpenGL, Vol 2: 84
 - POSIX terminal I/O, Vol 2: 83
 - terminal I/O, Vol 2: 83
 - termio, Vol 2: 83
- UID, Vol 1: 15
 - Vol 3: 18, 123
- UNC, Vol 2: 15
- Uniform Data Model (UDM), Vol 3: 9
- unit testing, Vol 2: 123
 - Vol 3: 250
 - Vol 4: 208
- Universal Naming Convention, Vol 2: 15
- UNIX
 - architecture, Vol 1: 8–10
 - evolution, Vol 1: 8–10, 9
 - kernel, Vol 2: 10
 - shell, Vol 2: 10
 - utilities, Vol 2: 10
 - Windows versus UNIX, architecture, Vol 1: 10
- UNIX file system features, Vol 3: 14–17, 15
- UNIX Migration Project Guide (UMPG), Vol 1: 51
- UNIX Security, Vol 3: 18, 121

UNIX/64
 data model, **Vol 3:** 8–11, 9
 unixODBC driver manager, **Vol 2:** 96
 UnixWare, **Vol 1:** 9
 unnamed pipes, **Vol 3:** 23, 135
 user and operations training plan, **Vol 1:** 50
 user authentication, **Vol 1:** 15
 user interface architecture
 Windows, **Vol 3:** 148, 150–151
 XWindows, **Vol 3:** 147
 user interface differences, **Vol 3:** 23–24, 23
 user interfaces, **Vol 1:** 16
 user level security, **Vol 3:** 121
 user mode, **Vol 1:** 7

V

validation, **Vol 3:** 221
virtual address space, **Vol 3:** 11, 35, 101, 138
virtual memory, **Vol 3:** 14
virtual memory management, **Vol 1:** 14
Visual Studio .NET 2003, **Vol 3:** 33–37
 building, **Vol 3:** 33
 debugging, **Vol 3:** 37
Visual Studio tools, **Vol 3:** 263
 Vol 4: 221

W

wait, **Vol 3:** 49–51, 49
WaitForMultipleObjects, **Vol 3:** 49
WBEM, **Vol 5:** 24
Web Service, **Vol 5:** 24
Web-Based Enterprise Management
 (WBEM), **Vol 5:** 24
Win32 subsystem, **Vol 1:** 7, 11, 12, 33
win32, Interix, **Vol 2:** 101
win32/64, from SFU, **Vol 2:** 107
win64
 tools, **Vol 2:** 139
 Vol 3: 265
 Vol 5: 2
Win64
 data model, **Vol 3:** 8–11, 9
 overview, **Vol 3:** 7–11
WinDBG, **Vol 3:** 33, 224
Window
 creating, **Vol 4:** 143–144, 143
 types, **Vol 4:** 138
window creation, **Vol 3:** 154
windowproc, **Vol 3:** 208
Windows
 connectivity to UNIX, **Vol 5:** 21
 multiple users, **Vol 5:** 23
 security model, comparison with UNIX, **Vol 2:** 16
 server, **Vol 5:** 8, 21, 22
 Telnet, **Vol 5:** 22
 Terminal Services, **Vol 5:** 23

 UNIX versus Windows, architecture, **Vol 1:**
 10–20, 10
 Windows 2003 architecture, **Vol 1:** 7–8, 7
 Windows 2003 Event Log, **Vol 5:** 25
 Windows API, **Vol 5:** 27
 functional categories, **Vol 3:** 8
 overview, **Vol 3:** 7–11, 8
 Windows Event objects, **Vol 3:** 133
 Windows forms
 advantages, **Vol 4:** 17
 Windows Forms, **Vol 4:** 133–134, 133
 Windows Forms versus X Windows
 Look and Feel, **Vol 4:** 136–138, 136
 User Interface, **Vol 4:** 28, 134, 135–136
 Window Types, **Vol 4:** 138–139, 138
 Windows Installer Service, **Vol 2:** 98
 Vol 3: 242
 self-healing, **Vol 3:** 242
 Windows Management Instrumentation
 (WMI), **Vol 3:** 230
 Windows Messages, **Vol 3:** 132
 Windows NTFS, **Vol 2:** 52
 Windows Open System Architecture
 (WOSA), **Vol 3:** 139
 Windows Platform SDK, **Vol 2:** 24
 Windows registry, **Vol 3:** 237
 Windows Scripting Host, **Vol 3:** 142
 Windows Security, **Vol 3:** 19–21, 19
 Windows service, **Vol 3:** 13, 142
 Vol 4: 20, 130–131, 131
 Windows Services for UNIX
 command- line, **Vol 2:** 10
 installation, **Vol 2:** 10
 overview, **Vol 2:** 8–10, 8
 Windows Subsystems, types of
 subsystems, Win32 subsystem, POSIX
 subsystem, **Vol 1:** 11
 WinInstall, **Vol 3:** 244
 Winsock, **Vol 3:** 23, 139
 WinSocket, **Vol 3:** 127
 working directory, **Vol 3:** 113
 WRQ Reflection, **Vol 5:** 23
 WSH, **Vol 5:** 25
 wvisible, **Vol 2:** 109

X

X Lib, **Vol 3:** 164, 211
X library, **Vol 3:** 24
X Server, **Vol 3:** 147, 150
 Vol 4: 34, 135
X terminals, **Vol 3:** 24
X Toolkit Intrinsic, **Vol 3:** 24, 164
X windows, **Vol 4:** 22, 28, 133, 134, 135, 136,
 138, 139, 140, 141, 143, 144, 145, 147,
 148, 154, 155, 157, 161, 162, 164, 166,
 167, 169, 170, 171, 173, 174, 175, 176,
 177, 178, 179, 180, 183

- X Windows, **Vol 3**: 24
 - Vol 5**: 21, 22–23, 22, 23
 - building, **Vol 2**: 82
 - characters, **Vol 2**: 82
 - client, **Vol 2**: 80
 - clients, **Vol 5**: 22
 - header, **Vol 2**: 82
 - libraries, **Vol 2**: 82
 - manager, **Vol 2**: 81
 - programs, Interix, **Vol 2**: 81
 - server, **Vol 2**: 80
 - Vol 5**: 21, 22
 - X11 server, **Vol 2**: 81
- X Windows versus Windows Forms
 - Look and Feel, **Vol 4**: 136–138, 136
 - User Interface, **Vol 4**: 28, 134, 135–136
- Window Types, **Vol 4**: 138–139, 138
- X11 server, **Vol 2**: 81
- Xlib, **Vol 3**: 24, 164, 198, 205, 209, 211, 212
- XmCreateFileDialog, **Vol 3**: 164
- XML
 - framework, introduction, **Vol 4**: 13–15, 13
- XML Web Services, **Vol 4**: 189
- XtAppInitialize, **Vol 3**: 154
- XtOpenApplication, **Vol 3**: 154, 155
- XtVaAppInitialize, **Vol 3**: 152, 154, 160
- XtVaOpenApplication, **Vol 3**: 154
- XWindows and Win32/64 GUI
 - comparison, **Vol 3**: 147–149
- XWindows and Windows
 - terminology mapping, **Vol 3**: 208–212