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, 8, 21
  See, 8
4.3 BSD extensions, 25
64-bit mode, 8, 147-151
  data types, 160
  libraries, 156
–64 option, 8, 21

A

addresses, optimization, 141
address space, 75
aliasing
  and pointer placement, 138
  memory, 125
  optimization, 138
analyzer, parallel, 2, 3
–ansi option, 21
a.out files, 25
architecture
  instruction set, 134
  optimizing programs, 134
archive libraries, 53
archiver. *See ar* command
ar command, 47-50
  command syntax, 48
  options, 49
argument registers, 162
arguments
  store, 162
arrays
  optimization, 127
as assembler, 28
assembly language programs
  porting to n32, 158
assembly language programs, linking, 28

B

basic block counting
  inclusive, 104
basic block counting. *See* profiling
bit masks, 151
BLOCK DATA, 71
BSD 4.3 extensions, 25
–Bsymbolic, compiling, 70
build procedure
  n32, 158

C

C++
  building DSOs, 65
  language definitions, 13
  ld options, 65
  precompiled headers, 13
cache mapping, improve, 111
CC compiler. *See* drivers
–cckr option, 22
char, 160
C language
  floating point, 119
  precompiled headers, 13
code
  arithmetic, 149
  assumptions, 147
  hints, 147
  overflow 32 bits, 149
  portable, 150
  porting to 64-bit system, 159
  porting to n32-bit systems, 155
  shifts, 149
  signed ints, 142
  sizeof(int)==sizeof(long), 148
  sizeof(int)==sizeof(void*), 147
  sizeof(long)==4, 148
  sizeof(void*)==4, 148
  typedefs, 162
  writing for 64-bit applications, 147-151
  zero-extension, 142
Common Object File Format, 9
COMMON symbols, 71
compiler back end, 2
compiler drivers. *See* drivers, 2
compiler front end, 2
compiler options. *See* drivers
compiler system
  32-bit mode, 8
  64-bit mode, 8
  components, 2
  macros, 161
  n32-bit mode, 8
  overview, 2
  predefined types, 161
compiling with *–Bsymbolic*, 70
constant format strings, 151
constants, 149
  negative values, 151
conventions, syntax, xviii
–C option, 21
–c option, 3, 21
copt optimizer, 2
cord, 111-114
  command options, 112
  command syntax, 111
  example, 112
  feedback files, 111
  –feedback option, 112
  –o out_file option, 112
  –v option, 112
–cord option, 22
.Counts file, 93
cpp preprocessor, 2

D

–D__EXTENSIONS__ option, 25
–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 alignment
  optimizing, 136

data type
  signed, 142
data types
  sizes, 160
debugging
  driver options, 34
  floating points, 124
disabling traps, 135
disassemble object file, 34
dis command, 34, 35
  command syntax, 35
  options, 35
dlclose(), 74
dlerror(), 74
dlopen(), 73
dlsym(), 74
–*Dname* option, 22
double, 160
do while, optimization, 138
drivers
  as assembler, 28
  bypassing, 2
  CC compiler, 2
  cc compiler, 2
  –*c* option, 3
  defaults, 20
  f77/90 compiler, 2
  fec preprocessor, 2
  file name suffixes, 11
  input file suffixes, 11
  –*KPIC*, 10
  linking, 3
  –*non_shared*, 10
  omit linking, 3
  optimizing programs, 120
  options, 21, 34
  options to *ld*, 26
  –*show* option, 3
  stages of compilation, 3
DSOs, 1, 9, 10, 53-78
  building new DSOs, 63
  C++, 65
  converting libraries, 72
  creating DSOs, 63
  dlclose(), 74
  dlerror(), 74
  dlopen(), 73
  dlsym(), 74
  dynamic loading diagnostics, 74
  exporting symbols, 64
  guidelines, 56
  hiding symbols, 64
  libraries, shared, 56
  linking, 30
  loading dynamically, 73
  mmap() system call, 75
  munmap() system call, 75
  naming conventions, 63
  QuickStart, 60-62
  QuickStart registry file, 66
  registry files, 66-68
  search path, 69
  sgidladd(), 74
  shared libraries, 56
  starting quickly, 60
  unloading dynamically, 74
  versioning, 75
dump command. *See elfdump*
dwarfdump command, 34, 36
  options, 37
DWARF symbolic information, 36
dynamic linking, 1, 9, 73
Dynamic Shared Objects. *See* DSOs

E

elfdump command, 34, 38
  command syntax, 38

options, 38
Elf object file, 36
ELF. *See* executable and linking format
–elspec option, 22
environment
optimizing programs, 135
environment variables
32-bit compilation, 8
64-bit compilation, 8
n32-bit compilation, 8
–E option, 22
executable and linking format, 1, 9
executable files, 9
exporting symbols, 64
extension
sign, 149
zero, 149

F

f77/90 compiler, 2
fecc preprocessor, 2
fec preprocessor, 2
bypassing, 2
–feedback option, 22
fef77/90p analyzer, 2, 3
fef77/90 preprocessor, 2
file command, 34, 40
command syntax, 40
example, 41
options, 40
files
executable, 9
header, 12
include, 12
listing properties, 34
naming conventions, 11
precompiled header, 13
relocatable, 9
file type, determining, 40
float, 160
float.h include file, 160
floating points
debugging, 124
optimization, 119-124
format
object file, 1, 9
Fortran
floating point, 119
optimization, 141
–fullwarn option, 22
functions
implicitly declared, 149

G

global code motion
optimization, 127
global offset table, 10
global optimizer, 137-142
–G option, 22
–g option, 22, 34
GOT, 10
-gprof, 105
example, 107
guidelines
porting, 157

H

header files, 12-20
multiple languages, 13
portable, 150

precompiled, 13
specification, 12
–*help* option, 22
high-order bit, 149

I

–*Idirname* option, 23
IEEE
floating points, 121
optimization, 121
if-then-else statements
optimization, 141
implicitly declared function, 149
include files, 12
float.h, 160
inttypes.h, 162
limits.h, 160
multiple languages, 13
n32, 158
inclusive basic block counting, 104
indirect
calls, using, 138
input file names, 11
int, 148, 160, 162
integer
scaling, 150
inttypes.h include file, 162

K

–*KPIC* option, 10, 23

L

ld
and assembly language programs, 28
C++, 65
command syntax, 26
DSOs, 65
dynamic linking, 1, 9
example, 28
libraries, default search path, 29
libraries, specifying, 28
link editor, 2
multilanguage programs, 31
options, 27, 65
registry files, 66
–*shared* option, 63
LD_BIND_NOW, 70
LD_LIBRARY_PATH, 93
libdl, 73
libraries
archive, 53
global data, 58
header files, 12
libdl, 73
locality, 58
non-shared, converting to DSOs, 72
paging, 58
routines to exclude, 57
routines to include, 57
self-contained, 57
shared, 1, 9
shared, static, 10, 53
specifying, 28
static data, 57
tuning, 58
lib.so functions
optimization, 142
limits.h include file, 160
linking
dynamic. *See ld*
omit, 3
linking. *See ld*

loader
  runtime. *See rld*
loading
  symbols, 64
local variables
  optimization, 138
long, 160, 162
long double, 160
long long, 160
loops
  inner, optimization, 129
  unrolling, 133

M

machine instructions, 34
macro preprocessors, 2
macros
  NARGSAVE, 162
  predefined, 161
  typedefs, 162
makefiles, 158
maximum integer type, 163
memory
  referencing, 125, 136
–mips1 option, 23
–mips2 option, 23
–mips3 option, 23
–mips4 option, 23
MIPS Instruction Set Architecture, 161
mmap() system call, 75
mode
  32-bit, 8
  64-bit, 8
  n32-bit, 8
mon.out file. *See* profiling, pc sampling
multilanguage programs
  and *ld*, 31
  header files, 13
multiprocessed executables, profiling, 110
munmap() system call, 75

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, 8, 21
naming source files, 11
NARGSAVE macro, 162
negative values
  problems, 151
nm command, 34, 41-44
  character codes, 43
  command syntax, 41
  example, 43
  example of undefined symbol, 33
  options, 41
  undefined symbol, 33
–nocpp option, 23
–non_shared option, 23
–nostdinc option, 23
nprof, 104-110
  command options, 105
  command syntax, 105
  example, 107

O

object file information
  disassemble, 34
  format, 1, 9
  listing file properties, 34
  listing section sizes, 34, 45
  symbol table information, 34, 41
  tools, 34
  using, 34
  using *dwarfdump*, 34
  using *elfdump*, 34, 38
–o filename option, 23
–Onum option, 24
operating system
  64 bit, 147-151
optimization, 117-143
  addresses, 141
  and register allocation, 143
  C, 138-142
  do while, 138
  Fortran, 138, 141
  function return values, 138
  global, 137-142
  if-then-else statements, 141
  libc.so functions, 142
  –O0 compiler option, 118
  –O1 compiler option, 118
  –O2 compiler option, 119
  –O3 compiler option, 119
  options, 118
  pointer placement, 138
  pointers, 140
  signed data types, 142
  STDARG, 141
  stdarg.h, 141
  subscripts, 140
  switch statements, 141
  tables, 141
  tips for improving, 137
  unions, 138
  value parameters, 138
  VARARG, 141
  varargs.h, 141
  variables, global vs. local, 138
optimizer, 2
  copt optimizer, 2
optimizing programs
  –32 option, 134
  –64 option, 134
  –align option, 136
  benefits, 117
  data alignment, 136
  debugging, 118
  floating points, 119-124
  –GCM
    aggressive_speculation option, 127
    array_speculation option, 127
    pointer_speculation option, 128
    postpass option, 128
    prepass option, 128
    static_load_speculation option, 128
  global code motion, 127
  IEEE floating points, 121
  –mips option, 134
  –n32 option, 134
  –OPT
    alias=name option, 125
    div_split option, 122
    fast_complex option, 123
    fast_exp option, 123
    fast_sqrt option, 123
    fold_reassociate option, 123
    IEEE_arithmetic option, 121
    IEEE_comparisons option, 123
    IEEE option, 120
    recip option, 123
    roundoff option, 120
    rsqrt option, 124
    space option, 125
  roundoff, 120

shared code, 135
software pipelining, 129
–*SWP*
back_substitution option, 129
backtracks option, 130
body_ins_count_max option, 130
fix_recurrences option, 132
if_conversion option, 132
interleave_reductions option, 133
trip_count_min option, 133
unroll_times_max option, 133
–*TARG*
isa=mips option, 134
madd option, 124, 134
processor option, 134
target architecture, 134
target environment, 135
–*TENV*
align_aggregates option, 136
large_GOT option, 135
small_GOT option, 135
X option, 135
TENV
misalignment option, 137
transformations, 120
–*OPT* option, 24
overflow of code, 149

P

page size, 58
paging
alignment, 58
paging, reduce, 111
parallel analyzer, 2, 3
parameters
optimization, 138
pca analyzer, 2, 3
pc compiler. *See* drivers
–*pch* option, 24
pc sampling. *See* profiling
PIC. *See* position-independent code
pixie, 91-104
and *-gprof*, 105
and *nprof*, 105
and *prof –clock* example, 101
and *prof –heavy* example, 97
and *prof –i* example, 96
and *prof –lines* example, 98
and *prof –pids*, 103
and *prof –procedures* example, 100
–*autopixie* option, 92
command option, 91
command syntax, 91
.Counts file, 93, 103
–*counts* option, 92
examples, 93
–*liblist* option, 92
output size, 94
–*pids* option, 92
restricting output, 94
setting search path, 93
–*verbose* option, 92
pointer, 148, 160, 162
pointer placement
and aliasing, 138
example, 138
pointers
example, 140
global code motion, 128
optimization, 140
referencing memory, 125
–*P* option, 24
porting code, 159
porting guidelines, 157
position-independent code, 1, 9, 10, 63
precompiled header files, 13-19
automatic, 14

controlling, 18
deletion, 18
performance, 19
requirements, 15
reuse, 16
preprocessing, 2
preprocessors
macro, 2
printf command, 151
problems, 148
constants, 149
floating points, 124
implicitly declared functions, 149
negative values, 151
porting code, 147
printf, 151
scanf, 151
sizeof(int)==sizeof(long), 148
sizeof(int)==sizeof(void*), 147
sizeof(long)==4, 148
solving, 150
types, 147
procedures, rearrange, 111
prof
Also see nprof.
Also see profiling
–clock example, 101
–heavy example, 97
–invocations example, 96
–lines example, 98
–procedures example, 100
PROFDIR environment variable, 89
profiling, 82-111
Also see nprof and *prof*
basic block counting, 91-104
–clock option, 83, 84
example, 101
command options, 83
command syntax, 83
–dis option, 84
–dso option, 83, 84
–exclude option, 83, 84, 87, 94
–g option, 94
–gprof option, 84
–heavy option, 83, 84
example, 97
–invocations option, 83, 84
example, 96
–lines option, 83, 84
example, 98
–merge option, 83, 85
multiple data files, 89
multiprocessed executables, 110
nprof, 104
–only option, 85, 87, 94
overview, 82
–pcsample option, 85
pc sampling, 86-89
example, 89
–pids option, 103
–pixie option, 85
–p option, 86
procedure invocation example, 96
–procedures option, 85
example, 100
–quit option, 85, 94, 99
summing results, 103
–testcoverage option, 85
–zero option, 85

Q

QuickStart DSOs. *See* DSOs, QuickStart

R

rearrange procedures, 111
reduce paging, 111
registers

allocation, 143
argument, 162
temp, 162
registry file. *See* DSOs
relocatable files, 9
relocation bits, removing, 34
remove
relocation bits, 34
symbol table, 34
reorder procedures, 111
resolve text symbols, 70
return values, optimization, 138
rld, 54
dynamic linking, 73
libdl, 73
search path, 69, 93
roundoff
floating points, 120
optimization, 120
runtime issues
n32, 159
runtime linker. *See rld*

S

scalar optimizer, *copt*, 2
scalar variables
word size, 142
scanf function, 151
search path
rld, 69, 93
selecting
compilation mode, 8
sgidladd(), 74
shared code
optimizing, 135
shared libraries, static, 53
shared library, 1, 9
shared objects, dynamic, 53
short, 160
–show option, 3, 24
sign bit set, 149
signed data type
optimization, 142
signed ints
64-bit code, 142
sign extension, 148, 149
size command, 34, 45, 45-46
command syntax, 45
example, 46
sizeof(int)==sizeof(long), 148
sizeof(int)==sizeof(void*), 148
sizeof(long)==4, 148
sizeof(void*)==4, 148
size of object file, 34
software pipelining, 129
enable/disable, 129
–S option, 24
source code
n32, 158
source file names, 11
stdarg.h, 141
STDARG. *See* optimization
stdio.h header file, 12
storing arguments, 162
strings
printf, 151
scanf, 151
strip command, 34, 46
command options, 47
command syntax, 46
subscripts
example, 140
optimization, 140

suffixes
  input files, 11
switch statements
  optimization, 141
–*SWP* option, 24
symbol resolution, 70
symbols
  exporting, 64
  loading, 64
symbol table
  data, 34
  dumping data, 41
  get listing, 43
  removing, 34
syntax, conventions, xviii

T

–*TARG* option, 24
temp registers, 162
–*TENV* option, 24
tools
  inclusive basic block counting, 104
  procedure rearranger, 111
traps
  disable, 135
troubleshooting
  constants, 149
  implicitly declared functions, 149
  negative values, 151
  printf, 151
  scanf, 151
  sizeof(int)==sizeof(long), 148
  sizeof(int)==sizeof(void*), 147
  sizeof(long)==4, 148
  sizeof(void*)==4, 148
  solving problems, 150
truncation of code, 149
type, determining for files, 40
typedefs, 151, 162
types
  assumptions, 147
  change in size, 149
  char, 160
  constants, 149
  double, 160
  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, 25
unions
  optimization, 138

V

VARARG. *See* optimization
varargs.h, 141
variables
  scalar, 142
virtual address space, 75

W

–*woff* option, 25
word-size scalar variables, 142

X

–*xansi* option, 25
–*xgot* option, 25

Z

zero extension, 149
zero-extension code, 142