

---

# INDEX

## A

### -a option:

- nm** utility, 230–31
- objdump** utility, 235–36
- strings** utility, 237

Actions, jCVS, 194–95

Ada, 44

**add** option, CVS, 186–87

Adding break points, 156–58

Aegis, 13

### Applets:

- building, 271–72
- testing with Netscape, 272

### Applications, Java:

- building, 270–71
- running, 271
- using **gcj** compiler to build, 271

**ar** utility, 13, 226–28

Architecture, creating, 4

Archive, 226

Arcom SBC-GX1 board, 258–60

Artesyn PM/PPC mezzanine card, 260

Assembler code, creating, 62

Assemblers, 10, 12, 43–44, 91–92, *See also*

### Compilers

GNU assembler, 44

Automatic indentation, VIM, 40–41

## B

### -b option:

- cscope**, 221
- diff** command, 217

**-B** option, **diff** command, 217

**-bad** option, **indent** program, 209

**-bap** option, **indent** program, 209

**-bbb** option, **indent** program, 210

Berkley style, 208–9

Block reordering pass, 46

Branches, 8

**break** command, GDB, 141, 146, 156–58

### Break points, GDB:

- adding, 156–58
- continuing from, 158
- deleting, 160
- disabling, 159
- enabling, 159–60

Build process, 101

## C

C language, and GCC, 45

- c option:
  - cscope**, 221
  - diff** command, 218
- C++ code, compiling, 66
- C++ language, and GCC, 45
- C90/C95/C99, 45
- Caldera Inc., 16
- cbrowser** utility, 203, 221
- ccl** compiler, 70
- C\_INCLUDE\_PATH** environment variable, 55, 58
- clean** target, 104–5
- collect2** program, 70
- commit** option, CVS, 186
- Compilation:
  - assembling a program, 91–92
  - compiling a program, 53–91
  - compiling to intermediate levels, 61–62
    - creating assembler code, 62
  - with debug support, 63–64
  - of Fortran programs, 96–97
  - include files, 92
  - libraries:
    - creating, 92
    - standard libraries, 93–94
  - linking a program, 91
  - with optimization, 63–64
  - of Pascal programs, 94–96
    - Free Pascal (fpc), 95–96
    - GNU Pascal, 96
  - simple, 59
  - of source code for languages other than C, 66–70
    - C++ code, 66
    - Java code, 68–70
    - Objective C code, 66–67
  - static and dynamic linking, 65–66
  - warning and error messages, handling, 92
- COMPILER\_PATH** environment variable, 55
- Compilers, 10, 12, 43–91, *See also* Assemblers
  - compilation process, stages in, 44
  - default file types, 60–61
  - g77** compiler, 96–97
  - GCC (GNU Compiler Collection), 47
    - GCC 3.x new features, 46–47
    - GNU C library, 45
    - languages supported by, 44–45
      - and standard libraries, 45
  - GNU C compiler, 43, 135
    - installing, 48–58
    - languages supported by, 43
  - Oberon, 98
  - Ruby, 98
  - Smalltalk, 98
- Concurrent Versions System (CVS), 7, 12, 175–201
  - add** option, 186–87
  - checkout** command, with **-r** option, 189
  - client, using, 180–81
  - commit** option, 186
  - configuring, 177–79
  - conflict resolution, 185–86
  - diff** option, 184–85
  - differences, determining, 184–85
  - installing, 176–80
  - jCVS, 190–96
    - actions/verbs, 194–95
    - checking out a project, 191–93
    - client, configuring, 191
    - editing projects, 195–96
    - installing, 190
    - server definitions file, setting up, 193–94
    - using, 191–96
    - WorkBench, 196
  - local repositories, 181–82
  - managing, 176–80
  - policies, 176
  - project:
    - adding files to, 186–87
    - checking back in, 186
    - checking out, 182–83
    - finding the status of, 183–84
    - importing into the repository, 179–80
    - removing files from, 187–88
    - renaming files within, 188

- project management and communication, 176
- remote repositories, 182
- remove** option, 187–88
- secure remote access with, 199–201
- Secure Shell (ssh) access, 199–201
- status** option, 183
- tags and releases, 189
- update** option, 185–86, 188
- using Emacs with, 196–99
  - pcl-cvs, 197–99
  - working copy:
    - removing, 188–89
- configure** script, GCC, 49, 52
- continue** command, GDB, 138, 144, 158
- CPLUS\_INCLUDE\_PATH** environment variable, 55
- cproto** utility, 222–23
- Cross compilation tools, 247–50
  - BDM (background debugging mode), 249–50
  - cross debugger, 248
    - building, 250
  - in-circuit emulators, 249
  - JTAG (Joint Test Action Group), 249–50
  - software emulators, 248
- Cross-platform development, 239–61
  - Arcom SBC-GX1 board, 258–60
  - Artesyn PM/PPC mezzanine card, 260
  - development cycle, 241–43
    - building on the host machine, 241–42
    - remote debugging, 243
    - transfer to the target machine, 242–43
    - writing code, 241
  - hardware used for, 258–60
  - host machine, 240
  - native and cross compilers, 241
  - target machine, 240
- cscope** utility, 203, 219–21
  - cscope.out** symbol file, 219
  - initial screen, 219
  - options used with, 221
  - using, 219–21
- ctags utilities, 18
- Cut and paste commands, Emacs, 24
- CVS, *See* Concurrent Versions System (CVS)
- cvs diff** command, 215
- \$CVSEDITOR** environment variable, 180
- CVSROOT** variable, 181–82
- CVS\_RSH** environment variable, 181
- D**
- d** option:
  - ltrace** utility, 224
  - objdump** utility, 236
- Davis, John E., 18
- ddd** debugger, 172
- Dead code elimination pass, 46
- Deadlines, creating, 14
- Debian Project, 16
- Debuggers, 5, 10, 12
  - ddd** debugger, 172
  - KDE debugger, 169
  - xxgdb** debugger, 173–74
- debugging** option, **objdump** utility, 236
- Default goal, 104
- delete** command, GDB, 160
- Deletion commands, Emacs, 21
- Dependencies, resolving, 15
- Development team, managing, 14, 15
- diff** option, CVS, 184–85
- diff** utility, 203, 215–18
  - defined, 215
  - diff3** utility, 218
  - other forms of, 218
  - options used with, 217–18
  - sdiff** utility, 218
  - using, 215–18
- Directives, **make** utility, 131–33
  - ifdef** directive, 132
  - ifeq** directive, 132
  - ifndef** directive, 133
  - ifneq** directive, 132
- disable** command, GDB, 159

**disable display** command, GDB, 154

**display** command, GDB, 153–54

Documentation, 1

Dynamic linking, 65–66

## E

**e l** command, **sdiff** utility, 218

E-mail and collaboration, 13

**e r** command, **sdiff** utility, 218

**echo** command, suppression of, 111

\$EDITOR environment variable, 180

Editors, 10, 11

Emacs, 11, 17–32, 135

extensibility, 17–18

folding code, 18

Jed, 11, 18, 32–36

syntax, 18

tag support, 18

VIM, 37–42

what to look for in, 17–18

Elisp, 17

Emacs, 11, 17–32, 135

auto-completion option, 23

basic concepts, 20–24

buffers, 24–25

c-mode commands, 26

commands related to **TAGS**, 29

compiling with, 30–31

commands, 31

cut and paste commands, 24

defined, 18–19

deletion commands, 21

file commands, 22

initial screen, 19

language modes, 26

movement commands, 20–21

search and replace commands, 22–24

options, 23

starting from within X-Windows, 19

tags, 27–29

tags-search function, 28

tutorial, 20

using with CVS, 196–99

pcl-cvs, 196–99

windows, 24–25

commands, 25

Xemacs, 32

Embedded Linux, 245

Embedded systems, 239, 243–61

Arcom SBC-GX1 board, 258–60

Artesyn PM/PPC mezzanine card, 260

defined, 243–44

development systems for, 246–47

everyday examples of, 244

and moving parts, 244

operating systems, 245

and power consumption, 245

software applications for, 246

target system hardware, 246–47

real-time, 247

testing methodology, 247

and timing/clock-cycle changes due to  
code optimization, 65

Empty commands, 107

**enable** command, GDB, 159–60

**enable display** command, GDB, 154

End-user manuals/guides, 9

Environment variables, GNU C compiler:

**C\_INCLUDE\_PATH**, 55

**COMPILER\_PATH**, 55

**CPLUS\_INCLUDE\_PATH**, 55

**GCC\_EXEC\_PREFIX**, 55

**LD\_LIBRARY\_PATH**, 55

**LIBRARY\_PATH**, 55

**OBJC\_INCLUDE\_PATH**, 55

**TMPDIR**, 54

**etags** command, 27

**etags** utilities, 18

Explicit rules, **make** utility, 118

## F

**-f** option:

**ltrace** utility, 224

**objdump** utility, 236

**sed** utility, 215

**strings** utility, 237

**-f reffile** option, **cscope**, 221

**file** command, GDB, 137, 166

File commands, Emacs, 22

**finish** command, GDB, 144–46, 166–68

Folding code, 18

Fortran programs:

  compilation of, 96–97

  and GCC, 45, 45–46

FQDN (fully qualified domain name), 194, 200

**frame** command, GDB, 146

Free Pascal (fpc), 95–96

Functional testing, 7

## G

**g77** compiler, 96–97

**gcc** command, 55–56

  machine dependent options, 77–90

    AMD29K options, 79

    ARC options, 87

    ARM options, 79–80

    AVR options, 88

    Clipper options, 86

    Convex options, 78–79

    DEC Alpha options, 85–86

    H8/300 options, 86

    HPPA options, 84–85

    i386 options, 84

    IA-64 options, 89

    Intel 960 options, 85

    M32R/D options, 80

    M68hc1x options, 77–78

    M88K options, 80–81

    M680x0 options, 77

    MCore options, 88–89

    MIPS options, 83–84

    MN10200 options, 80

    MN10300 options, 80

    NS32K options, 88

    RS/6000 and PowerPC options, 81–82

    RT options, 82–83

    S/390 and zSeries options, 89

    SH options, 86

    SPARC options, 78

    System V options, 87

    TMS320C3x/C4x options, 87

    V850 options, 87

    VAX options, 78

    Xtensa options, 89–90

  options, 68–70

    assembler option, 76

    C language options, 70–71

    C-only warning options, 73

    C++ language options, 71–72

    code generation options, 90–91

    debugging options, 73–74

    directory options, 77

    language independent options, 72

    linker options, 76–77

    machine dependent options, 77–90

    Objective-C language options, 72

    optimization options, 74–75

    overall options, 70

    preprocessor options, 76

    target options, 77

    warning options, 72–73

**GCC** (GNU Compiler Collection), 12, 47

  file types used with, 60–61

  GCC 3.x new features, 46–47

    GCC version 3.0.1 improvements, 47

    GCC version 3.0.2 improvements, 47

    GCC version 3.0.3 improvements, 47

    improved documentation, 47

    language improvements, 46

    new language support, 46

    new targets, addition of, 47

    optimization improvements, 46

    target specific improvements, 47

**GCC\_EXEC\_PREFIX** environment variable, 55

**gdb**, *See also* GNU debugger

  common commands, 137–38

  downloading/building, 168

  final installation, 168

  installing, 168

  running on host using serial port, 253–54

  sample **gdb** session, 138–41

  using with GNU Emacs, 164–65

**gdbserver** program, 250–58, 253

  attaching to a running process using, 255

  running on host using TCP/IP, 252–53

- running on target using serial port, 253
  - running on target using TCP/IP, 251–52
  - serial cable, 254
  - step-by-step procedures, 250–51
  - using with GNU debugger, 250–55
- glibc, 45
- Global Null pointer elimination test elimination pass, 46
- GNU assembler, 44
- GNU binary utilities, 203, 226–38
  - addr2line** utility, 237–38
  - ar** utility, 226–28
  - ldd** utility, 238
  - nm** utility, 228–31
    - listing debug symbols, 231
    - listing line numbers in source files, 230–31
  - objcopy** utility, 231–32
  - objdump** utility, 232–36
  - ranlib** utility, 228
  - size** utility, 236
  - strings** utility, 237
  - strip** utility, 228–31
- GNU C compiler, 43, 135
  - building/installing, 48–54
    - configure** script, running, 49–51
    - final install, 52
    - make bootstrap** command, 51
    - make install** command, 52
    - typical directory tree after installation, 52–54
    - unpacking source code, 48–49
  - and development tools installations, 58
  - downloading the latest version, 48
  - environment variables, 54–56
    - C\_INCLUDE\_PATH**, 55
    - COMPILER\_PATH**, 55
    - CPLUS\_INCLUDE\_PATH**, 55
    - GCC\_EXEC\_PREFIX**, 55
    - LD\_LIBRARY\_PATH**, 55
    - LIBRARY\_PATH**, 55
    - OBJC\_INCLUDE\_PATH**, 55
    - TMPDIR**, 54
  - installing, 48–58
  - languages supported by, 43
  - location of include files, setting, 57–58
  - location of libraries, setting, 56–57
  - manual pages path, setting, 58
  - PATH variable, setting, 56
  - post-installation tasks, 56–58
- GNU debugger (GDB), 12, 135–74
  - break** command, 141, 146, 156–58
  - break points:
    - adding, 156–58
    - continuing from, 158
    - deleting, 160
    - disabling, 159
    - enabling, 159–60
  - continue** command, 138, 144, 158
  - controlling execution, 144–46
  - defined, 136
  - delete** command, 160
  - disable** command, 159
  - disable display** command, 154
  - display** command, 153–54
  - enable** command, 159–60
  - enable display** command, 154
  - file** command, 137
  - files and shared libraries, 163
  - finish** command, 144–46, 166–68
  - frame** command, 146
  - getting started with, 136–43
    - common **gdb** commands, 137–38
    - passing command-line arguments to program being debugged, 141–43
    - sample **gdb** session, 138–41
  - info display** command, 154
  - kill** command, 144
  - list** command, 141
  - next** command, 137–38, 141, 144, 146
  - object file formats supported by, 136
  - optimized code:
    - debugging, 160–62
  - print expr** command, 138
  - pwd** command, 154
  - quit** command, 137–38, 141, 144
  - return** command, 144
  - run** command, 138, 142–44
  - running processing:
    - debugging, 165–68

- set** command, 155–56
- set env** command, 155
- show** command, 143
- show environment** command, 154–55
- show path** command, 154
- stack, dealing with, 146–51
- step** command, 137–38, 144–46
- sum.c** program, 151
- undisplay** command, 154
- unset env** command, 155
- using stubs with, 256
- variables:
  - automatically displaying with each command, 153–54
  - displaying, 151–56
  - environment variables, 154–55
  - modifying, 155–56
  - program, 151–52
- GNU Emacs, 12
- GNU make utility, *See* make utility
- GNU Pascal, 96
- GNU Public License (GPL), 190
- GNU style, 206–7
- H**
- h** option:
  - ltrace** utility, 224
  - objdump** utility, 233–34
- Hardware platform, 10–11
  - selection criteria for, 13–14
- hexdump** utility, 13
- I**
- i** option, **diff** command, 217
- IDE drives, 278–79
- ifdef** directive, 132
- ifeq** directive, 132
- ifndef** directive, 133
- ifneq** directive, 132
- Implicit rules, **make** utility, 118–19
- indent** utility, 203, 204–14
  - blank lines/comments, 209–10
  - braces, formatting, 210–11
  - coding styles, selecting, 206–9
  - declarations, formatting, 211–12
  - efficient use of, 206
  - getting started with, 205–6
  - long lines, breaking, 212–13
  - options summary, 213–14
- .indent.pro**, 206
- info break** command, 159–60
- info display** command:
  - GDB, 154
- J**
- Java:
  - applets:
    - building, 271–72
    - testing with Netscape, 272
  - applications:
    - building, 270–71
    - running, 271
    - using **gcj** compiler to build, 271
  - compiler, 264
  - Embedded Java, 274
  - and GCC, 45
  - Java 2 Micro Edition (J2ME), 275
  - Java 2 SDK, 267–69
  - Java code, compiling, 270–71
  - Java Virtual Machine (JVM), 263–64
  - Jboss, 266–67
  - Jikes, 272–74
  - Kaffe, 264–65
  - platform independent development with, 263–75
  - real-time, 274
  - source code file:
    - creating, 270
    - wireless applications, 275
- Java 2 Micro Edition (J2ME), 275
- Java 2 SDK, 267–69
  - Enterprise Edition, 267
  - getting/installing from Sun Microsystems, 269
- jar** files, creating, 269
- Standard Edition, 267–68
  - non-blocking I/O, 267–68
  - security enhancements, 268
  - Web services, 268

- Java code, compiling, 68–70
- Java Runtime Environment, 190
- Java Virtual Machine (JVM), 263–64
- Jboss, 266–67
- jcl** program, 70
- jCVS, 190–96
  - actions/verbs, 194–95
  - checking out a project, 191–93
  - client, configuring, 191
  - editing projects, 195–96
  - installing, 190
  - Project Window, 192–93
  - server definitions file, setting up, 193–94
  - using, 191–96
  - WorkBench, 196
- Jed, 11, 18, 32–36
  - configuring, 33
  - folding code, 35–36
    - commands, 36
  - text interface, 34
  - using, 34
- Jikes, 272–74
- jvgenmain** program, 70
- K**
- K&R style, 207–8
- Kaffe, 264–65
- kdbg**, 169–71
- KDE debugger, 169–71
- Kibaale Children's Center (KCC), 37
- kill** command, 144
- killall** command, 179
- L**
- L** command, **sdiff** utility, 218
- l** option:
  - ar** utility, 227
  - nm** utility, 230–31
- ldconfig** program, 57
- LD\_LIBRARY\_PATH** environment variable, 55, 57
- Libraries:
  - creating, 92
  - GNU C library, 45
  - location of, 163
  - shared, 163
  - standard libraries, 45, 93–94
- LIBRARY\_PATH** environment variable, 55
- Linux:
  - configuring X-Windows on, 13
  - as operating system, 11
- Linux Development Platform Specifications (LDPS), 15
- Linux development workstation, 277–82
  - disk drive, 278–79
  - distributed development:
    - remote access, 280
    - source code repository, 280–81
  - memory, 278
  - parts of, 277–78
  - processor, 278
  - sizing the task, 279
  - standalone development workstation, 279–80
- Linux Standard Base (LSB), 15–16
  - current contributors to, 16
  - libraries, 15–16
- list** command, GDB, 141
- ltrace** utility, 203, 223–24
  - options used with, 224
- M**
- make bootstrap** command, 51
- make** utility, 13, 101–34
  - basic terminology, 103–5
  - control structures, 131–33
    - for** control structure, 133
  - default goal, 104
  - default goal, building, 108
  - dependencies, 104
  - directives, 131–33
    - ifdef** directive, 132
    - ifeq** directive, 132
    - ifndef** directive, 133
    - ifneq** directive, 132
  - include files, 109–10
  - input files, 103, 105–6



- latest distribution:
  - compilation, 133–34
  - installation, 134
  - obtaining, 133
- makefile** contents, 106–7
  - comments and blank lines, 107
  - rule to build the default goal, 107
  - rules to build individual objects, 107
  - rules to build other goals, 107
  - rules to clean previous builds, 107
  - rules to install, 107
  - variables, 106
- multiple **makefiles**/directories, working with, 122–29
  - building targets, 128–29
  - cleaning process, 129
  - individual targets, making, 129
  - makefile** in **common-dir** directory, 125–26
  - makefile** in **dns-dir** directory, 127–28
  - makefile** in **ftp-dir** directory, 126
  - makefile** in **tfp-dir** directory, 127
  - makefile** in the top directory, 123–25
- non-default goal, building, 108
- phony target, 104–5
- rules, 103, 110–19
  - anatomy of, 110–11
  - basic **makefile**, 111–13
  - command echoing, 111
  - example **makefiles**, 111–18
  - explicit, 118
  - implicit, 118–19
  - multiple commands in the same line, 110
  - with no dependencies, 111
- running, 108
- shell to execute commands, 109
- special features, 130–31
  - non-stop execution, 130–31
  - running commands in parallel, 130
- target, 104
- updating a target, rules for, 108
- variables, 119–22
  - automatic, 121–22
  - defining, 120
  - pre-defined, 121
  - types of, 120–21
- Makefile.in**, 106
- MandrakeSoft, 16
- Marketing Requirement Document (MRD), 3, 5
- Marketing/sales, 1–3
- Members, archive, 226
- Memory leakage, 6
- Meta key, 20
- Miscellaneous tools, 13, 203–38, 219–21
  - cbrowser** utility, 203, 221
  - cproto** utility, 222–23
  - cscope** utility, 203, 219–21
    - cscope.out** symbol file, 219
    - initial screen, 219
    - options used with, 221
    - using, 219–21
  - diff** utility, 203, 215–18
    - defined, 215
    - diff3** utility, 218
    - other forms of, 218
    - options used with, 217–18
    - sdiff** utility, 218
    - using, 215–18
- GNU binary utilities, 203, 226–38
  - addr2line** utility, 237–38
  - ar** utility, 226–28
  - ldd** utility, 238
  - nm** utility, 228–31
  - objcopy** utility, 231–32
  - objdump** utility, 232–36
  - ranlib** utility, 228
  - size** utility, 236
  - strings** utility, 237
  - strip** utility, 228–31
- indent** utility, 203, 204–14
  - blank lines/comments, 209–10
  - braces, formatting, 210–11
  - coding styles, selecting, 206–9
  - declarations, formatting, 211–12
  - efficient use of, 206
  - getting started with, 205–6
  - long lines, breaking, 212–13
  - options summary, 213–14

**ltrace** utility, 203, 223–24  
     options used with, 224  
**sed** utility, 203, 215  
**strace** utility, 203, 224–25  
 Moolenaar, Bram, 37  
 Movement commands, Emacs, 20–21  
**N**  
**-n** option, **diff** command, 218  
**-nbad** option, **indent** program, 210  
**next** command, GDB, 137–38, 141, 144, 146  
**nm** utility, 228–31  
     debug symbols, listing, 231  
     line numbers, listing in source files, 230–31  
**O**  
**-o filename** option, **ltrace** utility, 224  
 Oberon, 44, 98  
**OBJC\_INCLUDE\_PATH** environment variable, 55  
**objcopy** utility, 231–32  
**objdump** utility, 232–36  
     file disassembly, 234–35  
     header information, displaying, 232–33  
     library files, displaying information about, 236  
     section headers, displaying, 233–34  
     source code disassembly, 235–36  
 Objective C, and GCC, 45  
 Objective C code, 45  
     compiling, 66–67  
 OpenSSH, 199  
 Operating system, 10, 11  
 OSDN (Open Source Developers Network), 7  
**P**  
**-p** option:  
     **diff** command, 217  
     **unified diff** command, 216  
 Pascal programs:  
     compilation of, 94–96  
         Free Pascal (fpc), 95–96  
         GNU Pascal, 96

pcl-cvs, 196–99  
     installing, 197  
     using, 197–99  
 Phony targets, 104–5  
 PRCS, 13  
**print expr** command, GDB, 138  
**printf()** function, 92  
 Project management software, 14  
**ps** command, GDB, 166  
**pserver** method, 182  
 pSOS, 245  
**pwd** command, GDB, 154  
**Q**  
**Q** command, **sdiff** utility, 218  
 QNX, 245  
 Quality Assurance (QA), 6  
**quit** command, GDB, 137–38, 141, 144  
**R**  
**R** command, **sdiff** utility, 218  
**-r** option:  
     **diff** command, 218  
     **ltrace** utility, 224  
**-R** option, **cscope**, 221  
**ranlib** utility, 13, 228  
 RCS, 13  
 Red Hat Software, 16  
 Register-renaming pass, 46  
 Regression (stress) testing, 6–7  
 Release notes, 8  
**remove** option, CVS, 187–88  
**return** command, GDB, 144  
 Review meetings, 5–6  
 RIP (Routing Information Protocol), 4  
 Ruby, 98  
 Rules, **make** utility, 103, 110–19  
     anatomy of, 110–11  
     basic **makefile**, 111–13  
     command echoing, 111  
     example **makefiles**, 111–18  
     explicit, 118

- implicit, 118–19
  - multiple commands in the same line, 110
  - with no dependencies, 111
- run** command, GDB, 138, 142, 143, 144
- S**
- s option:
  - ltrace** utility, 224
  - nm** utility, 228
- S option, **objdump** utility, 235, 236
- Sanity testing, 6
- Santa Cruz Operations, 218
- SCCS, 13
- SCSI drives, 278–79
- Search and replace commands, Emacs, 22–24
  - options, 23
- sed** utility, 203, 215
- set** command, GDB, 155–56
- set env** command, GDB, 155
- show** command, GDB, 143
- show environment** command, GDB, 154–55
- show path** command, GDB, 154
- silent flag, 111
- Smalltalk, 44, 98
- sob option, **indent** program, 210
- Software build process, 101
- Software development:
  - complexity of, 1
  - components of a development system, 10–13
    - assemblers, 12
    - compilers, 12
    - debuggers, 12
    - editors, 11
    - e-mail and collaboration, 13
    - hardware platform, 10–11
    - operating system, 11
    - version control systems, 12–13
    - X-Windows, 13
  - hardware platform, selection criteria for, 13–14
  - Linux Development Platform Specifications (LDPS), 15
  - Linux Standard Base (LSB), 15–16
    - current contributors to, 16
    - libraries, 15–16
  - managing the development process, 14–15
    - deadlines, creating, 14
    - dependencies, resolving, 15
    - development team, managing, 15
  - miscellaneous tools, 13
  - software development tools, selection criteria for, 14
- Software development project life cycle, 2–9
  - architecture, creating, 4
  - coding, 5–6
  - design documents, creating, 4–5
  - documentation, 8–9
  - functional specifications, writing, 4
  - implementation, 5–6
  - new features, 9
  - requirement gathering, 2–3
  - software releases, 8
    - branches, 8
    - release notes, 8
  - support, 9
  - testing, 6–7
    - functional testing, 7
    - regression (stress) testing, 6–7
    - sanity testing, 6
    - unit testing, 6
- Software development tools, and GNU C compiler, 58
- Software testing, 1
- Static linking, 65–66
- Static Single Assignment (SSA), 46
- status** option, CVS, 183
- step** command, GDB, 137–38, 144–46, 146
- Step into actions, debuggers, 138
- Step over actions, debuggers, 138
- strace** utility, 203, 224–25
- Stress testing, 6–7
- strip** utility, 228–31
- sum.c** program, GDB, 151
- Support cycle, 1

SuSE, 16

Syntax, editors, 18

## T

**-t** option:

**ar** utility, 227

**objdump** utility, 236

TAB character, 103, 107, 110

Tag support, editors, 18

**:tags** Command, VIM, 41–42

Tags-search function, Emacs, 28

**tar** command, 48

Target machine connection, 250–58

    debugging the debug session, 256–58

**gdb**, running on host using serial port,  
        253–54

**gdbserver** program, 250–58, 253

        attaching to a running process using,  
            255

        running on host using TCP/IP, 252–  
            53

        running on target using serial port,  
            253

        running on target using TCP/IP,  
            251–52

        serial cable, 254

        step-by-step procedures, 250–51

        using with GNU debugger, 250–55

        using stubs with GNU debugger, 256

Technical documentation, 9

Testing, 6–7

    functional testing, 7

    regression (stress) testing, 6–7

    sanity testing, 6

    unit testing, 6

**TMPDIR** environment variable, 54

**:tselect** Command, VIM, 41

**-tt** option, **ltrace** utility, 224

TurboLinux Inc., 16

## U

**-u** option, **diff** command, 217

**undisplay** command, GDB, 154

**unified diff** command, 216

Unit testing, 6

**unset env** command, GDB, 155

**update** option, CVS, 185–86, 188

User manuals, 9

## V

**-V** option:

**ar** utility, 226

**ltrace** utility, 224

VA Linux, 16

Verbs, jCVS, 194–95

Version control systems, 10, 12–13

VIM, 37–42

    automatic indentation, 40–41

    basic concepts, 38

    basic editing, 38–42

        navigation, 38–39

    defined, 37

    folding code, 42

        commands, 39

**gvim**, 37

    indenting options, 40–41

    insert mode, 39–40

**:tags** command, 41–42

**:tselect** command, 41

    using tags with, 41–42

vim (vi) editor, 11

VxWorks, 245

## W

Windows CE, 245

## X

**-x** option, **ar** utility, 227

X-Windows, 13

Xemacs, 11, 32

**xxgdb** debugger, 135, 173–74

## Y

**-y** option, **diff** command, 217, 218