Index

Symbols

. (period), 144

!, 48

?:, 49–50, 81

&, 42
  not allowed with parallel-left-indexed parallel variable, 136

&&, 45, 81

&=, 57

%, 51

%%, 51–52

++, 48

–=, 55

***=**, 56

/=, 56

^=, 57

||, 48, 81

|=, 57, 78

<?, 50–51

<?=, 51–66

>?, 50–51

>?=, 51, 56

>=, 48

Numbers

64-bit integers, 60
  and communication functions, 63
  and I/O functions, 64
  and library functions, 63
  and mathematical functions, 64
  and string functions, 64
  declaring, 60
  hints on using, 61
  performance using, 62
  semantics of, 61

A

active positions, 11, 67
  See also positions
  and scan sets, 187
  obtaining the number of, 116
  using cast to obtain number of, 116
  when shape first selected, 67
  when there are no, 76–79

allocate_detailed_shape, 245
  CM-5 version, using, 259

allocate_shape, 112–113, 216

ANSI, 3

arrays
  See also parallel arrays
  and parallel structures, 30
  and pointers, 89

arrays of shapes, 119
  and pointers, 109–120
  partially specifying, 108–109

atoll, 64

axis, 19, 154
  serial, 254
  weighting, 255

axis sequence, 249

axis_mask, 211, 242

Copyright © 1990–1995 Thinking Machines Corporation

B

<bitop.h>, 307
bitwise AND, 57, 180
bitwise exclusive OR, 57, 180
bitwise OR, 57, 81, 180
  used to prevent code from executing, 78
bitwise reduction operators, 56–58
block scope, branching into, 22, 28
bools, 58, 115
boolsizeof, 59, 230
border behavior, 155
  and **pcoord**, 144
break, 40
  and **everywhere**, 76
  behavior in nested **where** statement, 74

C

C operators
  with scalar and parallel operands, 44–47
  with scalar LHS and parallel RHS, 45–47
  with scalar operands, 43–44
  with two parallel operands, 47–48
C*
  and C, 3–4
  program development facilities of, 4
C* program
  compiling, 15
  executing, 16
casts, 116–118
  parallel-to-scalar, 46, 118
  scalar-to-parallel, 116
  to a different shape, 117
CMC_allocate_shared_table, 279
CMC_change_pointer_shape, 273
CMC_combiner_add, 180
CMC_combiner_copy, 180
CMC_combiner_logand, 180
CMC_combiner_logior, 180
CMC_combiner_logxor, 180
CMC_combiner_max, 180
CMC_combiner_min, 180
CMC_combiner_multiply, 180
CMC_combiner_overwrite, 225, 229
CMC_combiner_t, 191
CMC_communication_direction_t, 191
CMC_downward, 188
CMC_exclusive, 185
CMC_free_shared_table, 280
CMC_global_to_local, 297
CMC_global_to_physical_index, 298
CMC_inclusive, 185
CMC_initialize_shared_table, 279
CMC_local_to_global, 294
CMC_lookup_shared_table, 280
CMC_make_pointer, 273
CMC_no_field, 191, 229
CMC_none, 191
CMC_pointer_mem_addr, 273
CMC_pointer_mem_stride, 273
CMC_scan_inclusion_t, 191
CMC_segment_bit, 186
CMC_segment_mode_t, 191
CMC_sendaddr_t, 217
CMC_shape_axis_ordering, 259
CMC_shape_physical_axis_mask, 259
CMC_shape_subgrid_axis_increment, 259
CMC_shape_subgrid_axis_length, 259
CMC_shape_subgrid_axis_outer_count, 259
CMC_shape_subgrid_axis_outer_increment, 259
CMC_shape_subgrid_axis_sequence, 259

Version 7.2, May 1995
Copyright © 1990–1995 Thinking Machines Corporation

CMC_shape_subgrid_orthogonal_length, 259

CMC_shape_subgrid_size, 259, 270

CMC_start_bit, 186

CMC_upward, 188

CMMD, and global/local programming, 283

cmmd_cswrapper_gen, 294

collision_mode, 222

collisions, 128
  in get operations, 221–222
  with parallel left indexing, 127–130

combiner, types of, 179–213

conditional expression, 49–50

conditional operator, 81

context, 67
  See also **where**
  effect on other contexts, 73
  resetting, 69, 74

continue, 40
  and **everywhere**, 76
  behavior in nested **where** statement, 74

coordinates, 22, 81, 151, 153

copy_multispread, 182, 212, 240–243

copy_reduce, 197–199

copy_spread, 200–203, 240

.cs, 8

cs
  -local option, 299
  -noansilibs option, 63
  -noprintflibs option, 65

<cscomm.h>, 150

<csgl.h>, 294

<csshape.h>, 258, 260, 272

<cstable.h>, 279

current, 95, 101–102

current shape, 11, 37, 67, 100
  and pointers, 87, 88

D

deallocate_shape, 113–114

demotion, parallel-to-scalar, 46

dimensions, 107
  maximum number of, 113
  partially specifying, 109

dimof, 23, 33, 109, 145
  and **pcoord**, 145–146

dimulh, 309

direction. *See* upward direction, downward direction

downward direction, 190
  and scan sets, 188

dumulh, 309

E

elements, 7, 9, 24, 122
  and positions, 27–28
  choosing, 180–190
  corresponding, 27, 47
  operations on, 71
  sorting by rank, 230–232

else clause, 69–70

enumerate, 203–205

enums, parallel, 66

everywhere, 75–76
  in functions, 99

exclusive operation, 185

extern, and shapes, 110

F

ffb instruction, 309

fgetpos, 63

from_grid, 163–165

from_grid_dim, 156–163

Copyright © 1990–1995 Thinking Machines Corporation

from_torus, 170–173

from_torus_dim, 170–173

fsetpos, 63

function prototyping, 96

functions
  and shapes, 100–101
  as shape-valued expressions, 101
  intrinsic, 22
  overloading, 104–105
  passing by reference, 98
  using parallel variables with, 95–98

G

garbage positions, 246
  effect on performance, 258

general communication, 151

get function, 219–224
  and parallel structures or parallel arrays, 222
  collisions in, 222

get operation, 123–124, 220
  and collisions, 127–128, 221–222
  in functions, 99, 133
  inactive positions in, 130–131

global, 212–213, 215

global/local program
  compiling and linking, 299
  debugging, 300
  flow control in, 286
  sample, 302
  writing, 291

global/local programming, 283
  global and local data in, 288
  I/O in, 290
  inquiry functions in, 294
  limitations of, 286
  overview of, 284
  prototype files in, 293
  shape and context in, 287

goto, 40
  and **everywhere**, 76
  behavior in nested **where** statement, 74
  branching into block containing shape declaration, 22
  branching into block with parallel variable declaration, 28

grid communication, 150, 151, 179
  and inactive positions, 155–156
  and **pcoord**, 143–146
  aspects of, 153–156
  direction of, 154
  distance of, 155

H

hyperplane, 210, 240

I

if, 57, 78

inactive positions, 72
  See also positions
  and parallel left indexing, 130–134
  and scan sets, 187
  and send operations, 226
  behavior in grid communication, 155–156

index variable, use of, 125–126

initializing, using parallel variables, 41

integer inquiry functions, 307

integers, 64-bit. *See* 64-bit integers

L

labs, 63

ldiv, 63

leadz, 308

leadz_nz, 308

left index, 34, 42
  and scalar variables, 34
  parallel, 122–139
    and **pcoord**, 143
    limitations of, 136
    what can be indexed, 136

Copyright © 1990–1995 Thinking Machines Corporation

precedence of, 35

llabs, 64

lldiv, 64

local shape, assigning to a global shape, 111–112

logical AND operator, 45, 81

logical OR operator, 81

long long int. *See* 64-bit integers

looping through all positions, 79

M

make, 5

make_multi_coord, 240–243

make_send_address, 215–219, 230, 233, 237

matrix
  multiplying diagonals in, 137–146
  transposing, 142–144

maximum operator, 50–51

maximum reduction operator, 56

memory stride, 271, 273

minimum operator, 50–51

minimum reduction operator, 56

modulus operator, 51–52

multicoordinate, 240
  obtaining, 241

multispread, 182, 208–211

N

notify bit, 225, 226, 229

O

overload, 105

overloading, 95, 104–105

P

palloc, 107, 114–116

parallel arrays
  declaring, 30–31
  elements of, 31
  getting, 222
  initializing, 32
  parallel indexes into, 90–93
  sending, 228–230

parallel right indexing, 90

parallel structures
  declaring, 28–30
  getting, 222
  initializing, 32
  sending, 228–230

parallel unions. *See* unions, parallel

parallel variables, 9
  allocating storage for, 115–116
  choosing an individual element of, 13–14, 34
  communicating with scalar variables, 232
  compared with scalar, 24–25
  declaring, 25–28
  declaring multiple, 26–28
  declaring with a shape-valued expression, 118–119
  initializing, 32–35, 41, 42
  mapping to another shape, 134–136
  memory layout in CM-5 C*, 269
  not of current shape, 42
  obtaining information about, 32–33
  passing as argument to function, 95–96
  returning from function, 97–98
  scope of, 28
  sending, 224–228
  unary operators for, 48

parallel-to-scalar assignment, 46
  when no positions are active, 77

passing by value, 98

pcoord, 81, 139–143
  and **enumerate**, 203
  and grid communication, 143–146

Copyright © 1990–1995 Thinking Machines Corporation

pfree, 115

physical, 119

physical grids, 246, 262
  effect on performance, 256

pointer arithmetic, 89–90

pointers
  scalar-to-parallel, CM-5 C*
    representation of, 271
  scalar-to-parallel, 86–88
    adding a parallel variable to, 92–93
    and parallel structures, 30
    as arguments to a function, 96
  scalar-to-scalar, 85
  to shapes, 86

popcnt, 310

poppar, 311

positions, 9
  See also active positions, inactive positions
  and elements, 27–28
  definition of, 19
  looping through all, 79

positionsof, 22, 33, 109
  and **where**, 70

Prism, 5
  and global/local programs, 300

promotion, **bool** to **int**, 59

promotion, scalar to parallel, 44, 45, 116, 127

prototype files, in global/local programming, 293

R

rank, 19, 107, 112, 149
  sorting elements by, 230–232

rank function, 205–208, 230

rankof, 23, 33
  and a partially specified shape, 108
  and fully unspecified shape, 108

read_from_position, 232–234

read_from_pvar, 234

reduce, 195–197

reduction assignment, 14
  and global, 212
  parallel-to-parallel, 54
  parallel-to-scalar, 52
  when no positions are active, 77–78
  with a parallel LHS, 58
  with send operation, 129

reduction operators, 52–58
  list of, 54–55
  precedence of, 58
  unary, 53–54

return, 40
  and **everywhere**, 76
  behavior in nested **where** statement, 74

S

sbit, 183, 186, 187, 194

scalar variables, 10
  communicating with parallel variables, 232
  contrasted with ANSI definition, 24
  in left index, 34
  promoted to parallel, 44

scan, 180, 190–195
  difference from **reduce**, 196

scan class, 180–213
  subset of hyperplane, 210

scan set, 183–185

scan subclass, 183, 194

scan subset, 194

scope
  of parallel variables, 28
  of shapes, 22–24

segment bit, 186, 187

send address, 151, 153, 215
  obtaining a single, 216–217
  obtaining more than one, 217–219

send function, 224, 230
  and parallel arrays or parallel structures, 228–230
  and parallel variables, 224–228

Copyright © 1990–1995 Thinking Machines Corporation

differences from send operation, 226

send operation, 124–125
and collisions, 128–130
and **send** function, 226
comparing parallel left indexing and send, 226
in functions, 99, 133
inactive positions in, 131–133
with parallel left indexing, 225

serial axes, 254
effect on performance, 257

shape aliasing, 273

shape names, predeclared, 101, 119

shape selection, 11

shape-valued expression, 33, 39
declaring parallel variable with, 118–119
in casts, 118

shapeof, 33
used with **void** shape, 103
using to access a pointer's shape, 272

shapes, 8
See also current shape
as arguments to functions, 100
choosing, 20
creating copies of, 110, 114
deallocating, 113–114
declaring, 21–22
declaring multiple, 22
default, 39
default layout on CM-5, 246
definition of, 19
dynamically allocating, 112–113
equivalence of, 110–111
fully unspecified, 107–108
maximum number of dimensions in, 19
not allowed in structures, 30
obtaining information about, 22–23
partially specified, 107–110
returned by functions, 101
scope of, 22
switching between, 74

smode, 186

spread, 199–200, 208

start bit, 186

<stdlib.h>, 113, 115

strtoll, 64

strtoul, 63

strtoull, 64

structures. *See* parallel structures

subgrid layout, controlling, 253

subgrid length, effect on performance, 256

subgrid sequence, 250

subgrid-orthogonal-length, 257

subgrids, 248

switch
branching into block containing shape declaration, 22
branching into block with parallel variable declaration, 28

T

table lookup utility, 279

to_grid, 165

to_grid_dim, 165

to_torus, 173–177

to_torus_dim, 173–177

torus, 170

U

unary operators and parallel variables, 48

unions, parallel, 65

upward direction, and scan sets, 188

V

variables. *See* parallel variables, scalar variables

Copyright © 1990–1995 Thinking Machines Corporation

void predeclared shape name, 95, 102–104
  used when returning a pointer, 104

W

where, 67–71, 145
  and parallel-to-scalar assignment, 71
  and **positionsof**, 70
  and scalar code, 71–72
  controlling expression of, 68
  nesting, 72–73

while, 81

with, 11, 37–39, 67, 233, 237
  nesting, 39–41, 73–74
  using a shape-valued expression with, 39

wrapping, 155

write_to_position, 236–238

write_to_pvar, 238

Version 7.2, May 1995
Copyright © 1990–1995 Thinking Machines Corporation