Index

Symbols

%VAL operator, 116
  example, 116

A

adjustable arrays, 81
ALIGN compiler directive, 210
ALL intrinsic function, 134
allocatable arrays, example, 112
ALLOCATABLE attribute, 112
ALLOCATE statement, 109
ALLOCATED intrinsic function, 112
ANY intrinsic function, 134
array arguments, 72
  adjustable dummies, 81
  array descriptors, 73
  array sections, 88
  assumed-layout dummies, 83
  assumed-shape dummies, 81
  assumed-size dummies, 82
  CM arrays, 73, 131
  controlling homes of, 75
  controlling in/out copying, 92
  explicit-shape dummies, 81
  front-end arrays, 73
  homes of, 74
  layouts of, 79
  reshaping, 77
  retrieving properties of, 100
  shapes of, 77
  to intrinsic functions, 131
  types of, 77
  with all scalar indices, 73, 78
  with elided axes, 78
ARRAY attribute, 46
array constructors
  and **DATA** attribute, 54
  as CM array arguments, 53, 139
  forms of, 53
array homes
  argument arrays, 74
  assumed-layout arrays, 83
    example, 84
  assumed-shape arrays, 81
  assumed-size arrays, 82
  automatic arrays, 98
  character arrays, 19
  common arrays, 103
  controlled with **COMMON**, 104
  controlled with **LAYOUT**, 75, 104
  error checking with **-safety**, 74
  local arrays, 22
  mismatched arguments, 74
  mixed-home operations, 23, 156
  not affected by **FORALL**, 156
  not affected by initialization, 55
  verifying with **-list**, 76
array layout, 185
  -noaxisreorder switch, 191, 198
  -nopadding switch, 187
  axis sequence, 191
  axis weights, 199
  canonical, 186
  garbage elements, 186, 202
  LAYOUT compiler directive, 196
  machine arrays, 187
  NEWS ordering, 197
  padding, 186, 202
  physical grids, 186, 200
  processor masks, 205
  SEND ordering, 197

Copyright © 1989–1994 Thinking Machines Corporation

array layout *(cont'd)*
serial axes, 197, 201
subgrid sequence, 192
subgrids, 189, 200
array object, defined, 5
array operations, defined, 20
array processing, defined, 3
array sections
and communication, 68
as arguments. *See* array arguments
conformable, 68, 124
default forms, 67
negative strides, 65
of multidimensional parents, 66
parent arrays, 65
rank of, 67, 123
scalar and triplet subscripts, 67
scalar and vector-valued subscripts, 123
triplet subscripts, 65
vector-valued subscripts, 121
array transfer utilities, 24
arrays
allocatable, example, 112
as arguments. *See* array arguments
automatic, 98, 102
example, 99
common. *See* common arrays
deferred-shape, 108
rank of, 46
reshaping, 77, 140
retrieving properties of, 100
shape of, 46
vector-valued subscripts, 121
arrays, conformable. *See* conformable arrays
ASSOCIATED intrinsic function, 110
automatic arrays, 98

C

calling a local routine, 237
CM processors, 11
cmf command. *See* compiler command **cmf**
CMF_ALLOCATE_ARRAY utility, 113
example, 114
CMF_DEALLOCATE_ARRAY utility, 113
example, 114
CMF_FE_ARRAY_FROM_CM utility, 24
CMF_FE_ARRAY_TO_CM utility, 24
CMF_RANDOM utility, 51
CMF_RANDOM_LONG_S_INTEGER utility, 51
CMGL_BROADCAST_SERIAL_ARRAY function, 223
CMGL_global_to_local subroutine, 229
CMGL_global_to_physical subroutine, 231
CMGL_local_to_global subroutine, 225
cmld linker, 27
common arrays
declaring, 104
homes of, 103
initializing, 105
COMMON compiler directive, 104
example, 106
COMMON statement, 104
communication mechanisms, 158
compiler command **cmf**
switches
-common_initialized, 106
-double_precision, 48
-fecommon, 105
-list, 76
-noaxisreorder, 191
-nodirectives, 105
-nopadding, 33, 188
-safety, 74
using, 26
compiler directives
ALIGN, 210
COMMON, 104
LAYOUT, 75, 196
compiling a global/local program, 242
computations on arrays, defined, 7

Version 2.2, October 1994
Copyright © 1989–1994 Thinking Machines Corporation

conditionals
See also **WHERE**
front end controlling CM, 63
example, 64
conformable arrays
defined, 20
in masked assignments, 58
layout in CM memory, 20
constants
defining, 48
typed literal, 49
control constructs
from Fortran 90, 13
front end controlling CM, 63
convolution example, 176
Fortran 77, 174
COUNT intrinsic function, 134
CSHIFT intrinsic function, 126

D

DATA attribute, 54
with common arrays, 106
DATA statement, 52
with common arrays, 106
data types
See also declarations
precision of, 47
supported in CM Fortran, 19
DEALLOCATE statement, 109
debugging, global/local programs, 244
declarations
attributed, 46
function calls in, 55, 102
of lower bounds, 47
with initialization, 54, 105
deferred-shape arrays, 108
dependences, in array operations, 6
descriptors, of arrays, 72
DIAGONAL intrinsic function, 139
DIMENSION attribute, 46
directives, compiler. *See* compiler directives by name
DOT_PRODUCT intrinsic function, 144

E

elemental array assignment, 147
elemental operations, defined, 17
EOSHIFT intrinsic function, 128
examples
%VAL, 116
allocatable arrays, 112
array-valued function, 96
assumed-layout arrays, 84
automatic arrays, 99
CMF_ALLOCATE_ARRAY, 114
CMF_DEALLOCATE_ARRAY, 114
COMMON directive, 106
convolution, 176
Fortran 77, 174
FMALLOC, 116
front-end conditionals, 64
histogram, 173
Fortran 77, 172
INTENT statement, 93
Laplace solver, 181
passing array as argument, 89
passing array sections as arguments, 90
prime number sieve, 178, 179
Fortran 77, 177
simple program, 18
execution models
defined, 29
global, 29, 30
CM-5 SPARC nodes, 31
CM-5 vector-units, 32
global/local, 29, 39
nodal, 29, 34
CM-5 nodal (hostless), 34
CM-5 nodal (host/node), 37
simulator, 29, 41

F

FFREE procedure in cmf77 library, 116
FIRSTLOC intrinsic function, 137
FMALLOC procedure in cmf77 library, 116
example, 116

Copyright © 1989–1994 Thinking Machines Corporation

FORALL statement, 147
as-if-simultaneous execution, 154
assignments in, 149
data movement with, 158
mask expression in, 153
masked intrinsic functions in, 154
parallel prefix operations with, 163
restrictions on, 149, 157
send operations with, 168
serial vs. parallel execution of, 156
spread operations with, 152
syntax of, 148
foreign Fortran compilers, 27, 103
Fortran 77, and CM Fortran, 4, 11
Fortran 90, and CM Fortran, 77, 147
front-end processor, 11
functions
See also intrinsic functions, and functions by name
array-valued, 95
example, 96
scalar-valued, 95
statement, 95
user-defined, 95

G

garbage data, in CM arrays, 33
global communication, see communication, 9
global execution model, 29, 30
CM-5 SPARC nodes, 31
CM-5 vector-units, 32
global program, how to write, 236
global/local execution model, 29, 39
global/local library
CMGL_BROADCAST_SERIAL_ARRAY, 223
CMGL_global_to_local, 229
CMGL_global_to_physical, 231
CMGL_local_to_global, 225
global/local programming
arrays, 217
parallel, 217
serial, 219
calling a local routine, 237
compiling, 242
data, 217
debugging, 244
flow of control, 216
header files, 243
library routines, 222
linking, 242
performance, 221
profiling not supported, 245
program construction, 236
prototype file, 241
restrictions, 221
writing a local routine in C, 239
writing a local routine in CM Fortran, 237
writing the global program, 236

H

header files, for global programs, 243
histogram example, 173
Fortran 77, 172
homes of arrays. *See* array homes

I

I/O
none from local routines, 238, 240
statements, 25
IMPLICIT statement, 19
initialization expressions, 47
INTENT attribute, 92, 111
INTENT statement, 92
examples, 93
interface blocks
and argument homes, 85
and argument layouts, 85
and argument shapes, 85
with contiguous array sections, 91
with **INTENT** attribute, 92
with **LAYOUT** directive, 85
with noncanonical array sections, 90
intrinsic functions
argument keywords in, 127

Version 2.2, October 1994
Copyright © 1989–1994 Thinking Machines Corporation

array construction, 139
DIAGONAL, 139
MERGE, 142
PACK, 139
REPLICATE, 140
RESHAPE, 140
SPREAD, 143
UNPACK, 139
array inquiry, 100
LBOUND, 101
SHAPE, 100
SIZE, 100
UBOUND, 101
array location, 135
FIRSTLOC, 137
LASTLOC, 137
MAXLOC, 135
MINLOC, 135
array manipulation, 126
CSHIFT, 126
EOSHIFT, 128
TRANSPOSE, 128
array reduction, 132
ALL, 134
ANY, 134
COUNT, 134
MAXVAL, 132
elemental, 20
inquiry
ALLOCATED, 112
ASSOCIATED, 110
matrix multiplication, 144
MATMUL, 144
reduction, 132
transformational, 131
vector multiplication, 144
DOT_PRODUCT, 144
MATMUL, 144
irregular data, **FORALL** operations on, 161
iteration, with front-end control construct, 63

K

keywords, in intrinsic function calls, 127
kind types, 47
parameters for, 47

L

Laplace solver example, 181
LASTLOC intrinsic function, 137
LAYOUT compiler directive, 75, 196
See also array layout
LBOUND intrinsic function, 101
linking, global/local programs, 242
local routines
calling, 237
how to write in C, 239
how to write in CM Fortran, 237

M

masked array assignment
using **FORALL**, 153
using **WHERE**, 58
MATMUL intrinsic function, 144
MAXLOC intrinsic function, 135
MAXVAL intrinsic function, 132
MERGE intrinsic function, 142
MINLOC intrinsic function, 135

N

NEWS communication, 9
nodal execution model, 29, 34
CM-5 nodal (hostless), 34
CM-5 nodal (host/node), 37
nodal program, 214

P

PACK intrinsic function, 139
padding in CM arrays, 33
parallel prefix operations, using **FORALL**, 163
PARAMETER attribute, 48
PARAMETER statement, 48
parent arrays. *See* array sections

Copyright © 1989–1994 Thinking Machines Corporation

permuting array elements
using array sections, 121
using **FORALL**, 160
using vector-valued subscripts, 124, 159
POINTER attribute, 108
POINTER statement, 108
pointers, assignment statement, 108
precision, of numeric types, 47
prime number sieve example, 178, 179
Fortran 77, 177
procedures, 71
profiling, global/local programs (not supported), 245
prototype file, 241

R

random number generation, 51
REPLICATE intrinsic function, 140
replicating array elements, using vector-valued subscripts, 125
RESHAPE intrinsic function, 140
reshaping arrays, 77, 140
restrictions, global/local programming, 221
router communication, see communication, 9

S

SAVE attribute, 110
scalars, in array operations, 22
scan operations, 163
sections. *See* array sections
send-with-combiner operations, 168
using **FORALL**, 168
sequence association, 77
SHAPE intrinsic function, 100
shifting array elements
using array sections, 118
using **FORALL**, 158
using intrinsic functions, 126
simulator, 29, 41
SIZE intrinsic function, 100
SPREAD intrinsic function, 143
spreading array elements
using **FORALL**, 152, 160
using **SPREAD**, 143
statement functions, 95
subprograms, 71

T

TARGET attribute, 108
transformations. *See* intrinsic functions
TRANSPOSE intrinsic function, 128
triplet notation. *See* triplet subscripts
typed constants, 49

U

UBOUND intrinsic function, 101
UNPACK intrinsic function, 139
utilities, dynamic array allocation
CMF_ALLOCATE_ARRAY, example, 114
CMF_DEALLOCATE_ARRAY, example, 114

V

vector-length padding, 33
vector-valued subscripts, 121

W

WHERE construct, 59
nested, 60
WHERE statement, 58
writing a local routine in C, 239
writing a local routine in CM Fortran, 237

Version 2.2, October 1994
Copyright © 1989–1994 Thinking Machines Corporation