
Index

Numbers

- 32-bit mode, 8, 162
 - Also see* n32
 - compatibility, 155
 - data types, 160
 - definition, 155
 - libraries, 156
 - n32 definition, 155
 - overflow, 149
 - porting to n32, 157
- 32 option, 10, 23
- 4.3 BSD extensions, 27
- 64-bit mode, 8, 147-151
 - data types, 160
 - libraries, 156
- 64 option, 10, 23

A

- ABI
 - options, 136
- abi options, 8
- ABI specification, 8
- address aliases, 93-94
- addresses, optimization, 141
- address space, 77
- alias analysis, 93-94

- aliasing
 - and pointer placement, 139
 - memory, 124
 - optimization, 139
- analysis, dependence, 116-117
- analyzer, parallel, 2, 3
- ansi option, 23
- a.out files, 27
- architecture
 - instruction set, 136
 - optimizing programs, 136
 - options, 136
- archive libraries, 55
- archiver. *See* ar command
- ar command, 49-52
 - command syntax, 50
 - options, 51
- argument registers, 162
- arguments
 - store, 162
- arrays
 - 2 gigabyte, 163
- as assembler, 30
- assembly language programs
 - porting to n32, 158
- assembly language programs, linking, 30

B

- back substitution, 132

- bit masks, 151
- BLOCK DATA, 73
- blocking and permutation transformations, 111-114
 - controlling, 107
- block padding, 92-93
 - restrictions, 92
- branch elimination, 129
- BSD 4.3 extensions, 27
- Bsymbolic*, compiling, 72
- build procedure
 - n32, 158
- byte counters
 - and file size, 165

C

- C++
 - building DSOs, 67
 - language definitions, 15
 - ld* options, 67
 - precompiled headers, 15
- C++ programs
 - optimization, 139-143
- cache
 - conflicts and padding, 92
 - misses, 102
- cache optimization
 - LNO* option, 108
- cache parameters
 - controlling with LNO, 105
- CC compiler. *See* drivers
- cckr* option, 24
- char, 160
- C language
 - floating point, 117
 - precompiled headers, 15
- elist* option, 95

- code
 - arithmetic, 149
 - assumptions, 147
 - conversion, 128
 - hints, 147
 - overflow 32 bits, 149
 - portable, 150
 - porting to 64-bit system, 159
 - porting to n32-bit systems, 155
 - shifts, 149
 - signed ints, 143
 - `sizeof(int)==sizeof(long)`, 148
 - `sizeof(int)==sizeof(void*)`, 148
 - `sizeof(long)==4`, 148
 - `sizeof(void*)==4`, 149
 - transformation, 128
 - typedefs, 162
 - view transformations, 95
 - writing for 64-bit applications, 147-151
 - zero-extension, 143
- code generator, 126-136
 - Also see* optimizing programs
 - and optimization levels, 127, 128, 134
 - back substitution, 132
 - branch elimination, 129
 - cross-iteration optimization, 130-131
 - read-read elimination, 130
 - read-write elimination, 130
 - sub-expression elimination, 131
 - write-write elimination, 131
 - feedback, 136
 - frequency of execution, 136
 - if conversion, 128
 - if conversion and floating points, 129
 - instruction-level parallelism, 129
 - latency, 135
 - loop unrolling, 131, 134
 - memory exceptions, 129
 - modify default, 134
 - O0* option, 127
 - O1* option, 127

- code generator, (continued)
 - O2 option, 128-134
 - O3 option, 128-134
 - prefetch, 135
 - R10000 optimization, 129
 - recurrence breaking, 132
 - software pipelining, 133, 134
 - steps at -O2 and -O3, 133
 - COFF, 11
 - common block padding, 92-93
 - restrictions, 92
 - Common Object File Format, 11
 - COMMON symbols, 73
 - COMPILER_DEFAULTS_PATH environment
 - variable, 8
 - compiler back end, 2
 - compiler.defaults* file, 8
 - compiler drivers. *See* drivers, 2
 - compiler front end, 2
 - compiler options. *See* drivers
 - compiler system
 - 32-bit mode, 10
 - 64-bit mode, 10
 - components, 2
 - macros, 161
 - n32-bit mode, 10
 - overview, 2
 - predefined types, 161
 - compiling with *-Bsymbolic*, 72
 - constant format strings, 151
 - constants, 149
 - negative values, 151
 - conventions, syntax, xviii
 - conversion of code, 128
 - C option, 23
 - e option, 3, 23
 - copt* optimizer, 2
 - cord* option, 24
 - counters, internal byte, 165
 - cpp* preprocessor, 2
 - C programs
 - optimization, 139-143
 - cross-file inlining, 90
 - cross-iteration optimization, 130-131
 - read-read elimination, 130
 - read-write elimination, 130
 - sub-expression elimination, 131
 - write-write elimination, 131
- D**
- D__EXTENSIONS__ option, 27
 - D_MIPS_FPSET, 161
 - D_MIPS_ISA, 161
 - D_MIPS_SIM, 161
 - D_MIPS_SZINT, 161
 - D_MIPS_SZLONG, 161
 - D_MIPS_SZPTR, 161
 - data
 - prefetching, 102
 - data alignment
 - optimizing, 138
 - data type
 - signed, 142
 - data types
 - sizes, 160
 - debugging
 - driver options, 36
 - floating points, 122
 - defaults
 - compilation modes, 8
 - specification file, 8
 - dependence analysis, 109, 116-117
 - directives
 - LNO, 109

- disabling traps, 137
- disassemble object file 36
- dis* command, 36, 37
 - command syntax, 37
 - options, 37
- dlclose()*, 77
- dlderror()*, 76
- dlopen()*, 75
- dlsym()*, 76
- Dname* option, 24
- double, 160
- drivers
 - as* assembler, 30
 - bypassing, 2
 - CC compiler, 2
 - cc* compiler, 2
 - e* option, 3
 - defaults, 8, 22
 - f77/90* compiler, 2
 - fec* preprocessor, 2
 - file name suffixes 13
 - input file suffixes 13
 - KPIC*, 12
 - linking, 3
 - non_shared*, 12
 - omit linking, 3
 - optimizing programs, 118
 - options, 23, 36
 - options to *ld*, 28
 - show* option, 3
 - stages of compilation, 3
- DSOs, 1, 10, 11, 55-80
 - building new DSOs, 65
 - C++, 67
 - converting libraries, 73
 - creating DSOs, 65
 - dlclose()*, 77
 - dlderror()*, 76
 - dlopen()*, 75

- DSOs, (continued)
 - dlsym()*, 76
 - dynamic loading diagnostics, 76
 - exporting symbols, 66
 - guidelines, 58
 - hiding symbols, 66
 - libraries, shared, 58
 - linking, 32
 - loading dynamically, 75
 - mmap()* system call, 77
 - munmap()* system call, 77
 - naming conventions, 65
 - QuickStart, 62-64
 - QuickStart registry file 68
 - registry file 68-70
 - search path, 71
 - sgldladd()*, 76
 - shared libraries, 58
 - starting quickly, 62
 - unloading dynamically, 77
 - versioning, 77
- dump* command. *See elfdump*
- dwarfldump* command, 36, 38
 - options, 39
- DWARF symbolic information, 38
- dynamic linking, 2, 10, 75
- Dynamic Shared Objects. *See* DSOs

E

- elfdump* command, 36, 40
 - command syntax, 40
 - options, 40
- Elf object file 38
- ELF. *See* executable and linking format
- elimination
 - branches, 129
 - read-read, 130
 - read-write, 130

- elimination, (continued)
 - sub-expression, 131
 - write-write, 131
 - elspec* option, 24
 - environment
 - optimizing programs, 137
 - environment variable
 - COMPILER_DEFAULTS_PATH, 8
 - environment variables
 - 32-bit compilation, 10
 - 64-bit compilation, 10
 - n32-bit compilation, 10
 - E* option, 24
 - executable and linking format, 1, 10, 11
 - executable *fi les* 11
 - exporting symbols, 66
 - expressions
 - optimizing, 125
 - extension
 - sign, 150
 - zero, 150
- F**
- f77/90* compiler, 2
 - fec* preprocessor, 2
 - fec* preprocessor, 2
 - bypassing, 2
 - feedback
 - and code generator, 136
 - feedback* option, 24
 - fef77/90p* analyzer, 2, 3
 - fef77/90* preprocessor, 2
 - file* command, 36, 42
 - command syntax, 42
 - example, 43
 - options, 42
 - fi le inlining* 85-94
 - fi les*
 - 2 gigabyte size, 165
 - compilation specification, 8
 - executable, 11
 - header, 13
 - include, 13
 - internal byte counters, 165
 - listing properties, 36
 - naming conventions, 13
 - precompiled header, 15
 - relocatable, 11
 - size, 165
 - fi le type, determining* 42
 - fi ssion*
 - controlling, 104
 - LNO, 110
 - loops, 100
 - flist* option, 95
 - fl oat* 160
 - float.h* include *fi le* 160
 - floating points
 - debugging, 122
 - if conversion, 129
 - optimization, 117-122
 - optimizing, 125
 - reassociation, 125
 - Force, 121
 - format
 - object *fi le* 1, 10
 - Fortran
 - floating point 117
 - padding global arrays, 92
 - program optimization, 127
 - Fortran programs
 - optimization, 139-143
 - fullwarn* option, 24
 - functions
 - implicitly declared, 149

fusion
 controlling, 104
 LNO, 110
 loop, 99

G

gather-scatter, 102
 controlling, 105
global arrays
 padding, 92
global offset table, 12
global offset table overflow, 25
global optimizer, 138-143
-G option, 24
-g option, 24, 36
GOT, 12
GOT overflow, 25
guidelines
 porting, 157

H

header files 13-22
 multiple languages, 14
 portable, 150
 precompiled, 15
 specification 14
-help option, 24
high-order bit, 149

I

-dirname option, 25
IEEE
 floating points 119

IEEE, (continued)
 optimization, 119
if conversion, 128
if-then-else statements
 optimization, 142
implicitly declared function, 149
include files 13
 float.h 160
 inttypes.h, 162
 limits.h, 160
 multiple languages, 14
 n32, 158
indirect
 calls, using, 139
-INLINE , 89-90
 all option, 89
 file option, 90
 must option, 90
 never option, 90
 none option, 89
inliner
 standalone, 90
inlining, 85-94
 benefits 88
input file names 13
instruction
 mips4 recip, 126
 mips4 rsqrt, 126
 prefetching, 102
instruction-level parallelism, 129
int, 148, 160, 162
integer
 overflow, 125
 scaling, 150
interleaving
 reduction, 132
internal byte counters
 and file size 165

inttypes.h include file 162

-IPA, 91-94

addressing=ON option, 94

alias=ON option, 93

forceddepth option, 91

maxdepth option, 91

Olimit option, 91

opt_alias=ON option, 94

plimit option, 91

space option, 91

ISA

 options, 136

isa options, 8

ISA specification 8

K

-KPIC option, 12, 25

L

latency

 and code generator, 135

ld

 and assembly language programs, 30

 C++, 67

 command syntax, 28

 DSOs, 67

 dynamic linking, 2, 10

 example, 30

 libraries, default search path, 31

 libraries, specifying, 30

 link editor, 2

 multilanguage programs, 33

 options, 28, 67

 registry file 68

-shared option, 65

LD_BIND_NOW, 72

libdl, 75

libraries

 archive, 55

 global data, 60

 header file 13

libdl, 75

 locality, 60

 non-shared, converting to DSOs, 73

 paging, 60

 path, 9

 routines to exclude, 59

 routines to include, 59

 self-contained, 59

 shared, 1, 10

 shared, static, 12, 55

 specifying, 30

 static data, 59

 tuning, 60

lib.so functions

 optimization, 142

limits.h include file 160

linking

 dynamic. *See ld*

 omit, 3

linking. *See ld*

LNO. *See optimizing programs, -LNO* option

loader

 runtime. *See rld*

loading

 symbols, 66

local variables

 optimization, 139

long, 160, 162

long double, 160

long long, 160

loop interchange, 97-98

loop-nest optimization. *See optimizing programs, -LNO* option

loops
 blocking, 98
 fusion, 100
 fusion, 99
 interchanging, 97
 optimizing, 104
 parallel, 102
 unrolling, 98
loop unrolling
 code generator, 131

M

machine instructions, 36
macro preprocessors, 2
macros
 NARGSAVE, 162
 predefined, 161
 typedefs, 162
makefile, 158
maximum integer type, 163
memory
 2 gigabyte arrays, 163
 referencing, 124, 138
memory allocation
 arrays, 163
memory exceptions
 if conversion, 129
-mips1 option, 25
-mips2 option, 25
-mips3 option, 25
-mips4 option, 25
mips4 recip instruction, 126
mips4 rsqrt instruction, 126
MIPS Instruction Set Architecture, 161
mips options, 8
mmap() system call, 77

mode
 32-bit, 8
 64-bit, 8
 n32-bit, 8
modeling
 controlling, 107
-multiget option, 25, 29
multilanguage programs
 and *ld*, 33
 header files, 14
munmap() system call, 77

N

n32, 158
 assembly language programs, 158
 build procedure, 158
 include files, 158
 libraries, 156, 158
 porting environment, 158
 porting guidelines, 157
 runtime issues, 159
 source code changes, 158
n32-bit mode, 8
-n32 option, 10, 23
naming source files, 13
NARGSAVE macro, 162
negative values
 problems, 151
nm command, 36, 43-46
 character codes, 45
 command syntax, 43
 example, 46
 example of undefined symbols, 35
 options, 43
 undefined symbols, 35
-nocpp option, 25
-non_shared option, 25

`-nostdinc` option, 25

O

object file information

- disassemble, 36
- format, 1, 10
- listing file properties, 36
- listing section sizes, 36, 47
- symbol table information, 36, 43
- tools, 36
- using, 36
- using `dwarfdump`, 36
- using `elfdump`, 36, 40

`-o filename` option, 26

`-Onum` option, 26

operating system

- 64 bit, 147-151

operations

- relational, 125
- unsigned relational, 125

optimization, 83-143

- addresses, 141
- Also see* optimizing programs
- and register allocation, 143
- C++ programs, 139-143
- C programs, 139-143
- Fortran, 139-143
- function return values, 139
- global, 138-143
- if-then-else statements, 142
- `libc.so` functions, 142
- `-O0` compiler option, 85
- `-O1` compiler option, 85
- `-O2` compiler option, 85
- `-O3` compiler option, 85
- options, 85
- pointer placement, 139
- pointers, 140

optimization, (continued)

- signed data types, 142
 - STDARG, 141
 - `stdarg.h`, 141
 - subscripts, 140
 - switch statements, 142
 - tables, 142
 - tips for improving, 138
 - unions, 139
 - value parameters, 139
 - VARARG, 141
 - `varargs.h`, 141
 - variables, global vs. local, 139
- optimizer, 2
- `copt` optimizer, 2
- optimizing programs
- `-32` option, 136
 - `-64` option, 136
 - alias analysis, 93-94
 - `-align` option, 138
 - Also see* code generator
 - benefits 84
 - cache, 102
 - code generator, 126-136
 - overview, 126
 - common block padding, 92-93
 - restrictions, 92
 - data alignment, 138
 - debugging, 84
 - dependence analysis, 116-117
 - floating points 117-122
 - Fortran optimization, 127
 - IEEE floating points 119
 - ignoring pragmas, 104
 - `-INLINE` option, 89-90
 - inlining benefits 88
 - interprocedural analysis, 85-94
 - `-IPA` option, 91, 93
 - `-LNO` option, 94-117
 - blocking, 98-99

optimizing programs, (continued)

- blocking and permutation transformations, 107-108
- cache optimization, 108
- code transformation, 95
- controlling cache parameters, 105
- controlling dependence analysis, 109
- controlling fusion and fission, 104
- controlling gather-scatter, 105
- controlling illegal transformations, 108
- controlling prefetch, 108
- controlling transformations, 107
- directives, 109-117
- fission, 110
- fusion, 110
- gather-scatter, 102-103
- loop fission, 100-102
- loop fusion, 99-100
- loop interchange, 97-98
- optimization levels, 104
- outer loop unrolling, 98-99
- pragmas, 109-117
- prefetching, 102
- running LNO, 94
- `-mips` option, 136
- `-m32` option, 136
- `-OPT` option, 118-126
 - `alias=any` option, 124
 - `alias=name` option, 124
 - `alias=restrict` option, 125
 - `alias=typed` option, 124
 - `alias=unnamed` option, 124
 - `div_split`, 125
 - `div_split` option, 121
 - `fast_complex` option, 121
 - `fast_exp` option, 121
 - `fast_io` option, 121
 - `fast_sqrt` option, 121
 - `fold_reassociate`, 125
 - `fold_reassociate` option, 121
 - `fold_unsafe_relops`, 125

optimizing programs, (continued)

- `fold_unsigned_relops`, 125
- `IEEE_arithmetic` option, 119
- `IEEE` option, 118
- `recip`, 126
- `recip` option, 122
- `roundoff` option, 118
- `rsqrt`, 126
- `rsqrt` option, 122
- `space` option, 123
- pragmas, ignore, 104
- prefetch pragmas, 114-116
- shared code, 137
- target architecture, 136
- target architecture options, 136-137
- target environment, 137
- `-T ARG` option
 - `isa=mips` option, 136
 - `madd` option, 122, 136
- `-TENV` option, 137-138
 - `align_aggregates` option, 138
 - X option, 137
- transformation pragmas, 111-114
- transformations, 118
- `-OPT` option, 26
 - `div_split` option, 125
 - `fold_reassociate` option, 125
 - `fold_unsafe_relops`, 125
 - `fold_unsigned_relops` option, 125
 - `recip` option, 126
 - `rsqrt` option, 126
- overflow
 - integer, 125
 - integers, 125
- overflow of code, 149
- overflow of global offset table, 25

P

- padding, blocks, 92-93
 - restrictions, 92
- page size, 60
- paging
 - alignment, 60
- parallel analyzer, 2, 3
- parallel loops, 102
- parameters
 - optimization, 139
- pca* analyzer, 2, 3
- pc* compiler. *See* drivers
- pch* option, 26
- PIC. *See* position-independent code
- pixie*
 - and SpeedShop, 144
- pointer, 148, 160, 162
- pointer placement
 - and aliasing, 139
 - example, 139
- pointers
 - example, 140
 - optimization, 140
 - referencing memory, 124
- P* option, 26
- p* option, 26
- porting code, 159
- porting guidelines, 157
- position-independent code, 2, 10, 12, 65
- pragmas
 - ignore, 104
 - LNO, 109
- precompiled header files 15-21
 - automatic, 16
 - controlling, 20
 - deletion, 20
 - performance, 21
- precompiled header files (continued)
 - requirements, 17
 - reuse, 18
- prefetch
 - and code generator, 135
 - controlling, 108
- prefetching instructions, 102
- prefetch pragmas, 114-116
- preprocessing, 2
- preprocessors
 - macro, 2
- printf* command, 151
- problems, 149
 - constants, 149
 - floating points, 122
 - implicitly declared functions, 149
 - negative values, 151
 - porting code, 147
 - printf*, 151
 - scanf*, 151
 - `sizeof(int)==sizeof(long)`, 148
 - `sizeof(int)==sizeof(void*)`, 148
 - `sizeof(long)==4`, 148
 - solving, 150
 - types, 147
- processor specification 8
- proc options, 9
- prof*
 - and SpeedShop, 144

Q

QuickStart DSOs. *See* DSOs, QuickStart

R

-r10000 option, 137

- `-r5000` option, 137
- `-r8000` option, 137
- read-read elimination, 130
- read-write elimination, 130
- recip instruction, 126
- recurrence breaking
 - back substitution, 132
 - code generator, 132
 - reduction interleaving, 132
- reduction interleaving, 132
- registers
 - allocation, 143
 - argument, 162
 - blocking, 98
 - temp, 162
- registry file. *See* DSOs
- relational operations
 - unsigned, 125
- relational operators
 - integer overflow, 125
- relocatable files, 11
- relocation bits, removing, 36
- remove
 - relocation bits, 36
 - symbol table, 36
- resolve text symbols, 72
- return values, optimization, 139
- rld*, 56
 - dynamic linking, 75
 - libdl*, 75
 - search path, 71
- roundoff
 - floating points, 118
 - optimization, 118
- rsqrt instruction, 126
- runtime issues
 - n32, 159
- runtime linker. *See* *rld*

S

- scalar optimizer, *copt*, 2
- scalar variables
 - word size, 142
- scanf* function, 151
- search path
 - rld*, 71
- selecting
 - compilation mode, 8
 - instruction set, 8
 - ISA, 8
 - processor, 8
- sgidladd()*, 76
- shared code
 - optimizing, 137
- shared libraries, static, 55
- shared library, 1, 10
- shared objects, dynamic, 55
- short, 160
- `-show_defaults` option, 9
- `-show` option, 3, 26
- sign bit set, 149
- signed data type
 - optimization, 142
- signed ints
 - 64-bit code, 143
- sign extension, 148, 150
- size* command, 36, 47, 47-48
 - command syntax, 47
 - example, 48
- `sizeof(int)==sizeof(long)`, 148
- `sizeof(int)==sizeof(void*)`, 148
- `sizeof(long)==4`, 148
- `sizeof(void*)==4`, 149
- size of object files, 36

- software pipelining
 - and code generator, 133
 - S option, 26
 - source code
 - n32, 158
 - source file names 13
 - specifying compilation mode, 8
 - SpeedShop, 144
 - pixie* command, 144
 - prof* command, 144
 - ssrun* command, 144
 - standalone inliner, 90
 - stdarg.h*, 141
 - STDARG. *See* optimization
 - stdio.h* header file 14
 - storing arguments, 162
 - strings
 - printf*, 151
 - scanf*, 151
 - strip* command, 36, 48
 - command options, 49
 - command syntax, 48
 - sub-expression elimination, 131
 - subscripts
 - example, 140
 - optimization, 140
 - suffices
 - input files 13
 - switch statements
 - optimization, 142
 - symbol resolution, 72
 - symbols
 - exporting, 66
 - loading, 66
 - symbol table
 - data, 36
 - dumping data, 43
 - get listing, 46
 - symbol table, (continued)
 - removing, 36
 - syntax, conventions, xviii
- ## T
- TARG option, 26
 - temp registers, 162
 - TENV option, 26
 - transformation
 - of code, 128
 - transformation pragmas, 111-114
 - transformations
 - controlling illegal, 108
 - controlling with LNO, 107
 - view code, 95
 - traps
 - disable, 137
 - troubleshooting
 - constants, 149
 - implicitly declared functions, 149
 - negative values, 151
 - printf*, 151
 - scanf*, 151
 - `sizeof(int)==sizeof(long)`, 148
 - `sizeof(int)==sizeof(void*)`, 148
 - `sizeof(long)==4`, 148
 - `sizeof(void*)==4`, 149
 - solving problems, 150
 - truncation of code, 149
 - type, determining for files 42
 - typedefs, 151, 162
 - types
 - assumptions, 147
 - change in size, 149
 - char, 160
 - constants, 149
 - double, 160

types, (continued)
 float 160
 int, 148, 160, 162
 largest integer type, 163
 long, 160, 162
 long double, 160
 long long, 160
 pointer, 148, 160, 162
 problems, 147
 scaling integer, 150
 short, 160
 sizes, 160

typographical conventions, xviii

U

-Uname option, 27

unions

 optimization, 139

unsigned relational operations, 125

V

VARARG. *See* optimization

varargs.h, 141

variables

 scalar, 142

virtual address space, 77

W

-woff option, 27

word-size scalar variables, 142

write-write elimination, 131

X

-xansi option, 27

XFS

 file size 165

-xgot option. *See* *-multigot* option

Z

zero extension, 150

zero-extension code, 143