



INMOS Dx314 ANSI C Toolset Master Index

INMOS Limited



INMOS is a member of the SGS-THOMSON Microelectronics Group

Master index

This master index covers five manuals belonging to the Toolset Documentation set; the notation used to refer to individual documents is as follows:

User Guide indicates the 'ANSI C Toolset User Guide' 72-TDS-345-01.

Tools Reference indicates the 'ANSI C Toolset Reference Manual' 72-TDS-346-01.

Language Reference indicates the 'ANSI C Language and Libraries Reference Manual' 72-TDS-347-01.

Optimizing Compiler Guide indicates the 'ANSI C Optimizing Compiler User Guide' 72-TDS-348-01.

Performance Note indicates the document: 'Performance Improvement with the Dx314 ANSI C Toolset' 72-TDS-354-00.

Symbols

..., ellipsis. See Ellipsis

!, **idebug**: *Tools Reference* 135, 138, 145, 159

::, **idebug**: *Tools Reference* 149

idebug: *Tools Reference* 119
idump: *Tools Reference* 175
isim: *Tools Reference* 306

#alias: *Tools Reference* 220

#define
linker directive: *Tools Reference* 221
syntax: *Tools Reference* 12

#elif: *Language Reference* 380, 384
syntax: *Tools Reference* 12

#else: *Tools Reference* 13
syntax: *Tools Reference* 12

#endif: *Tools Reference* 13
syntax: *Tools Reference* 12

#error: *Language Reference* 380, 385

syntax: *Tools Reference* 13

#if, syntax: *Tools Reference* 13

#ifdef, syntax: *Tools Reference* 13

#ifndef, syntax: *Tools Reference* 13

#include
filename syntax: *Tools Reference* 14

icc directive: *Tools Reference* 14

linker directive: *Tools Reference* 221

nesting **icc** directives: *Tools Reference* 14

#line, syntax: *Tools Reference* 14


#mainentry: *Tools Reference* 221

#PRAGMA


EXTERNAL: *User Guide* 198

LINKAGE: *Tools Reference* 222

© INMOS Limited 1992. This document may not be copied, in whole or in part, without prior written consent of INMOS.

 **inmos**[®], IMS and occam are trademarks of INMOS Limited.

INMOS Limited is a member of the SGS-THOMSON Microelectronics Group.

 is a registered trademark of the SGS-THOMSON Microelectronics Group.

The C compiler implementation was developed from the Perihelion Software "C" Compiler and the Codemist Norcroft "C" Compiler.

INMOS Document Number: 72 TDS 360 00

TRANSLATE: *User Guide* 199

#pragma: *Language Reference* 380, 385, 387; *Optimizing Compiler Guide* 8

IMS_codepatchsize: *Tools Reference* 15; *Language Reference* 388

IMS_descriptor: *Tools Reference* 15, 17; *Language Reference* 388
for dynamic code loading: *User Guide* 235, 238
parameters: *Tools Reference* 18

IMS_linkage: *Tools Reference* 15, 222; *Language Reference* 388; *Performance note* 8

IMS_modpatchsize: *Tools Reference* 15; *Language Reference* 388

IMS_nolink: *User Guide* 202; *Tools Reference* 15, 17; *Language Reference* 359, 388, 407

IMS_nosideeffects: *Language Reference* 388; *Optimizing Compiler Guide* 8

IMS_off: *Tools Reference* 15; *Language Reference* 388
parameters: *Tools Reference* 16

IMS_on: *Tools Reference* 15; *Language Reference* 388
parameters: *Tools Reference* 16

IMS_translate: *User Guide* 199; *Tools Reference* 15; *Language Reference* 388
introduction: *User Guide* 12
syntax: *Tools Reference* 14; *Language Reference* 413

#reference: *Tools Reference* 221

#section: *Tools Reference* 222; *Performance note* 8

#undef, syntax: *Tools Reference* 19

\$
idebug: *Tools Reference* 119
idump: *Tools Reference* 175

%
idebug: *Tools Reference* 119, 150
imap: *Tools Reference* 273
isim: *Tools Reference* 306

@, iserver: *Tools Reference* 293

+, idebug: *Tools Reference* 160

++, idebug: *Tools Reference* 159

***, idebug:** *Tools Reference* 126, 150, 155, 157, 162

**** , idebug:** *Tools Reference* 157, 162

\, in filenames: *Tools Reference* 14

__asm: *Tools Reference* 21; *Language Reference* 389
syntax: *Language Reference* 414
use when optimizing: *Optimizing Compiler Guide* 9

__CC_NORCROFT: *Language Reference* 388

__SIGNED_CHAR__: *Language Reference* 388

__ERRORMODE: *Language Reference* 388

__ICC: *Language Reference* 388

__IMS_BOARD_B004: *Language Reference* 28

__IMS_BOARD_B008: *Language Reference* 28

__IMS_BOARD_B010: *Language Reference* 28

__IMS_BOARD_B011: *Language Reference* 28

__IMS_BOARD_B014: *Language Reference* 28

__IMS_BOARD_B015: *Language Reference* 28

__IMS_BOARD_B016: *Language Reference* 28

__IMS_BOARD_CAT: *Language Reference* 28

__IMS_BOARD_DRX11: *Language Reference* 28

__IMS_BOARD_QT0: *Language Reference* 28

__IMS_BOARD_UDP_LINK: *Language Reference* 28

__IMS_clock_priority: *Language Reference* 365

__IMS_entry_term mode: *Language Reference* 366

__IMS_heap_init_implicit: *Language Reference* 363

__IMS_heap_size: *Language Reference* 363

__IMS_heap_start: *Language Reference* 363

__IMS_HOST_APOLLO: *Language Reference* 28

__IMS_HOST_IBM370: *Language Reference* 28

__IMS_HOST_NEC: *Language Reference* 28

__IMS_HOST_PC: *Language Reference* 28

__IMS_HOST_SUN3: *Language Reference* 28

__IMS_HOST_SUN386i: *Language Reference* 28

__IMS_HOST_SUN4: *Language Reference* 28

__IMS_HOST_VAX: *Language Reference* 28

__IMS_OS_CMS: *Language Reference* 28

__IMS_OS_DOS: *Language Reference* 28

__IMS_OS_HELIOS: *Language Reference* 28

__IMS_OS_SUNOS: *Language Reference* 28

__IMS_OS_VMS: *Language Reference* 28

__IMS_PData: *Language Reference* 364

__IMS_retval: *Language Reference* 366

__IMS_sbrk_alloc_request: *Language Reference* 363

__IMS_stack_base: *Language Reference* 363

__IMS_stack_limit: *Language Reference* 363

__IMS_startenv: *Language Reference* 365

__IMS_StartTime: *Language Reference* 365

__IOFBF: *Language Reference* 16

__IOLBF: *Language Reference* 16

__IONBF: *Language Reference* 16

__lsh: *Tools Reference* 20

__params: *Tools Reference* 20

__PTYPE: *Language Reference* 388

A

abort: *Language Reference* 18, 36, 422, 426
for dynamic code loading: *User Guide* 237
setting action: *Language Reference* 290

ABORT_EXIT: *Language Reference* 32

ABORT HALT: *Language Reference* 32

ABORT QUERY: *Language Reference* 32

abs: *Language Reference* 18, 37

Absolute value
float type: *Language Reference* 119

- floating point number: *Language Reference* 118
- integer number: *Language Reference* 37
- acos: *Language Reference* 11, 38
- acosf: *Language Reference* 27, 39
- Action strings, in makefiles: *Tools Reference* 267
- Alias: *User Guide* 271
 - check: *User Guide* 271
- Aliasing: *Language Reference* 409
- align: *Tools Reference* 346, 347
- alloc86: *Language Reference* 29, 40
- Allocate
 - channel: *Language Reference* 71
 - channels to links: *User Guide* 76
 - DOS memory: *Language Reference* 40
 - memory: *Language Reference* 68, 211
 - process: *Language Reference* 239
 - semaphore: *Language Reference* 283
 - software to hardware: *User Guide* 76
- Alphabetic character, test for: *Language Reference* 7, 183
- Alphanumeric character, test for: *Language Reference* 7, 182
- Analyse: *User Guide* 111, 135, 271; *Tools Reference* 116, 117
 - use when debugging: *User Guide* 113
- ANSI C
 - argument promotions: *Language Reference* 382, 407
 - compiler: *Tools Reference* 3
 - introduction: *User Guide* 10
 - optimizing: *Optimizing Compiler Guide* 3; *Performance note* 15
 - concurrency, libraries: *User Guide* 50
 - function prototypes, performance considerations: *Performance note* 17
 - implementation data: *Language Reference* 395
 - language, use when optimizing: *Optimizing Compiler Guide* 7
 - language extensions: *Language Reference* 387
 - new features: *Language Reference* 381
 - runtime library: *Language Reference* 3
 - standard: *User Guide* 10
 - compliance data: *Language Reference* 415
 - standard functions: *Language Reference* 6
 - toolset
 - development cycle: *User Guide* 21
 - performance improvements: *Performance note* 1
 - running benchmarks: *Performance note* 27
 - toolset introduction: *User Guide* 9
 - trigraphs: *Tools Reference* 24; *Optimizing Compiler Guide* 12
 - escape: *Language Reference* 386
- Append string: *Language Reference* 306, 317
- Arc cosine function: *Language Reference* 38
- Arc sine function: *Language Reference* 42
- Arc tangent function: *Language Reference* 46
- Areg: *User Guide* 134
- argv: *Language Reference* 365
- Arguments
 - ANSI C, default promotions: *Language Reference* 382, 407; *Performance note* 17
 - to main: *Language Reference* 400, 415

- variable: *Language Reference* 346
 - argv: *Language Reference* 365
 - Arithmetic, configuration language: *User Guide* 87
 - Arithmetic right shift: *Tools Reference* 8; *Performance note* 25
 - Array subscripting, or..., pointer update, performance considerations: *Performance note* 16
 - Arrays
 - as arguments to C functions: *User Guide* 154
 - as parameters, in configuration: *User Guide* 72
 - avoiding workspace: *Performance note* 21
 - constant, in configuration: *User Guide* 87
 - implementation: *Language Reference* 396, 418
 - in configuration language: *User Guide* 88
 - searching: *Language Reference* 65
 - subranges: *Tools Reference* 149, 159
 - asctime: *Language Reference* 21, 41
 - asin: *Language Reference* 11, 42
 - asinf: *Language Reference* 27
 - Assembler: *Tools Reference* 341
 - directives: *Tools Reference* 346
 - errors: *Tools Reference* 379
 - invoking: *Tools Reference* 7, 341
 - language: *Tools Reference* 343
 - syntax: *Tools Reference* 376
 - transputer instructions: *Tools Reference* 345
 - Assembly code: *User Guide* 253; *Language Reference* 389
 - asm statement: *User Guide* 253
 - literal bytes: *Language Reference* 390
 - opcodes: *User Guide* 253
 - operands: *Language Reference* 389
 - Assert
 - condition: *Language Reference* 44
 - debug condition: *Language Reference* 98
 - assert: *Language Reference* 7, 44, 422
 - assert.h: *Language Reference* 7
 - Assigning code to transputers: *User Guide* 22, 78
 - Asynchronous process: *User Guide* 55
 - atan: *Language Reference* 11, 46
 - atan2: *Language Reference* 11, 47
 - atan2f: *Language Reference* 27, 48
 - atanf: *Language Reference* 27, 49
 - atexit: *Language Reference* 18, 50
 - atof: *Language Reference* 18, 52
 - atoi: *Language Reference* 18, 54
 - atol: *Language Reference* 18, 56
 - Attributes, configuration: *User Guide* 86, 90
- ## B
- B004: *User Guide* 112; *Tools Reference* 317
 - B008: *User Guide* 113; *Tools Reference* 317
 - motherboard: *User Guide* 111
 - B014, motherboard: *User Guide* 111
 - B016, motherboard: *User Guide* 111

- Backslash, in filenames: *Tools Reference* 14
- BACKTRACE**: *Tools Reference* 146
- Backtrace: *User Guide* 271
- Backus-Naur Form
C language extensions: *Language Reference* 413
configuration language: *User Guide* 261
- bdos: *Language Reference* 29, 58
- Benchmarks: *Performance note* 27
- Big endian: *User Guide* 271
- binary. See *output.format*
- Binary lister: *Tools Reference* 237
command line: *Tools Reference* 238
errors: *Tools Reference* 251
- Binary output, *ieprom*: *User Guide* 227; *Tools Reference* 202
- Bit fields, implementation: *Language Reference* 403
- BitCnt: *Language Reference* 31, 59
- BitCntSum: *Language Reference* 31, 60
- BitRevNBits: *Language Reference* 31, 61
- BitRevWord: *Language Reference* 31, 63
- Bits in a byte, number of: *Language Reference* 9
- blkb: *Tools Reference* 346, 348
- blkw: *Tools Reference* 346, 349
- Block mode, *ieprom*: *Tools Reference* 203
- Block move: *Tools Reference* 21; *Performance note* 19, 22
- BlockMove: *Language Reference* 31, 64
- BNF: *User Guide* 261; *Language Reference* 413
- Boards
boot from link: *User Guide* 111
boot from ROM: *User Guide* 111
connections: *User Guide* 111
IMS B008: *User Guide* 111
IMS B014: *User Guide* 111
IMS B016: *User Guide* 111
types: *User Guide* 112
wiring: *Tools Reference* 108
- boards.inc: *Tools Reference* 53
- Booleans, in configuration language: *User Guide* 87
- Boot from link: *User Guide* 80; *Tools Reference* 177
boards: *User Guide* 111
collector memory map: *Tools Reference* 94, 97
default collector output: *Tools Reference* 86
loading mechanism: *User Guide* 110
- Boot from ROM: *Tools Reference* 92, 177, 195
boards: *User Guide* 111
code, debugging: *User Guide* 119
configurer options: *Tools Reference* 52
- Bootable code: *User Guide* 271; *Tools Reference* 81
- bootable.file: *Tools Reference* 198
- bootlink.h: *Language Reference* 29
- Bootstrap: *User Guide* 271
alternatives: *Tools Reference* 93
example: *Tools Reference* 421
loaders: *Tools Reference* 93, 422
- Bptr0: *User Guide* 134
- Bptr1: *User Guide* 134
- Branch-chaining optimization: *Optimizing Compiler Guide* 42
- Break key: *Tools Reference* 320

- Breakpoint debugging
See also *Debugging*; *Interactive debugging*
methods: *Tools Reference* 109
- Breakpoints: *User Guide* 144; *Tools Reference* 124, 308
commands: *Tools Reference* 124
hardware support: *User Guide* 125
menu: *Tools Reference* 124
phantom: *User Guide* 153
setting and clearing: *User Guide* 128
- Breg: *User Guide* 134
- Broken-down time
converted to string: *Language Reference* 41
structure: *Language Reference* 21, 22
- bsearch: *Language Reference* 18, 65
- BUFSIZ: *Language Reference* 16
- Build files, libraries: *User Guide* 274
- Building libraries: *Tools Reference* 211
- Built-in functions: *Tools Reference* 21; *Performance note* 22
- byte: *Tools Reference* 346, 350
- byte.select: *Tools Reference* 200
- C**
- C, implementation, compatibility issues: *Tools Reference* 8
- C main program: *User Guide* 49, 66, 79, 80, 81; *Language Reference* 357
- C runtime libraries
full: *User Guide* 224
reduced: *User Guide* 224
- C.ENTRY: *User Guide* 38
- C.ENTRYD: *User Guide* 37; *Language Reference* 357
- C.ENTRYD.RC: *User Guide* 37; *Language Reference* 357
- call without gsb: *User Guide* 249; *Tools Reference* 17; *Language Reference* 31, 67
- callc.lib: *User Guide* 205
- Calling conventions: *Language Reference* 407
- Calling functions, performance considerations: *Performance note* 23
- calloc: *Language Reference* 18, 68
- Capability: *User Guide* 272; *Tools Reference* 287, 291
specific host: *Tools Reference* 293
- CASE, debugging occam: *User Guide* 146
- Case
convert to lower case: *Language Reference* 342, 343
test for lower case: *Language Reference* 188
test for upper case: *Language Reference* 192
- ceil: *Language Reference* 11, 69
- ceilf: *Language Reference* 27, 70
- centryd1.c: *Language Reference* 358, 368
- centryd2.c: *Language Reference* 358, 368
- ChanAlloc: *User Guide* 58; *Language Reference* 24, 71
- CHANGE FILE**: *Tools Reference* 147
- Change processor, debugging: *Tools Reference* 137
- ChanIn: *User Guide* 59, 96; *Language Reference* 24, 72
- ChanInChanFail: *User Guide* 97; *Language Reference* 24, 73

- ChanInChar:** *User Guide* 59, 96; *Language Reference* 24, 74
- ChanInInt:** *User Guide* 59, 96; *Language Reference* 24, 75
- ChanInit:** *User Guide* 58; *Language Reference* 24, 76
- ChanInTimeFail:** *User Guide* 97; *Language Reference* 24, 77
- CHANNEL:** *Tools Reference* 144
- Channel, data type:** *User Guide* 50; *Language Reference* 25
- Channel:** *User Guide* 6, 48, 58, 70
- allocate function:** *Language Reference* 71
- character input:** *Language Reference* 74
- character output:** *Language Reference* 80
- communication:** *Performance note* 23
- direct:** *User Guide* 77, 96
- edge:** *User Guide* 77
- fault handling:** *User Guide* 97
- hard:** *User Guide* 274
- host server, in configuration:** *User Guide* 73
- initialization:** *User Guide* 58; *Language Reference* 76
- input:** *User Guide* 59
- function:** *Language Reference* 72
- recovery from failure:** *Language Reference* 73, 77
- input and output:** *User Guide* 75
- integer input:** *Language Reference* 75
- integer output:** *Language Reference* 81
- output:** *User Guide* 59
- function:** *Language Reference* 78
- recovery from failure:** *Language Reference* 79
- placement:** *User Guide* 76, 184
- reserved:** *User Guide* 216
- reset:** *User Guide* 97; *Language Reference* 83
- secure input:** *Language Reference* 73, 77
- secure output:** *Language Reference* 79, 82
- soft:** *User Guide* 77, 277
- virtual:** *User Guide* 77
- advanced techniques:** *User Guide* 183
- channel.h:** *User Guide* 50, 58; *Language Reference* 22, 24
- ChanOut:** *User Guide* 59, 96; *Language Reference* 24, 78
- ChanOutChanFail:** *User Guide* 97; *Language Reference* 24, 79
- ChanOutChar:** *User Guide* 59, 96; *Language Reference* 24, 80
- ChanOutInt:** *User Guide* 59, 96; *Language Reference* 24, 81
- ChanOutTimeFail:** *User Guide* 97; *Language Reference* 24, 82
- ChanReset:** *Language Reference* 24, 83
- char:** *Performance note* 17
- See also Character**
- default promotion:** *Language Reference* 382
- implementation:** *Language Reference* 395
- plain:** *Language Reference* 403, 417
- signedness:** *Tools Reference* 8; *Performance note* 19, 25
- CHAR_BIT:** *Language Reference* 9
- CHAR_MAX:** *Language Reference* 9
- CHAR_MIN:** *Language Reference* 9
- Character**
- constants, integer:** *Language Reference* 402
- escape codes:** *Language Reference* 380, 384, 386
- handling functions:** *Language Reference* 7
- input on channel:** *Language Reference* 74

- multibyte:** *Language Reference* 402, 416
- locale:** *Language Reference* 402
- output on channel:** *Language Reference* 80
- sequences, ANSI trigraphs:** *Language Reference* 386
- sets:** *Language Reference* 402, 416
- execution:** *Language Reference* 402
- source:** *Language Reference* 402
- signedness:** *Tools Reference* 8
- wide:** *Language Reference* 417
- See also wchar_t**
- Checking a network:** *Tools Reference* 125
- Clear file stream:** *Language Reference* 84
- clearerr:** *Language Reference* 14, 84
- Clearing error flags:** *User Guide* 114, 126; *Tools Reference* 166, 321
- clibs.lnk:** *User Guide* 39
- clibsrld.lnk:** *User Guide* 39
- Clock:** *User Guide* 134
- See also Timer**
- addition of values:** *Language Reference* 266
- comparison of values:** *Language Reference* 264
- rate:** *User Guide* 63
- subtraction of value:** *Language Reference* 265
- clock:** *Language Reference* 21, 85, 427
- clock_t:** *Language Reference* 21
- Clock0:** *User Guide* 134
- Clock1:** *User Guide* 134
- Clocks, displayed on Monitor page:** *User Guide* 136
- CLOCKS_PER_SEC:** *Language Reference* 21
- CLOCKS_PER_SEC_HIGH:** *Language Reference* 24
- CLOCKS_PER_SEC_LOW:** *Language Reference* 24
- close:** *Language Reference* 26, 87
- Close file stream:** *Language Reference* 120
- cnonconf.lnk:** *User Guide* 37
- Code**
- listing:** *Tools Reference* 242
- place in memory, configuration statements:** *User Guide* 179
- position in memory:** *User Guide* 71, 179; *Tools Reference* 54, 86, 88; *Performance note* 6, 7, 8
- Collector:** *User Guide* 26
- command line:** *Tools Reference* 82
- error messages:** *Tools Reference* 100
- example:** *User Guide* 44, 45
- input files:** *Tools Reference* 85
- output files:** *Tools Reference* 85
- non-bootable:** *Tools Reference* 91
- Command line:** *Tools Reference* 325
- Command line options**
- icc:** *Tools Reference* 4, 5
- icconf:** *Tools Reference* 51, 52
- icollect:** *Tools Reference* 84
- idebug:** *Tools Reference* 111
- iemit:** *Tools Reference* 178
- ieprom:** *Tools Reference* 197
- ilibr:** *Tools Reference* 208
- ilink:** *Tools Reference* 219
- ilist:** *Tools Reference* 239
- imakef:** *Tools Reference* 258
- imap:** *Tools Reference* 273
- iserver:** *Tools Reference* 285
- isim:** *Tools Reference* 304

- iskip:** *Tools Reference 318*
- optimizing compiler: *Tools Reference 6; Optimizing Compiler Guide 5*
- specify transputer target: *Tools Reference 339*
- comment:** *Tools Reference 346, 351*
- Comments
 - in assembly code: *Tools Reference 345*
 - in EPROM control files: *Tools Reference 197*
- common:** *Tools Reference 346, 352*
- Common subexpression elimination: *Optimizing Compiler Guide 45*
- Communicating Sequential Processes: *User Guide 6, 47, 272, 283*
- Communication. See Channel
- Compare
 - characters in memory: *Language Reference 217*
 - strings: *Language Reference 308*
 - times: *Language Reference 264*
- Compare memory, debugging: *Tools Reference 125*
- Compatibility, other C implementations: *Tools Reference 8*
- Compiler: *Tools Reference 3*
 - command line: *Tools Reference 3*
 - default: *Tools Reference 7*
 - control lines: *Language Reference 380*
 - diagnostics: *Tools Reference 22*
 - implementation data: *Tools Reference 331*
 - recoverable errors: *Tools Reference 31*
 - serious errors: *Tools Reference 38*
- terminology: *Tools Reference 22; Optimizing Compiler Guide 10*
- warnings: *Tools Reference 24*
- error modes: *User Guide 12; Tools Reference 7*
- introduction: *User Guide 10*
- libraries, occam: *User Guide 205, 211, 272*
- memory map: *Tools Reference 9*
- optimizations
 - general techniques: *Performance note 1*
 - in debugging: *User Guide 156*
 - optimizing: *Optimizing Compiler Guide 3; Performance note 15*
 - command line options: *Optimizing Compiler Guide 5*
 - global optimizations: *Optimizing Compiler Guide 45*
 - information messages: *Optimizing Compiler Guide 7*
 - language considerations: *Optimizing Compiler Guide 7*
 - local optimizations: *Optimizing Compiler Guide 41*
 - messages: *Optimizing Compiler Guide 10*
 - running: *Optimizing Compiler Guide 5*
- options: *Tools Reference 4, 5, 6*
- pragmas: *Tools Reference 15*
- predefines: *User Guide 16; Tools Reference 19*
- macros: *Tools Reference 19*
- preprocessor directives: *Tools Reference 12; Language Reference 384*
- implementation data: *Language Reference 421*
- selective loading of libraries: *Tools Reference 210*
- Compiling: *User Guide 25*
 - example: *User Guide 42, 45*
 - for a range of transputers: *Tools Reference 334*
 - for debugging: *User Guide 117*

- for dynamic loading: *User Guide 235*
- Concurrency: *User Guide 47*
- functions: *User Guide 51; Language Reference 22*
- hardware support: *User Guide 4*
- library support: *User Guide 49*
- model: *User Guide 11*
- support: *Language Reference 387*
- Conditionals, in configuration language: *User Guide 88*
- config.h:** *Language Reference 368*
- Configuration: *User Guide 272*
 - assigning code to processes: *User Guide 78*
 - checking: *User Guide 98*
 - code & data, placement in RAM: *User Guide 179*
 - code & data placement: *User Guide 17*
 - constants: *User Guide 264*
 - debugging considerations: *User Guide 96, 98, 99*
 - description: *User Guide 65*
 - example files: *Tools Reference 53*
 - examples: *User Guide 43, 81, 99*
 - hardware description: *User Guide 67*
 - introduction: *User Guide 16, 65*
 - language: *User Guide 85*
 - arrays: *User Guide 88*
 - booleans: *User Guide 87*
 - character set: *User Guide 86*
 - comments: *User Guide 85*
 - conditionals: *User Guide 88*
 - connections: *User Guide 92*
 - constants: *User Guide 86*
 - definition: *User Guide 261*
 - expressions and arithmetic: *User Guide 87*
 - identifiers: *User Guide 86*
 - implementation: *User Guide 261; Tools Reference 50*
- introduction: *User Guide 16*
- keywords: *User Guide 262*
- network definition: *User Guide 90*
- predefinitions: *User Guide 90, 262*
- replication: *User Guide 89*
- reserved words: *User Guide 261*
- statements: *User Guide 85*
- summary: *User Guide 93*
- syntax: *User Guide 266*
- syntax notation: *User Guide 261*
- types: *User Guide 86*
- mapping description: *User Guide 76*
- model: *User Guide 66*
- parameters. See *get_param*
- process termination: *User Guide 98*
- software description: *User Guide 70*
- software multiplexing: *User Guide 17*
- software routing: *User Guide 17*
- Configurer: *User Guide 26, 272; Tools Reference 49*
 - advanced toolset options: *Tools Reference 52*
 - command line: *Tools Reference 50*
 - default command line: *Tools Reference 52*
 - diagnostics
 - recoverable errors: *Tools Reference 60*
 - serious errors: *Tools Reference 75*
 - warnings: *Tools Reference 57*
 - errors: *Tools Reference 55*
 - information messages: *Tools Reference 56*
 - memory map: *Tools Reference 54; Performance note 6*
 - producing debuggable programs: *User Guide 118*
 - search paths: *Tools Reference 54*

- standard definitions: *Tools Reference* 53
- Configuring, for debugging: *User Guide* 153
- connect statement, in configuration description: *User Guide* 92
- Connecting
 - boards: *User Guide* 111
 - subnetworks: *User Guide* 111
- Connection database: *Tools Reference* 292
 - example: *Tools Reference* 295
 - format: *Tools Reference* 294
- Connection manager: *User Guide* 272
- Connections, in configuration
 - description: *User Guide* 92
 - edge: *User Guide* 75
 - prohibited: *User Guide* 92
- const: *Language Reference* 379, 382, 406; *Optimizing Compiler Guide* 8; *Performance note* 10
- Constants
 - arrays, in configuration: *User Guide* 87
 - configuration predefinitions: *User Guide* 264
 - floating point: *Language Reference* 380
 - in configuration language: *User Guide* 86
 - integer: *Language Reference* 380, 402
 - signal handling: *Language Reference* 12
 - syntax: *Language Reference* 384
- CONTINUE FROM**: *Tools Reference* 145
- Control character, test for: *Language Reference* 7, 185
- Conventions
 - command line options: *Tools Reference* 325
- command line syntax: *Tools Reference* 325
- error messages: *Tools Reference* 331
- filenames: *Tools Reference* 326
- imakefile extensions: *Tools Reference* 330
- search paths: *Tools Reference* 326
- standard file extensions: *Tools Reference* 327
- Conversion
 - char to double: *Language Reference* 52
 - error number to string: *Language Reference* 312
 - floating point: *Language Reference* 400
 - integers: *Language Reference* 399
 - lower to upper case: *Language Reference* 343
 - string to double: *Language Reference* 324
 - string to int: *Language Reference* 54
 - string to long int: *Language Reference* 56
 - time to string: *Language Reference* 97
 - to calendar time: *Language Reference* 221
 - to local time: *Language Reference* 202
 - upper to lower case: *Language Reference* 342
- Copy, characters in memory: *Language Reference* 218
- Core dump: *User Guide* 272; *Tools Reference* 311
 - listing: *Tools Reference* 250
- cos: *Language Reference* 11, 88
- cosf: *Language Reference* 27, 89
- cosh: *Language Reference* 11, 90
- coshf: *Language Reference* 27, 91

- Cosine function: *Language Reference* 88
 - CRC functions, résumé: *Language Reference* 429
 - CrcByte: *Language Reference* 31, 92, 429
 - CrcFromLsb: *Language Reference* 31, 93, 429
 - CrcFromMsb: *Language Reference* 31, 94, 429
 - CrcWord: *Language Reference* 31, 95, 429
 - creat: *Language Reference* 26, 96
 - Create file: *Language Reference* 96
 - See also fopen; open
 - Creg: *User Guide* 134
 - CSP: *User Guide* 6, 11, 47, 272, 283
 - cstartrd.lnk: *User Guide* 37, 80; *Language Reference* 357
 - cstartup.lnk: *User Guide* 37, 80; *Language Reference* 357
 - ctime: *Language Reference* 21, 97
 - ctype.h: *Language Reference* 7
 - Current location, in debugger: *Tools Reference* 146
 - Cursor positioning: *Tools Reference* 415
 - Cyclic redundancy functions, résumé: *Language Reference* 429
- D**
- Data
 - listing all: *Tools Reference* 248
 - output on channel: *Language Reference* 78
 - place in memory, configuration statements: *User Guide* 179
 - representation: *Language Reference* 395
 - data: *Tools Reference* 346, 353
 - Data types, implementation: *Language Reference* 395
 - Date and time
 - broken-down
 - convert to string: *Language Reference* 41
 - structure: *Language Reference* 22
 - daylight saving: *Language Reference* 427
 - defaults: *Language Reference* 405
 - functions: *Language Reference* 21
 - local time zone: *Language Reference* 427
 - DBL_DIG: *Language Reference* 8
 - DBL_EPSILON: *Language Reference* 8
 - DBL_MANT_DIG: *Language Reference* 8
 - DBL_MAX: *Language Reference* 9
 - DBL_MAX_10_EXP: *Language Reference* 9
 - DBL_MAX_EXP: *Language Reference* 8
 - DBL_MIN: *Language Reference* 8
 - DBL_MIN_10_EXP: *Language Reference* 8
 - DBL_MIN_EXP: *Language Reference* 8
 - Dead code elimination: *Optimizing Compiler Guide* 42
 - Deadlock: *User Guide* 146, 272
 - Debug
 - messages: *Language Reference* 99
 - support functions: *Tools Reference* 131, 145

debug: *Tools Reference* 346, 354

debug_assert: *Language Reference* 31, 98

debug_message: *Language Reference* 31, 99

debug_stop: *Language Reference* 31, 100

Debuggable programs: *User Guide* 116

Debugger: *User Guide* 27; *Tools Reference* 107

command line: *Tools Reference* 109

environment variables: *Tools Reference* 112

errors: *Tools Reference* 166

hints: *User Guide* 145

kernel: *User Guide* 124

monitor commands

- definitions: *Tools Reference* 123–142
- editing functions: *Tools Reference* 120
- mapped by ITERM: *Tools Reference* 120
- summary: *Tools Reference* 120–122

monitor page

- commands: *Tools Reference* 119
- scroll keys: *Tools Reference* 122
- symbolic commands: *Tools Reference* 122

program hangs: *Tools Reference* 166

scroll keys: *Tools Reference* 119

symbolic functions: *Tools Reference* 142

Debugging: *User Guide* 115

See also Interactive debugging; Monitor page; Post-mortem debugging

abusing hard links: *User Guide* 149

arrays as arguments: *User Guide* 154

B004 boards: *Tools Reference* 116

boot from ROM code: *User Guide* 119

breakpoint: *User Guide* 123

catching concurrent processes: *User Guide* 152

commands, only available in interactive mode: *User Guide* 128

compiler optimisations: *User Guide* 156

confidence check: *User Guide* 151

configuration: *User Guide* 153

configured programs: *User Guide* 118

current location: *Tools Reference* 146

deadfix.occ: *User Guide* 148

deadlock.occ: *User Guide* 147

direct channel functions: *User Guide* 118

error modes: *User Guide* 118

errors in the full library: *User Guide* 155

errors in the reduced library: *User Guide* 155

examining the active network: *User Guide* 149

example, C: *User Guide* 157

goto process: *User Guide* 176

hard parity errors: *User Guide* 120, 122

important points: *User Guide* 149

information: *User Guide* 117

INSPECT: *User Guide* 150

inspecting channels: *User Guide* 175; *Tools Reference* 144

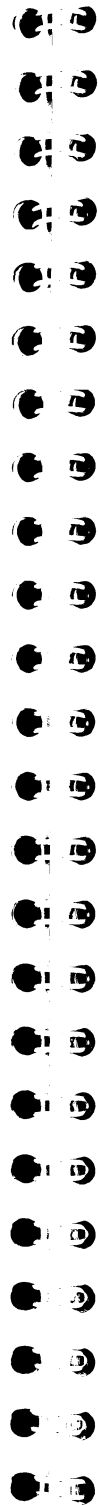
inspecting memory: *Tools Reference* 159

inspecting variables: *User Guide* 174

interactive: *Tools Reference* 228

INTERRUPT: *User Guide* 151

invalid pointers: *User Guide* 145



large shift values: *User Guide* 156

library functions, in absence of idebug: *User Guide* 143

loading programs: *User Guide* 112

low level: *User Guide* 132

memory size: *User Guide* 154

monitor page: *User Guide* 132

options, for different boards: *Tools Reference* 118

post-mortem: *User Guide* 119

program crashes: *User Guide* 152

program hangs: *User Guide* 152

program termination: *Tools Reference* 113

root transputer: *User Guide* 123

seterr: *User Guide* 154

single step: *Tools Reference* 312

soft configuration channels: *User Guide* 145

tracing processes: *User Guide* 175

TRAMs: *Tools Reference* 116

undetected program crashes: *User Guide* 152

use of isim: *User Guide* 116

Decimal digit, test for: *Language Reference* 7, 186

Declarators: *Language Reference* 382

- implementation: *Language Reference* 404, 421

Default

- argument promotions: *Language Reference* 382, 407; *Performance note* 17
- command line arguments: *User Guide* 35
- date: *Language Reference* 405
- error modes: *User Guide* 118
- time: *Language Reference* 405

defsym: *Tools Reference* 346, 355

DELETE: *Tools Reference* 260

Delete, file: *Language Reference* 345

descriptor: *Tools Reference* 346, 356

difftime: *Language Reference* 21, 101

Direct channels: *User Guide* 59, 77, 96, 97

Direct instructions: *User Guide* 253

DirectChanIn: *User Guide* 59; *Language Reference* 24, 102

DirectChanInChar: *User Guide* 59; *Language Reference* 24, 103

DirectChanInInt: *User Guide* 59; *Language Reference* 24, 104

DirectChanOut: *User Guide* 59; *Language Reference* 24, 105

DirectChanOutChar: *User Guide* 59; *Language Reference* 24, 106

DirectChanOutInt: *User Guide* 59; *Language Reference* 24, 107

Directives

- assembler: *Tools Reference* 346
- linker: *Tools Reference* 220
- preprocessor: *User Guide* 11; *Tools Reference* 12; *Language Reference* 380

Directory path: *Tools Reference* 326

Disassemble memory: *Tools Reference* 126

Display memory in hex: *Tools Reference* 129

Display reference: *Tools Reference* 248

Displaying object code: *Tools Reference* 237

div: *Language Reference* 18, 108

div_t: *Language Reference* 19

Division: *Language Reference* 108

dos.h: *Language Reference* 29

double: *Language Reference* 382, 396; *Performance note* 17, 18

Down: *User Guide* 111

DRAM timing parameters: *Tools Reference* 187

Dynamic code loading
functions: *Language Reference* 29

initialization: *User Guide* 232

input/output: *User Guide* 242

introduction: *User Guide* 15, 231

listing files: *Tools Reference* 250

occam: *User Guide* 248

E

Early write: *Tools Reference* 185

Edge: *User Guide* 75
channels: *User Guide* 77
host: *User Guide* 69
in configuration: *User Guide* 68

edge: *User Guide* 75

Editing functions: *Tools Reference* 120

Editing makefiles: *Tools Reference* 267

EDOM: *Language Reference* 8, 312, 426

EFILPOS: *Language Reference* 8, 426

EFIPOS: *Language Reference* 312

EIO: *Language Reference* 8, 312, 426

element: *User Guide* 90

Ellipsis: *Language Reference* 381

EMI: *User Guide* 273; *Tools Reference* 177
clock period: *Tools Reference* 185

Empty: *User Guide* 135

END OF FILE: *Tools Reference* 123, 147

End of file
character: *Language Reference* 16
test: *Language Reference* 121

end.offset: *Tools Reference* 200

ENTER FILE: *Tools Reference* 147

entry: *Language Reference* 380

Entry points
C.ENTRY: *User Guide* 38
C.ENTRYD: *User Guide* 37
C.ENTRYD.RC: *User Guide* 37
for dynamic code loading: *User Guide* 232, 237

enum: *Language Reference* 379, 382

enumeration: *Language Reference* 396

Enumeration types: *Language Reference* 382
implementation: *Language Reference* 403

Environment variables: *User Guide* 34; *Tools Reference* 416
accessing through **iserver:** *Tools Reference* 402

IBOARDSIZE: *Tools Reference* 87

ICARG: *Tools Reference* 7

ICCONFARG: *Tools Reference* 52

ICOLLECTARG: *Tools Reference* 85

ICONDB: *Tools Reference* 286, 293

ILIBRARG: *Tools Reference* 208

ILINKARG: *Tools Reference* 219

ILISTARG: *Tools Reference* 240

ISESSION: *Tools Reference* 286

ISIMBATCH: *Tools Reference* 313

ITERM: *Tools Reference* 116, 305

TRANSPUTER: *Tools Reference* 286, 292

used by **idebug:** *Tools Reference* 112

EOF: *Language Reference* 16

EPROM: *Tools Reference* 52, 92
code layout: *Tools Reference* 200
devices: *Tools Reference* 204

EPROM program convertor: *Tools Reference* 195
binary output: *Tools Reference* 202

block mode: *Tools Reference* 203

command line: *Tools Reference* 196

control file: *Tools Reference* 197

errors: *Tools Reference* 206

hex dump: *Tools Reference* 202

Intel extended hex format: *Tools Reference* 203

Intel hex format: *Tools Reference* 203

Motorola S-record format: *Tools Reference* 203

output files: *Tools Reference* 202

EPROM programming: *User Guide* 29, 225; *Tools Reference* 195
collecting: *User Guide* 228
configuring: *User Guide* 228
tools, introduction: *User Guide* 29

eprom.space: *Tools Reference* 198

ERANGE: *Language Reference* 8, 312, 422, 426

errno: *Language Reference* 5, 7, 426
on underflow: *Language Reference* 422

errno.h: *Language Reference* 7

Error: *User Guide* 111, 273

Error
handling: *Tools Reference* 331; *Language Reference* 7, 295
in file stream: *Language Reference* 122

modes: *User Guide* 12, 273; *Tools Reference* 7, 223

HALT: *User Guide* 273
in debugging: *User Guide* 118
selective loading of libraries: *Tools Reference* 210

STOP: *User Guide* 273

UNIVERSAL: *User Guide* 273

reporting: *User Guide* 33

runtime: *Tools Reference* 332

severities: *Tools Reference* 331

Error flag
clearing in a network: *User Guide* 114, 126; *Tools Reference* 166, 321

detection in interactive debugging: *Tools Reference* 117

displayed on Monitor page: *User Guide* 134, 135

of a subsystem: *User Guide* 111

setting: *Language Reference* 392
See also **abort;**
halt_processor;
set_abort_action

Error messages
assembler: *Tools Reference* 379

fatal runtime: *Language Reference* 32

format: *Tools Reference* 331

icc: *Tools Reference* 22

icconf: *Tools Reference* 55

icollect: *Tools Reference* 100

idebug: *Tools Reference* 166

idump: *Tools Reference* 176

iemit: *Tools Reference* 191

ieprom: *Tools Reference* 206

ilibr: *Tools Reference* 214

ilink: *Tools Reference* 230

ilist: *Tools Reference* 251

imakef: *Tools Reference* 268

imap: *Tools Reference* 281

iserver: *Tools Reference* 298
additional: *Tools Reference* 300

isim: *Tools Reference* 314

iskip: *Tools Reference* 321

optimizing compiler: *Optimizing Compiler Guide* 10

Escape codes: *Language Reference* 380

ESIGNUM: *Language Reference* 8, 312, 426

Ethernet: *User Guide* 273; *Tools Reference* 283

EVENT: *Language Reference* 25

Event: *User Guide* 273; *Tools Reference* 133, 311

Example, mapping description: *User Guide* 79

Examples

- analysing deadlock: *User Guide* 147
- bootstrap loader: *Tools Reference* 421
- collecting: *User Guide* 44, 45
- compiling: *User Guide* 42
- configuration: *User Guide* 43, 81, 99
- configuration files: *Tools Reference* 53
- connection database: *Tools Reference* 295
- CRC functions: *Language Reference* 431
- debugger monitor page: *User Guide* 133
- debugging C: *User Guide* 157
- debugging in post-mortem mode: *User Guide* 173
- debugging support functions: *User Guide* 142
- debugging occam: *User Guide* 166
- dynamic code loading: *User Guide* 236
- ieprom control file: *Tools Reference* 205
- imakef: *Tools Reference* 260
- occam: *Tools Reference* 263
- linking: *User Guide* 42, 45
- linking equivalent occam process: *User Guide* 224
- loading a program: *User Guide* 44
- multi-process program: *User Guide* 57
- phantom breakpoints: *User Guide* 153
- separate compilation: *User Guide* 45
- single process program: *User Guide* 52
- skip load: *User Guide* 113
- skipping a single processor: *Tools Reference* 319
- skipping multiple transputers: *Tools Reference* 319
- through-routing: *User Guide* 192
- transputer code: *Language Reference* 392
- type 1 interface: *User Guide* 219
- type 2 interface: *User Guide* 221
- type 3 interface: *User Guide* 223
- virtual channels: *User Guide* 104, 106

Executable code: *User Guide* 25

Execution character set: *Language Reference* 402

exit: *User Guide* 55, 98; *Language Reference* 18, 109, 120

- for dynamic code loading: *User Guide* 237
- status returned: *Language Reference* 427

EXIT FILE: *Tools Reference* 147

Exit program: *Language Reference* 109

EXIT_FAILURE: *Language Reference* 19

exit_noterminate: *User Guide* 98; *Language Reference* 31, 112

exit_repeat: *Language Reference* 31, 114

EXIT_SUCCESS: *Language Reference* 19

exit_terminate: *Language Reference* 31, 115

exp: *Language Reference* 11, 116

expf: *Language Reference* 27, 117

Exponential, floating point: *Language Reference* 236

Exponential function: *Language Reference* 116, 235

Exported names, listing: *Tools Reference* 244

Expressions, in configuration language: *User Guide* 87

Extended data types, occam: *User Guide* 273

Extensions

- file: *User Guide* 30; *Tools Reference* 254, 327
- language: *Language Reference* 387, 413

extern: *Tools Reference* 346, 357

External calls: *Performance note* 23

External memory interface: *User Guide* 273; *Tools Reference* 177

External references, listing: *Tools Reference* 251

extintel. See **output.format**

Extraction of library modules: *Tools Reference* 224

F

F, floating point suffix: *Language Reference* 380, 384

fabs: *Language Reference* 11, 118

fabsf: *Language Reference* 27, 119

facts.c: *User Guide* 157

- compiling and loading: *User Guide* 162

facts.occ: *User Guide* 166

- compiling and loading: *User Guide* 169

Fatal runtime errors: *Language Reference* 32

fclose: *Language Reference* 14, 120

feof: *Language Reference* 14, 121

ferror: *Language Reference* 14, 122

fflush: *Language Reference* 14, 123

fgetc: *Language Reference* 14, 124

fgetpos: *Language Reference* 14, 125, 426

fgets: *Language Reference* 14, 126

FILE: *Language Reference* 15

File

- buffering: *Language Reference* 16, 291
- close: *Language Reference* 87
- create temporary: *Language Reference* 338
- delete: *Language Reference* 345
- extensions: *User Guide* 30; *Tools Reference* 327
- imakef: *User Guide* 31; *Tools Reference* 254, 330
- imap source files: *Tools Reference* 273
- identification: *Tools Reference* 249, 327
- open: *Language Reference* 132
- pointer
 - repositioning: *Language Reference* 210
 - reset: *Language Reference* 157
- set to start: *Language Reference* 280
- read: *Language Reference* 276
- remove: *Language Reference* 278
- renaming: *Language Reference* 279
- size: *Language Reference* 127
- stream
 - buffering: *Language Reference* 294

clearing error: *Language Reference* 84
 close: *Language Reference* 120
 error: *Language Reference* 122
 position: *Language Reference* 155
 position indicator: *Language Reference* 125
 push character back: *Language Reference* 344
 read: *Language Reference* 140
 read character: *Language Reference* 124
 write: *Language Reference* 160
 write: *Language Reference* 356

Filename conventions: *Tools Reference* 326

FILENAME_MAX: *Language Reference* 16

filesize: *Language Reference* 26, 127

Fill memory: *Language Reference* 220

Find string: *Language Reference* 307
 in string: *Language Reference* 320

FINISH: *Tools Reference* 148

float: *Language Reference* 396; *Performance note* 17, 18
 default promotion: *Language Reference* 382

float.h: *Language Reference* 8

Floating point
 constants: *Language Reference* 380, 384
 conversion: *Language Reference* 400
 exponential: *Language Reference* 236
 implementation data: *Language Reference* 396, 418
 improving speed: *Performance note* 18
 instructions: *User Guide* 257

log: *Language Reference* 205
 multiply: *Language Reference* 195
 precision: *Performance note* 18
 remainder: *Language Reference* 130
 separation: *Language Reference* 146, 223
 truncation: *Language Reference* 400

floor: *Language Reference* 11, 128
 floorf: *Language Reference* 27, 129

Flowgraph optimization: *Optimizing Compiler Guide* 41

FLT_DIG: *Language Reference* 8
FLT_EPSILON: *Language Reference* 8
FLT_MANT_DIG: *Language Reference* 8
FLT_MAX: *Language Reference* 9
FLT_MAX_10_EXP: *Language Reference* 9
FLT_MAX_EXP: *Language Reference* 8
FLT_MIN: *Language Reference* 8
FLT_MIN_10_EXP: *Language Reference* 8
FLT_RADIX: *Language Reference* 8
FLT_ROUNDS: *Language Reference* 8

Flush file stream: *Language Reference* 123

fmod: *Language Reference* 11, 130, 423
 fmodf: *Language Reference* 27, 131

fn_info: *Language Reference* 30
 fnload.h: *User Guide* 233; *Language Reference* 29

fopen: *Language Reference* 14, 132
 mode strings: *Language Reference* 133

FOPEN_MAX: *Language Reference* 16

FPErrors: *User Guide* 134

fpos_t: *Language Reference* 15

fprintf: *Language Reference* 14, 134

Fptr0: *User Guide* 134
 Fptr1: *User Guide* 134

fputc: *Language Reference* 14, 138

fputs: *Language Reference* 14, 139

fread: *Language Reference* 14, 140

free: *Language Reference* 18, 142

Free memory: *Language Reference* 142, 143

Free variables: *User Guide* 273

free86: *Language Reference* 29, 143

freopen: *Language Reference* 14, 144

frexp: *Language Reference* 11, 146

frexpf: *Language Reference* 27, 148

from_host_link: *User Guide* 242; *Language Reference* 28, 149

from86: *Language Reference* 29, 150

fscanf: *Language Reference* 14, 151, 426

fseek: *Language Reference* 14, 155

fsetpos: *Language Reference* 14, 157

ftell: *Language Reference* 14, 159, 426

FTL_MIN_EXP: *Language Reference* 8

Full library. See *Library*

Function
 declarations: *Language Reference* 379, 381
 parameter lists: *Language Reference* 379
 variable: *Language Reference* 381
 prototypes: *Language Reference* 381; *Performance note* 17

fwrite: *Language Reference* 14, 160

G

Gateway: *User Guide* 273

General utility functions: *Language Reference* 17

GET ADDRESS: *Tools Reference* 147

Get character
 from file: *Language Reference* 169
 from stdin: *Language Reference* 170

get_bootlink_channels: *Language Reference* 29, 161, 364

get_code_details_from_channel: *User Guide* 233; *Language Reference* 30, 162

get_code_details_from_file: *User Guide* 233; *Language Reference* 30, 163

get_code_details_from_memory: *User Guide* 233; *Language Reference* 30, 164

get_details_of_free_memory: *Language Reference* 31, 165, 364

`get_details_of_free_stack_space`:
User Guide 238, 242; *Language Reference* 31, 166, 363

`get_init_chain_start`: *Language Reference* 367

`get_param`: *User Guide* 72, 80; *Language Reference* 31, 167, 364, 416

`GetArgsMyself`: *Language Reference* 365

`getc`: *Language Reference* 15, 169

`getchar`: *Language Reference* 15, 170

`getenv`: *Language Reference* 18, 171
 environment used: *Language Reference* 427

`getinit.s`: *Language Reference* 368

`getkey`: *Language Reference* 26, 172

`gets`: *Language Reference* 15, 173

`global`: *Tools Reference* 346, 358

Global compiler optimizations:
Optimizing Compiler Guide 45

Global static base: *User Guide* 201, 206; *Language Reference* 405, 407; *Performance note* 10
 dynamic code loading: *User Guide* 237, 249
 modifying runtime startup: *Language Reference* 359

`gmtime`: *Language Reference* 21, 174

Go to process: *Tools Reference* 129

`GOTO LINE`: *Tools Reference* 146

Grid, network topology: *User Guide* 5

H

HALT error mode: *User Guide* 118
 debugging: *User Guide* 118

`halt_processor`: *Language Reference* 31, 175

`HaltOnError`: *User Guide* 134

Hard channels: *User Guide* 274

Hardware characteristics: *Language Reference* 380

Hardware support
 for breakpointing: *User Guide* 125
 for concurrency: *User Guide* 4

Harness, dynamic code loading, for C: *User Guide* 232

Header files: *Language Reference* 5

Heap area: *Tools Reference* 87
 for dynamic code loading: *User Guide* 242
 for runtime startup: *Language Reference* 363
 mixed language programs: *User Guide* 205, 217
 position in memory: *User Guide* 71, 179; *Tools Reference* 54, 86; *Performance note* 6, 7
 size of: *User Guide* 71

`HELP`: *Tools Reference* 122, 142, 147

Help, page in debugger: *User Guide* 129

`hex`. See `output.format`

Hexadecimal
 arguments to `idump`: *Tools Reference* 175
 listing: *Tools Reference* 244

Hexadecimal digit, test for: *Language Reference* 7, 193

Hexadecimal format
 for environment variables: *User Guide* 35
 for EPROM: *User Guide* 227; *Tools Reference* 202

syntax: *User Guide* 35

High priority process: *User Guide* 63, 64, 71; *Language Reference* 258

Host: *User Guide* 274
 data: *Language Reference* 176
 dependencies: *User Guide* 33
 command line syntax: *User Guide* 33
 filenames: *User Guide* 34
 search paths: *User Guide* 34
 edge: *User Guide* 69
 environment variables: *User Guide* 34; *Language Reference* 171
 for capability: *Tools Reference* 293
 functions: *Language Reference* 28
 link, access: *Language Reference* 28
 sending command: *Language Reference* 332
 versions: *User Guide* xvii; *Tools Reference* xix; *Language Reference* ix; *Optimizing Compiler Guide* iii; *Performance note* iii

`host`: *User Guide* 69

Host file server: *User Guide* 274; *Tools Reference* 283
 terminating: *Tools Reference* 320

Host services: *User Guide* 80

`host.h`: *Language Reference* 28

`host_info`: *Language Reference* 28, 176

`hostlink.h`: *Language Reference* 28

`HUGE_VAL`: *Language Reference* 11

Hyperbolic
 cosine: *Language Reference* 90
 sine: *Language Reference* 299
 tangent: *Language Reference* 335

I

I/O: *Language Reference* 237
 buffering: *Language Reference* 16
 functions: *Language Reference* 14
 line buffering: *Language Reference* 16

IBM PC: *User Guide* 9
 386: *User Guide* 33

`IBOARDSIZE`: *User Guide* 35; *Tools Reference* 87, 113
 errors: *Tools Reference* 89

`icc`: *Tools Reference* 3

`channel_pointers`: *Tools Reference* 16

checking
`printf`: *Tools Reference* 16
`scanf`: *Tools Reference* 16
 stack: *Tools Reference* 16
 command line options: *Tools Reference* 4, 5, 6
 file extension defaults: *Tools Reference* 7

`inline_ops`: *Tools Reference* 16

introduction: *User Guide* 10
 memory map: *Tools Reference* 9
 optimizing compiler: *Optimizing Compiler Guide* 3; *Performance note* 15
 command line options: *Optimizing Compiler Guide* 5
 global optimizations: *Optimizing Compiler Guide* 45
 information messages: *Optimizing Compiler Guide* 7
 language considerations: *Optimizing Compiler Guide* 7
 local optimizations: *Optimizing Compiler Guide* 41
 messages: *Optimizing Compiler Guide* 10
 running: *Optimizing Compiler Guide* 5
 performance improvements: *Performance note* 1

search path: *Tools Reference 7*
 syntax: *Tools Reference 3*

ICCARG: *Tools Reference 7*

icconf: *Tools Reference 49*
 command line: *Tools Reference 50*
 error messages: *Tools Reference 55*
 introduction: *User Guide 16*

ICCONFARG: *Tools Reference 52*

icollect: *User Guide 26*
 command line, options: *Tools Reference 84*
 command line: *Tools Reference 82*
 environment variables: *Tools Reference 85, 87*
 errors: *Tools Reference 100*

ICOLLECTARG: *Tools Reference 85*

ICONDB: *User Guide 35; Tools Reference 286, 293*

idebug: *User Guide 27; Tools Reference 107*
 command line: *Tools Reference 109*
 options: *Tools Reference 111*
 environment variables: *Tools Reference 112*
 errors: *Tools Reference 166*
 help page: *User Guide 129*
 interactive mode: *Tools Reference 115*
 post-mortem debugging: *Tools Reference 113*
 restarting: *Tools Reference 115*

IDEBUGSIZE: *User Guide 35; Tools Reference 113*
 errors: *Tools Reference 166*

Identifiers: *Language Reference 380, 416*
 implementation: *Language Reference 402*
 in configuration language: *User Guide 86*

idump: *User Guide 28; Tools Reference 108, 175, 287, 317*
 errors: *Tools Reference 176*

IEEE 754: *User Guide 93*

iemit: *User Guide 29; Tools Reference 177*
 command line: *Tools Reference 178*
 DRAM timing parameters: *Tools Reference 187*
 errors: *Tools Reference 191*
 index page: *Tools Reference 180*
 input parameters: *Tools Reference 182*
 memory read cycle: *Tools Reference 188*
 memory write cycle: *Tools Reference 189*
 timing information: *Tools Reference 186*

ieprom: *User Guide 29, 225, 227; Tools Reference 195*
 command line: *Tools Reference 196*
 control file: *Tools Reference 197*
 errors: *Tools Reference 206*

IF, debugging occam: *User Guide 146*

if: *User Guide 88*

if...else: *User Guide 88*

ilibr: *User Guide 28; Tools Reference 207, 209*
 command line: *Tools Reference 208*
 command line options: *Tools Reference 208*
 error messages: *Tools Reference 214*

ILIBRARG: *Tools Reference 208*

ilink: *User Guide 25; Tools Reference 217*
 command line: *Tools Reference 218*
 indirect files: *Tools Reference 219*

ILINKARG: *Tools Reference 219*

ilist: *User Guide 28; Tools Reference 237*
 command line: *Tools Reference 238*
 command line options: *Tools Reference 239*
 errors: *Tools Reference 251*

ILISTARG: *Tools Reference 240*

imakef: *User Guide 28; Tools Reference 230, 253*
 command line: *Tools Reference 257*
 command line options: *Tools Reference 258*
 deleting intermediate files: *Tools Reference 260*
 errors: *Tools Reference 268*
 examples: *Tools Reference 260*
 file extensions: *Tools Reference 254, 330*
 file formats: *Tools Reference 266*
 linker indirect files: *Tools Reference 257, 259*
 occam examples: *Tools Reference 263*
 target files: *Tools Reference 254*

imap: *User Guide 28; Tools Reference 271; Performance note 3*
 command line: *Tools Reference 272*
 command line options: *Tools Reference 273*
 errors: *Tools Reference 281*
 output file structure: *Tools Reference 275*

Implementation
 arrays: *Language Reference 396*
 compiler diagnostics: *Tools Reference 331*
 configuration language: *User Guide 261*
 details: *Language Reference 395*
 structures: *Language Reference 397; Performance note 12*
 types: *Language Reference 395*
 unions: *Language Reference 399*

Importing C functions: *User Guide 205*

IMS B004: *Tools Reference 317*

IMS B008: *Tools Reference 317*

IMS B404: *Tools Reference 117*

IMS B405: *User Guide 68*

IMS T800: *User Guide 135*

IMS_descriptor: *Tools Reference 17*
 for dynamic code loading: *User Guide 235, 238*

IMS_nolink: *Tools Reference 17*

Include file: *User Guide 274*

INFO: *Tools Reference 144*

Information, facilities: *Performance note 3*

information%module: *Language Reference 370*

init: *Tools Reference 346, 359*

init.heap: *User Guide 207*

init.static: *User Guide 207*

initialise static: *Language Reference 361, 367*

Initialization
 channel: *User Guide 58; Language Reference 76*
 for dynamic code loading: *User Guide 232*
 process: *User Guide 53; Language Reference 245*
 semaphores: *User Guide 62; Language Reference 284*
 unions: *Language Reference 386*
 variable arguments: *Language Reference 349*

Inline functions: *Tools Reference 21; Performance note 22*

INMOS C
 concurrency: *User Guide 49*
 implementation, compatibility issues: *Tools Reference 8*
 introduction: *User Guide 10*

- Input/output functions: *Language Reference 14*
- INSPECT**: *User Guide 174; Tools Reference 143*
- Inspect memory: *Tools Reference 130*
- Instruction pointer: *User Guide 134*
- invalid: *User Guide 145*
- Instruction prefixing: *Performance note 1*
- Instruction set: *User Guide 49*
- int: *Language Reference 380, 396; Performance note 17*
- default promotion: *Language Reference 382*
- output on channel: *Language Reference 81*
- INT_MAX: *Language Reference 9*
- INT_MIN: *Language Reference 9*
- int86: *Language Reference 29, 178*
- int86x: *Language Reference 29, 179*
- intdos: *Language Reference 29, 180*
- intdosx: *Language Reference 29, 181*
- Integer
 - bitwise operations: *Language Reference 403*
 - constants: *Language Reference 380*
 - syntax: *Language Reference 384*
 - conversion: *Language Reference 399*
 - division: *Language Reference 108*
 - implementation data: *Language Reference 417*
 - input on channel: *Language Reference 75*
 - remainder on division: *Language Reference 403*
 - result of right shift: *Language Reference 403*
- intel. See *output.format*
- Intel extended hex format: *User Guide 227*
- ieprom: *Tools Reference 202*
- Intel hex format: *User Guide 227*
- ieprom: *Tools Reference 202*
- Interactive debugging: *User Guide 116, 123, 129*
- See also *Debugging*
- addresses of variables: *User Guide 164*
- backtracing: *User Guide 164, 172*
- backtracing to *main()*: *User Guide 165*
- breakpoint commands: *User Guide 132*
- browsing source code: *User Guide 130*
- clearing a breakpoint: *User Guide 173*
- collector option: *Tools Reference 99*
- entering *#include* files: *User Guide 166*
- inspecting by expression: *User Guide 165*
- inspecting variables: *User Guide 131, 164, 171*
- jumping down a channel: *User Guide 165, 172*
- jumping down channels: *User Guide 131*
- locating to code: *User Guide 130*
- modifying a variable: *User Guide 165, 172*
- modifying variables: *User Guide 132*
- program loading: *User Guide 126*
- program termination: *User Guide 128*
- quitting: *User Guide 166, 173*
- resuming program: *User Guide 172*
- runtime kernel: *User Guide 124*

- setting breakpoints: *User Guide 163, 171*
- starting a program: *User Guide 164, 171*
- tracing procedure calls: *User Guide 131*
- interface: *User Guide 71*
- INTERRUPT**: *Tools Reference 145*
- Interrupt, MS-DOS: *Language Reference 178, 179*
- Invalid pointers: *User Guide 145*
- io_and_hostinfo_init: *Language Reference 365*
- iocntrl.h: *Language Reference 26*
- lptr: *User Guide 134*
- lptrIntSave: *User Guide 134*
- isalnum: *Language Reference 7, 182, 422*
- isalpha: *Language Reference 7, 183, 422*
- isatty: *Language Reference 26, 184*
- iscntrl: *Language Reference 7, 185, 422*
- isdigit: *Language Reference 7, 186*
- ISEARCH: *User Guide 35; Tools Reference 14, 54, 326*
- iserver: *User Guide 26, 80, 109; Tools Reference 283, 317*
- access to functions: *Language Reference 287*
- accessing transputers: *Tools Reference 292*
- capability: *Tools Reference 287*
- command line: *Tools Reference 284*
- command line options: *Tools Reference 284*
- connection manager: *Tools Reference 297*
- environment variables: *Tools Reference 286*
- error codes: *Tools Reference 298*
- error messages: *Tools Reference 298*
- exit codes: *Tools Reference 298*
- functions: *Tools Reference 283*
- halt system error mode: *Tools Reference 287*
- loading programs: *Tools Reference 286*
- new features: *Tools Reference 297*
- passing parameters to a program: *Tools Reference 287*
- protocol: *Tools Reference 383*
- file commands: *Tools Reference 385*
- Fclose – close a file: *Tools Reference 386*
- Feof – test for end of file: *Tools Reference 395*
- Ferror – get file error status: *Tools Reference 396*
- FerrStat – Get file error status: *Tools Reference 398*
- Fflush – flush a stream: *Tools Reference 390*
- FGetBlock: *Tools Reference 388*
- FGetRec – read a record: *Tools Reference 393*
- Fgets – read a line: *Tools Reference 389*
- FileExists: *Tools Reference 398*
- Fopen – open a file: *Tools Reference 385*
- FopenRec: *Tools Reference 391*
- FPutBlock: *Tools Reference 389*
- FPutEOF: *Tools Reference 394*
- FPutRec – write a record: *Tools Reference 393*
- Fputs – write a line: *Tools Reference 390*

- Fread – read block of data:
Tools Reference 387
- Fseek – set position in a file:
Tools Reference 394
- Ftell – find position in a file:
Tools Reference 395
- Fwrite – write block of data:
Tools Reference 387
- Isatty: *Tools Reference 397*
- Remove – delete a file: *Tools Reference 396*
- Rename – Rename a file:
Tools Reference 397
- host commands
- Getenv – get environment variable: *Tools Reference 400*
- Getkey: *Tools Reference 399*
- Pollkey: *Tools Reference 399*
- System – run a command:
Tools Reference 401
- Time – get the time of day:
Tools Reference 401
- Translate – translate an environment variable: *Tools Reference 402*
- packets: *Tools Reference 383*
- record structured file commands: *Tools Reference 391*
- record structured file format:
Tools Reference 410
- reserved commands
- ALSYS: *Tools Reference 409*
- KPAR: *Tools Reference 410*
- MSDOS: *Tools Reference 408*
- SocketA: *Tools Reference 409*
- SocketM: *Tools Reference 409*
- server commands: *Tools Reference 383*
- CommandArgs: *Tools Reference 407*
- CommandLine: *Tools Reference 403*
- Core – read peeked memory:
Tools Reference 404
- Exit – exit the server: *Tools Reference 403*
- GetInfo: *Tools Reference 406*
- Version – find out about the server: *Tools Reference 405*
- termination codes: *Tools Reference 411*
- record structured files: *Tools Reference 298*
- session manager: *Tools Reference 284, 288, 297*
- customising interface: *Tools Reference 290*
- specifying the transputer to use:
Tools Reference 287
- stream identifier validation: *Tools Reference 298*
- subsystem reset: *Tools Reference 286*
- terminating: *Tools Reference 287*
on error: *Tools Reference 287*
- user interrupt: *Tools Reference 297*
- ISESSION: *User Guide 35; Tools Reference 286, 288*
- isgraph: *Language Reference 7, 187*
- isim: *User Guide 29, 144; Tools Reference 303*
- command line: *Tools Reference 303*
- command line options: *Tools Reference 304*
- errors: *Tools Reference 314*
- ISIMBATCH: *User Guide 35; Tools Reference 313*
- iskip: *User Guide 26, 109; Tools Reference 108, 317*
- command line: *Tools Reference 318*
- command line options: *Tools Reference 318*
- errors: *Tools Reference 321*
- islower: *Language Reference 7, 188, 422*
- ISO 646, character set: *Language Reference 386*
- ISO/IEC 9899:1990, standard:
User Guide 10

- isprint: *Language Reference 7, 189, 422*
- ispunct: *Language Reference 7, 190*
- ispy: *User Guide 98, 114, 126; Tools Reference 166, 321*
- isspace: *Language Reference 7, 191*
- istatic.c: *Language Reference 368*
- isupper: *Language Reference 7, 192, 422*
- isxdigit: *Language Reference 7, 193*
- ITERM: *User Guide 35; Tools Reference 113, 116, 305, 416*
- ITERM file
- example listing: *Tools Reference 418*
- format: *Tools Reference 413*
- keyboard: *Tools Reference 416*
- screen: *Tools Reference 414*
- use by simulator: *Tools Reference 305, 306*
- version: *Tools Reference 414*
- J
- JEDEC, symbol: *Tools Reference 186, 188*
- jmp_buf: *Language Reference 12*
- Jump instructions, in ROM: *Tools Reference 201*
- Jump into program: *Tools Reference 131*
- Jump tables: *Language Reference 393*
- Jumps: *Language Reference 393*

K

- Kernighan & Ritchie: *User Guide 10; Language Reference 379*
- Keyboard, read: *Language Reference 172*
- Keyboard definitions: *Tools Reference 416*
- Keywords: *Language Reference 380*
- configuration language: *User Guide 262*

L

- L
- floating point suffix: *Language Reference 380, 384*
- integer suffix: *Language Reference 384*
- L_INCR: *Language Reference 26*
- L_SET: *Language Reference 26*
- L_tmpnam: *Language Reference 16*
- L_XTND: *Language Reference 26*
- Labels, and __asm: *Language Reference 391*
- labs: *Language Reference 18, 194*
- LAN: *User Guide 274*
- language: *Tools Reference 346, 360*
- Language extensions, syntax: *Language Reference 413*
- Large shift values: *User Guide 156*
- Late write: *Tools Reference 185*
- LC_ALL: *Language Reference 10*
- LC_COLLATE: *Language Reference 10*
- LC_CTYPE: *Language Reference 10*
- LC_MONETARY: *Language Reference 10*

LC_NUMERIC: *Language Reference* 10
LC_TIME: *Language Reference* 10
lconv: *Language Reference* 10
LDBL_DIG: *Language Reference* 8
LDBL_EPSILON: *Language Reference* 8
LDBL_MANT_DIG: *Language Reference* 8
LDBL_MAX: *Language Reference* 9
LDBL_MAX_10_EXP: *Language Reference* 9
LDBL_MAX_EXP: *Language Reference* 8
LDBL_MIN: *Language Reference* 8
LDBL_MIN_10_EXP: *Language Reference* 8
LDBL_MIN_EXP: *Language Reference* 8
ldexp: *Language Reference* 11, 195
ldexpf: *Language Reference* 27, 196
ldiv: *Language Reference* 18, 197
ldiv_t: *Language Reference* 19
LFF files, listing: *Tools Reference* 250
Librarian: *User Guide* 28; *Tools Reference* 207
 command line: *Tools Reference* 208
 concatenated input: *Tools Reference* 207
 linked object input: *Tools Reference* 209
 options: *Tools Reference* 208
Library: *User Guide* 274
 ANSI functions: *Language Reference* 6
 build files: *User Guide* 274
 building: *Tools Reference* 211
 building optimized: *Tools Reference* 211
 character handling functions: *Language Reference* 7
 communication protocols: *Language Reference* 4
 date and time functions: *Language Reference* 21
 diagnostic functions: *Language Reference* 7
 extraction of modules: *Tools Reference* 224
 full: *User Guide* 80
 general utility functions: *Language Reference* 17
 header files: *User Guide* 14; *Language Reference* 5
 host functions: *Language Reference* 28
 implementation data: *Language Reference* 422
 index: *Tools Reference* 207, 210
 indirect files: *Tools Reference* 207, 209
 imakef: *Tools Reference* 257
 linking supplied libraries: *User Guide* 36; *Tools Reference* 220
 linking with program: *Language Reference* 4
 listing index: *Tools Reference* 246
 mathematics: *Language Reference* 11
 miscellaneous functions: *Language Reference* 25
 modules: *Tools Reference* 209
 occam: *User Guide* 213
 parallel processing: *Language Reference* 22
 reduced: *User Guide* 80; *Language Reference* 3
 runtime: *Language Reference* 3
 performance considerations: *Performance note* 2
 selective loading of: *Tools Reference* 210
 signal handling functions: *Language Reference* 12

 standard definitions: *Language Reference* 13
 string handling functions: *Language Reference* 20
 usage files: *User Guide* 274; *Tools Reference* 210
 imakef: *Tools Reference* 257
Limits: *Language Reference* 9
limits.h: *Language Reference* 9
LINE DOWN: *Tools Reference* 123
LINE UP: *Tools Reference* 122
Link: *User Guide* 274
Link map: *Tools Reference* 228
LINK0IN: *Language Reference* 25
LINK0OUT: *Language Reference* 25
LINK1IN: *Language Reference* 25
LINK1OUT: *Language Reference* 25
LINK2IN: *Language Reference* 25
LINK2OUT: *Language Reference* 25
LINK3OUT: *Language Reference* 25
Linker: *User Guide* 25, 275; *Tools Reference* 217
 command line: *Tools Reference* 218
 compatible transputer classes: *Tools Reference* 222
 directives: *Tools Reference* 220
 errors: *Tools Reference* 230
 extraction of library modules: *Tools Reference* 224
 indirect files: *User Guide* 36; *Tools Reference* 219
 imakef: *Tools Reference* 257, 259
 LFF output: *Tools Reference* 223
 selective loading of libraries: *Tools Reference* 210
 startup files: *User Guide* 36
 clibs.lnk: *User Guide* 224
 clibsrld.lnk: *User Guide* 224
 TCOFF output: *Tools Reference* 223
Linking
 example: *User Guide* 42, 45
 libraries: *Language Reference* 4
 mixed language programs: *User Guide* 204
 transputer targets: *Tools Reference* 333
Linkops: *User Guide* 275
linkquota: *User Guide* 67, 69, 187, 189
Links: *User Guide* 5; *Tools Reference* 133, 311
 introduction: *User Guide* 4
Lister: *User Guide* 28
 See also **ilist**
Little endian: *User Guide* 275
load_code_from_channel: *User Guide* 234; *Language Reference* 30, 198
load_code_from_file: *User Guide* 234; *Language Reference* 30, 199
load_code_from_memory: *User Guide* 234; *Language Reference* 30, 200
Loader: *User Guide* 275
Loading programs: *User Guide* 109
 example: *User Guide* 44
 for breakpoint debugging: *User Guide* 112
 for debugging: *User Guide* 112
 for interactive debugging: *User Guide* 126
 introduction: *User Guide* 26
 iserver: *Tools Reference* 283
 iskip: *Tools Reference* 320
 methods: *User Guide* 110
 onto boards and subnetworks: *User Guide* 110
 tools: *User Guide* 109

- LoadStart:** *User Guide* 179; *Tools Reference* 54, 55, 94, 96
- Local compiler optimizations:** *Optimizing Compiler Guide* 41
- Locale:** *Language Reference* 402, 427
See also **Set program locale data:** *Language Reference* 201
setting: *Language Reference* 293
- locale.h:** *Language Reference* 9
- localeconv:** *Language Reference* 9, 201
- localhost:** *Tools Reference* 293
- Localisation functions:** *Language Reference* 9
- localtime:** *Language Reference* 21, 202
- Location, in debugger:** *Tools Reference* 146
- location:** *User Guide* 71, 74, 179, 182; *Performance note* 7
- log:** *Language Reference* 11, 204
- log10:** *Language Reference* 11, 206
- log10f:** *Language Reference* 27, 207
- logf:** *Language Reference* 27, 205
- Logical name:** *Tools Reference* 416
- long:** *Language Reference* 380
- Long division:** *Language Reference* 197
- Long integers:** *Language Reference* 194
- LONG_MAX:** *Language Reference* 9
- LONG_MIN:** *Language Reference* 9
- longjmp:** *Language Reference* 12, 208
- Loop unrolling:** *Performance note* 15
- Loop-invariant code, optimization:** *Optimizing Compiler Guide* 46
- Low priority process:** *User Guide* 63, 64, 71; *Language Reference* 259
- Lower case**
convert to: *Language Reference* 7
convert to upper: *Language Reference* 343
test for: *Language Reference* 7, 188
- lseek:** *Language Reference* 26, 210

M

- Macros**
definition: *Tools Reference* 12
error handling: *Language Reference* 8
floating point: *Language Reference* 8, 9
implementation limits: *Language Reference* 9
in makefiles: *Tools Reference* 266
locale: *Language Reference* 10
predefined: *Language Reference* 388
signal handling: *Language Reference* 12
standard: *Language Reference* 14
time and date: *Language Reference* 21
- Main entry point:** *Tools Reference* 227
- main function:** *User Guide* 49, 66, 79, 80, 81; *Language Reference* 357
meaning of arguments: *Language Reference* 400
- MAIN.ENTRY:** *User Guide* 214
procedure interface: *User Guide* 218
- maininit:** *Tools Reference* 346, 361

- Make programs:** *Tools Reference* 253
Borland: *Tools Reference* 253
Gnu: *Tools Reference* 253
Microsoft: *Tools Reference* 253
Unix: *Tools Reference* 253
- Makefile generator:** *User Guide* 28; *Tools Reference* 253
command line: *Tools Reference* 257
errors: *Tools Reference* 268
- Makefiles:** *User Guide* 275
delete rule: *Tools Reference* 267
editing: *Tools Reference* 267
formats: *Tools Reference* 266
macros: *Tools Reference* 266
- malloc:** *Language Reference* 18, 211
- map1:** *Tools Reference* 346, 362
- map2:** *Tools Reference* 346, 363
- Master transputer, of a system:** *User Guide* 111
- math.h:** *Language Reference* 11
- mathf.h:** *Language Reference* 26
- Maths functions:** *Language Reference* 11
- max_stack_usage:** *User Guide* 237, 238; *Language Reference* 31, 212, 363; *Performance note* 11
- MB_CUR_MAX:** *Language Reference* 19
- MB_LEN_MAX:** *Language Reference* 9
- mblen:** *Language Reference* 18, 213
- mbstowcs:** *Language Reference* 18, 214
- mbtowc:** *Language Reference* 18, 215
- memchr:** *Language Reference* 20, 216; *Performance note* 20
- memcpy:** *Language Reference* 20, 217; *Performance note* 20
- MemConfig:** *Tools Reference* 177
- memcpy:** *Language Reference* 20, 218; *Performance note* 19
- memmove:** *Language Reference* 20, 219; *Performance note* 19
- MemnotWrD0:** *Tools Reference* 177
- Memory**
allocate: *Language Reference* 211
allocate DOS memory: *Language Reference* 40
allocate function: *Language Reference* 68
configuration
ASCII output: *Tools Reference* 180
customized: *Tools Reference* 177
file: *Tools Reference* 192
in PAL: *Tools Reference* 177
in ROM: *Tools Reference* 177, 200
PostScript output: *Tools Reference* 180
standard: *Tools Reference* 177, 185
table: *Tools Reference* 190
configurer: *Tools Reference* 177
command line: *Tools Reference* 178
default configuration: *Tools Reference* 180
errors: *Tools Reference* 191
input parameters: *Tools Reference* 182
interactive operation: *Tools Reference* 180
output files: *Tools Reference* 180
default layout, configured programs: *User Guide* 179
disassembly: *Tools Reference* 308
DOS transfer: *Language Reference* 150

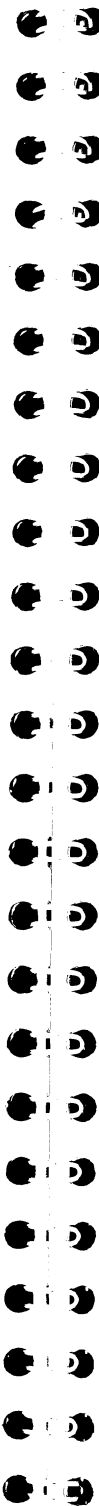
- freeing: *Language Reference* 142
 - Hex display: *Tools Reference* 129
 - improving use of: *Performance note* 5
 - initializing: *User Guide* 126, 217
 - inspecting: *Tools Reference* 310
 - insufficient: *Language Reference* 32
 - interface, configurable, T4 and T8 series: *Tools Reference* 177
 - mapper: *Tools Reference* 271
 - command line: *Tools Reference* 272
 - errors: *Tools Reference* 281
 - on-chip: *User Guide* 3
 - read cycle: *Tools Reference* 188
 - reallocate: *Language Reference* 277
 - reserved words
 - lprIntSave: *User Guide* 134
 - WdescIntSave: *User Guide* 134
 - reserving: *User Guide* 179
 - reserving on-chip. See **reserved**
 - segment ordering: *User Guide* 73
 - segment re-location: *User Guide* 73
 - use by
 - software virtual routing processes: *User Guide* 190
 - virtual routing software: *User Guide* 185
 - write cycle: *Tools Reference* 189
- memory:** *User Guide* 67, 90
- Memory dump:** *User Guide* 123
example: *User Guide* 174
- Memory dumper:** *User Guide* 28; *Tools Reference* 175
command line: *Tools Reference* 175
error messages: *Tools Reference* 176
- Memory map:** *Tools Reference* 134, 311; *Performance note* 3
boot from link (network): *Tools Reference* 97
boot from link (single processor): *Tools Reference* 94
boot from ROM: *Tools Reference* 98
collector output: *Tools Reference* 93
configurer: *Tools Reference* 54; *Performance note* 6
displayed on monitor page: *User Guide* 136
single processor program: *Performance note* 7
- Memory mapped devices, access:** *Performance note* 14
- memory.configuration:** *Tools Reference* 198
- memset:** *Language Reference* 20, 220; *Performance note* 19
- MemStart:** *User Guide* 135, 180; *Tools Reference* 94; *Performance note* 6, 7
- MemWait:** *Tools Reference* 185, 189
connection error: *Tools Reference* 191
- Messages.** See **Error messages**
- Minimum fp exponent:** *Language Reference* 8
- misc.h:** *Language Reference* 30
- Miscellaneous functions:** *Language Reference* 25
- Mixed language programming:** *User Guide* 197
heap area: *User Guide* 205
importing C code: *User Guide* 205
introduction: *User Guide* 17
linking: *User Guide* 204
occam libraries: *User Guide* 213
reduced runtime library: *User Guide* 212

- static area: *User Guide* 205
 - vector space: *User Guide* 213
 - workspace: *User Guide* 213
- mtime:** *Language Reference* 21, 221
- modf:** *Language Reference* 11, 223
- modff:** *Language Reference* 27, 224
- MODIFY:** *Tools Reference* 145
- Module data, listing:** *Tools Reference* 245
- MONITOR:** *Tools Reference* 148
- Monitor page:** *User Guide* 132
See also **Debugging**
breakpoint commands: *User Guide* 139
command format: *User Guide* 137
commands: *Tools Reference* 119
data displayed: *User Guide* 134
default address: *Tools Reference* 119
display virtual links: *Tools Reference* 141
Enter post-mortem: *Tools Reference* 140
examining memory: *User Guide* 137
exit: *Tools Reference* 140
locating processes: *User Guide* 137
selecting process: *User Guide* 138
simulator: *Tools Reference* 305
specifying process: *User Guide* 138
startup display: *User Guide* 133
switching processor: *User Guide* 138
- Monitoring the error status:** *Tools Reference* 320
- Motorola S-record format:** *User Guide* 227
ieprom: *Tools Reference* 202
- Move2D:** *Language Reference* 225
- Move2DNonZero:** *Language Reference* 227
- Move2DZero:** *Language Reference* 229
- MS-DOS:** *User Guide* 9, 33, 34, 35; *Tools Reference* 325, 416
function call: *Language Reference* 58
read registers: *Language Reference* 282
software interrupt: *Language Reference* 178, 179, 180, 181
system functions: *Language Reference* 29
- Multibyte characters, shift states:** *Language Reference* 402
- Multiple processes:** *Language Reference* 242
- Multiprocessor networks:** *User Guide* 49
- MUXER_ORDER:** *User Guide* 185
- N**
- Natural logarithm:** *Language Reference* 204
- NDEBUG:** *Language Reference* 7
- Network:** *User Guide* 275
configuration: *User Guide* 65
control of, software virtual routing: *User Guide* 185
definition: *User Guide* 90
dump: *Tools Reference* 135
 listing: *Tools Reference* 250
grid: *User Guide* 5
hardware description: *User Guide* 67
mapping description: *User Guide* 76
partitioning: *User Guide* 184, 190
pipeline: *User Guide* 5

software description: *User Guide* 70
 spanning tree: *User Guide* 188
 Tree: *User Guide* 5
 Next error: *Tools Reference* 127
 Node: *User Guide* 66, 90
 types: *User Guide* 91
 nodebug: *User Guide* 71
 Non-ANSI functions: *Language Reference* 25
 Non-bootable files
 dynamic code loading: *User Guide* 231
 format: *Tools Reference* 91
 Non-configured programs. See *cnonconf.lnk*; *icollect*
 Non-local jump: *Language Reference* 12, 208
 setting up: *Language Reference* 292
 Non-space printable character, test for: *Language Reference* 7
 noprofile: *User Guide* 71
 notMemRd: *Tools Reference* 184
 notMemS0: *Tools Reference* 184
 notMemS4: *Tools Reference* 184
 notMemWrB: *Tools Reference* 184
 NotProcess: *User Guide* 135
 NotProcess_p: *Language Reference* 25
 NULL: *Language Reference* 21
 NULL, implementation: *Language Reference* 422
 NULL pointer constant: *Language Reference* 14, 15, 19, 21
 implementation: *Language Reference* 409
 Numerical parameters, interpretation by *isim*: *Tools Reference* 306

O

O_APPEND: *Language Reference* 26
 O_BINARY: *Language Reference* 26
 O_RDONLY: *Language Reference* 26
 O_RDWR: *Language Reference* 26
 O_TEXT: *Language Reference* 26
 O_TRUNC: *Language Reference* 26
 O_WRONLY: *Language Reference* 26
 Object code: *User Guide* 275
 displaying: *Tools Reference* 237
 optimizing: *Optimizing Compiler Guide* 3; *Performance note* 15
 Object file, format: *User Guide* 11, 25
 occam
 compiler libraries: *User Guide* 272
 dynamic code loading: *User Guide* 248
 equivalent process: *User Guide* 214
 extended data types: *User Guide* 273
 interface code: *User Guide* 214
 libraries: *User Guide* 213
 mixing with C code: *User Guide* 197
 occam2.lnk: *User Guide* 38
 occam8.lnk: *User Guide* 38
 occama.lnk: *User Guide* 38
 offsetof: *Language Reference* 14
 On-chip memory: *User Guide* 3; *Performance note* 1, 5
 use for program stack: *User Guide* 217
 open: *Language Reference* 26, 231



Open file: *Language Reference* 132
 Open file stream: *Language Reference* 231
 Operating systems
 command lines: *User Guide* 33
 dependencies: *User Guide* 33
 MS-DOS: *User Guide* 33
 SunOS: *User Guide* 33
 Unix: *User Guide* 33
 VMS: *User Guide* 33
 Operations: *User Guide* 254
 Operators: *User Guide* 87
 unary: *Language Reference* 380
 Optimizing object code
 compact code: *Performance note* 1
 for space: *Optimizing Compiler Guide* 6, 45
 for time: *Optimizing Compiler Guide* 6, 45
 global optimizations: *Optimizing Compiler Guide* 45
 language considerations: *Optimizing Compiler Guide* 7
 local optimizations: *Optimizing Compiler Guide* 41
 performance techniques: *Performance note* 1
 run faster: *Performance note* 1
 using *icc*: *Optimizing Compiler Guide* 3; *Performance note* 15
 Options
 prefix: *User Guide* 33
 specify transputer target: *Tools Reference* 339
 standard: *Tools Reference* 325
 unsupported: *User Guide* 39; *Tools Reference* 326
 order: *User Guide* 71, 73, 179, 183, 185; *Performance note* 7
 Out of memory errors, *idebug*: *Tools Reference* 166
 output.address: *Tools Reference* 200

output.all: *Tools Reference* 199
 output.block: *Tools Reference* 199
 output.format: *Tools Reference* 199

P

PAGE DOWN: *Tools Reference* 123
 PAGE UP: *Tools Reference* 123
 Parallel processing
 data types: *User Guide* 50
 functions, summary: *User Guide* 51
 introduction: *User Guide* 6, 47
 model: *User Guide* 47
 on transputers: *User Guide* 48
 Parameters
 from configurer. See *get_param*
 GSB: *User Guide* 206
 passing: *Language Reference* 407
 passing by reference: *User Guide* 199
 passing by value: *User Guide* 199
 TIMER: *User Guide* 200
 Parity checked memory, initializing: *User Guide* 126
 Parity error registers, displayed on Monitor page: *User Guide* 136
 Parity errors, post-mortem debugging: *User Guide* 120, 122
 ParityAddr: *User Guide* 134
 ParityError: *User Guide* 134
 patch: *Tools Reference* 346, 364
 codefix: *Tools Reference* 365
 datafix: *Tools Reference* 366
 extoffset: *Tools Reference* 367
 limit: *Tools Reference* 368
 modnumber: *Tools Reference* 369

staticfix: *Tools Reference* 370

Path searching: *Tools Reference* 326

ppointer: *Language Reference* 29

Peek: *User Guide* 276

Peephole optimization: *Optimizing Compiler Guide* 41

Performance improvement techniques: *Performance note* 1
using optimizing compiler: *Optimizing Compiler Guide* 3; *Performance note* 15

perorr: *Language Reference* 15, 233, 426

Phantom breakpoints: *User Guide* 153

Pipeline, network: *User Guide* 5

place: *User Guide* 184
in configuration: *User Guide* 76

Placement channels: *User Guide* 76
processes: *User Guide* 76

Plain chars: *Language Reference* 403

Pointer update, versus, array subscripting: *Performance note* 16

Pointers, implementation data: *Language Reference* 418

Poke: *User Guide* 276

Poll keyboard: *Language Reference* 234

pollkey: *Language Reference* 26, 234

Porting C: *Tools Reference* 8

Post-mortem debugging: *User Guide* 115, 119
See also Debugging
communication on channels: *User Guide* 141

communication on links: *User Guide* 140

communication on virtual links: *User Guide* 141

hard parity errors: *User Guide* 120, 122

locating procedures and functions: *User Guide* 141

outline of method: *User Guide* 139

stopped process: *User Guide* 141

stopped process location: *User Guide* 140

waiting on run queue: *User Guide* 140

waiting on timer queue: *User Guide* 140

PostScript: *User Guide* 276

pow: *Language Reference* 11, 235

powf: *Language Reference* 27, 236

Pragmas: *Language Reference* 387
See also #pragma
icc: *Tools Reference* 15
optimizing compiler: *Optimizing Compiler Guide* 8

Preamble: *User Guide* 276

Predefines, in configuration language: *User Guide* 90

Prefixing instructions: *User Guide* 253; *Performance note* 1

Preprocessor directives: *Tools Reference* 12; *Language Reference* 380, 384
implementation data: *Language Reference* 421
use with assembler: *Tools Reference* 342

Printable character, test for: *Language Reference* 7, 187, 189

printf: *Language Reference* 15, 237

Priority: *User Guide* 276; *Tools Reference* 138
of execution: *User Guide* 73

process: *Language Reference* 244

priority: *User Guide* 71

PROC.ENTRY: *User Guide* 215
procedure interface: *User Guide* 219

PROC.ENTRY.RC: *User Guide* 215
procedure interface: *User Guide* 222

PROC_HIGH: *Language Reference* 24

PROC_LOW: *Language Reference* 24

ProcAfter: *User Guide* 63; *Language Reference* 23, 238

ProcAlloc: *User Guide* 53; *Language Reference* 23, 239
use with, dynamic code loading: *User Guide* 234

ProcAllocClean: *User Guide* 54; *Language Reference* 23, 241

ProcAlt: *User Guide* 61; *Language Reference* 23, 242

ProcAltList: *User Guide* 61; *Language Reference* 23, 243

ProcClockOut: *Tools Reference* 184, 185

Procedural interface data, listing: *Tools Reference* 247

Process, structure type: *User Guide* 50; *Language Reference* 24

Process: *User Guide* 6, 48, 276
allocate: *Language Reference* 239
asynchronous: *User Guide* 55
configuration attributes: *User Guide* 70
control: *User Guide* 49
creation: *User Guide* 51
defining new types: *User Guide* 74

descriptor: *User Guide* 134
invalid: *User Guide* 145
execution: *User Guide* 54
freeing workspace: *User Guide* 54
get parameters: *Language Reference* 253
get priority: *Language Reference* 244
initialization: *User Guide* 53; *Language Reference* 245
interface: *User Guide* 71
memory map: *Tools Reference* 139
pointers, in debugging: *User Guide* 135
prioritizing: *User Guide* 71; *Language Reference* 255
queue: *User Guide* 135; *Tools Reference* 312
displaying: *User Guide* 176; *Tools Reference* 138
rescheduling: *User Guide* 64; *Language Reference* 256
selection: *User Guide* 49
starting: *Language Reference* 257
starting multiples: *Language Reference* 252
stopping: *Language Reference* 262
suspending: *Language Reference* 269
synchronous: *User Guide* 56
termination: *User Guide* 54, 98
timing: *User Guide* 49; *Language Reference* 263
timing out: *Language Reference* 267
unused pointer: *User Guide* 53

process: *User Guide* 70, 90

process.h: *User Guide* 50, 54; *Language Reference* 22, 23

Processes, synchronising: *User Guide* 48, 58

Processor links: *User Guide* 67

names: *Tools Reference* 133
 types: *Tools Reference* 333; *Performance note* 1, 2

processor: *User Guide* 67, 90
 defining new types: *User Guide* 68

ProcGetPriority: *User Guide* 64; *Language Reference* 23, 244

ProcInit: *User Guide* 53; *Language Reference* 23, 245
 use with, dynamic code loading: *User Guide* 234

ProcInitClean: *User Guide* 54; *Language Reference* 23, 248

ProcJoin: *User Guide* 55; *Language Reference* 23, 250

ProcJoinList: *User Guide* 55; *Language Reference* 23, 251

ProcPar: *User Guide* 56; *Language Reference* 23, 252

ProcParam: *User Guide* 53; *Language Reference* 23, 253

ProcParList: *User Guide* 56; *Language Reference* 23, 254

ProcPriPar: *User Guide* 56; *Language Reference* 23, 255

ProcReschedule: *User Guide* 64; *Language Reference* 23, 256

ProcRun: *User Guide* 55; *Language Reference* 23, 257

ProcRunHigh: *User Guide* 55; *Language Reference* 23, 258

ProcRunLow: *User Guide* 55; *Language Reference* 23, 259

ProcSkipAlt: *User Guide* 61; *Language Reference* 23, 260

ProcSkipAltList: *User Guide* 61; *Language Reference* 261

ProcStop: *User Guide* 54; *Language Reference* 23, 262

ProcTime: *User Guide* 63; *Language Reference* 23, 263

ProcTimeAfter: *User Guide* 63; *Language Reference* 23, 264

ProcTimeAlt: *User Guide* 64

ProcTimeAltList: *User Guide* 64

ProcTimeMinus: *User Guide* 63; *Language Reference* 23, 265

ProcTimePlus: *User Guide* 63; *Language Reference* 23, 266

ProcTimerAlt: *Language Reference* 23, 267

ProcTimerAltList: *Language Reference* 23, 268

ProcWait: *User Guide* 63; *Language Reference* 23, 269

Program, execution time: *Language Reference* 85

Program development
 getting started: *User Guide* 41
 introduction: *User Guide* 21

Program hangs, debugging: *User Guide* 152

Program termination: *Language Reference* 109
 for configured programs: *Language Reference* 112, 115
 function call: *Language Reference* 50
 interactive debugging: *User Guide* 128
 with restart: *Language Reference* 114
 without terminating the server: *Language Reference* 112

Programmable memory interface: *User Guide* 3

Programs, loading: *User Guide* 109

Protocol: *User Guide* 276

iserver: *User Guide* 110; *Tools Reference* 383
 in debugger: *User Guide* 123
 SP: *User Guide* 110
 used by library: *Language Reference* 4

used by standard libraries: *User Guide* 123

Prototypes: *Language Reference* 381; *Performance note* 17

prtdiff_t: *Language Reference* 13

Pseudo-operations: *Language Reference* 389

Pseudo-random numbers: *Language Reference* 275

Punctuation character
 definition of: *Language Reference* 190
 test for: *Language Reference* 7, 190

putc: *Language Reference* 15, 270

putchar: *Language Reference* 15, 271

puts: *Language Reference* 15, 272

Q

qsort: *Language Reference* 18, 273

Qualifiers, implementation data: *Language Reference* 420

Queues
 process: *User Guide* 176; *Tools Reference* 138, 312
 timer: *Tools Reference* 312

Quit
 debugger: *Tools Reference* 138
 simulator: *Tools Reference* 311

Quotient, of division: *Language Reference* 197

R

R-mode programs: *Tools Reference* 108

raise: *Language Reference* 12, 274

RAM: *User Guide* 226; *Tools Reference* 92, 98
 external, configuring for: *User Guide* 181, 182
 on-chip
 configuring for: *User Guide* 181, 182
 improve use of: *Performance note* 1, 5

rand: *Language Reference* 18, 275

RAND_MAX: *Language Reference* 19

Random numbers: *Language Reference* 275
 seeding: *Language Reference* 304

Read
 character from file: *Language Reference* 124
 current time: *Language Reference* 337
 formatted input: *Language Reference* 151, 281
 formatted string: *Language Reference* 305
 from file: *Language Reference* 276
 from file stream: *Language Reference* 140
 from keyboard: *Language Reference* 172
 line
 from stdin: *Language Reference* 173
 from stream: *Language Reference* 126
 MS-DOS registers: *Language Reference* 282
 strobe: *Tools Reference* 184

read: *Language Reference* 26, 276

Read/write pointer, position: *Language Reference* 159

Real-time programming: *User Guide* 5

realloc: *Language Reference* 18, 277

Reduced library: *User Guide* 224; *Language Reference* 3

io related functions: *Language Reference* 17

performance considerations: *Performance note* 2

Redundant store elimination: *Optimizing Compiler Guide* 43

REFRESH: *Tools Reference* 122, 142

Refresh period: *Tools Reference* 184

register: *Language Reference* 403, 419; *Optimizing Compiler Guide* 8

Registers: *Language Reference* 419

Areg: *User Guide* 134

assigning value: *Tools Reference* 312

Bptr0: *User Guide* 134

Bptr1: *User Guide* 134

Breg: *User Guide* 134

Clock0: *User Guide* 134

Clock1: *User Guide* 134

Creg: *User Guide* 134

displayed on Monitor page: *User Guide* 135

Error: *User Guide* 134

FPErr: *User Guide* 134

Fptr0: *User Guide* 134

Fptr1: *User Guide* 134

HaltOnError: *User Guide* 134

lptr: *User Guide* 134

memory dump: *Tools Reference* 176

ParityAddr: *User Guide* 134

ParityError: *User Guide* 134

Tptr0: *User Guide* 134

Tptr1: *User Guide* 134

Wdesc: *User Guide* 134

RELOCATE: *Tools Reference* 122, 141, 146

Remainder, of division: *Language Reference* 197

remove: *Language Reference* 15, 278

rename: *Language Reference* 15, 279

Reopen file: *Language Reference* 144

rep: *User Guide* 89

Replication, in configuration language: *User Guide* 89

reserved: *User Guide* 67, 69, 179, 181; *Performance note* 6

Reserved channels, in occam equivalent processes: *User Guide* 216

Reserved words, configuration language: *User Guide* 261

Reset: *User Guide* 111, 276; *Tools Reference* 116

use when debugging: *User Guide* 113

Reset

channel: *Language Reference* 83

file pointer: *Language Reference* 157

Restarting programs: *Language Reference* 114

RESUME: *Tools Reference* 122, 142, 145

Resume program

from debugger: *Tools Reference* 132

from simulator: *Tools Reference* 310

ret instruction: *Language Reference* 394

RETRACE: *Tools Reference* 122, 141, 146

rewind: *Language Reference* 15, 280

Right shift: *Tools Reference* 8; *Performance note* 25

ROM: *Tools Reference* 92, 98, 195

ROM bootable code: *User Guide* 225

processing configurations: *User Guide* 226

Root transputer: *User Guide* 276

and debugger: *User Guide* 112

debugging: *Tools Reference* 107

loading over: *Tools Reference* 317

root.processor.type: *Tools Reference* 198

routecost: *User Guide* 67, 69, 186, 189, 190

router: *User Guide* 67, 69

ROUTER_ORDER: *User Guide* 185

Run queues: *User Guide* 135

displaying: *Tools Reference* 138, 312

Running programs, introduction: *User Guide* 26

Runtime

dynamic code loading: *User Guide* 231

errors, fatal: *Language Reference* 32

library: *User Guide* 80, 96; *Language Reference* 3

introduction: *User Guide* 13

startup system

introduction: *User Guide* 14

modifying: *Language Reference* 357

performance considerations: *Performance note* 2

S

Scalar types, implementation: *Language Reference* 395

Scalar workspace: *Tools Reference* 92

scanf: *Language Reference* 15, 281

SCHAR_MAX: *Language Reference* 9

SCHAR_MIN: *Language Reference* 9

Scheduling lists. See Process queues; Run queues

Scope rules: *User Guide* 145

Screen definitions: *Tools Reference* 414

Screen size: *Tools Reference* 414

SEARCH: *Tools Reference* 147

Search, array: *Language Reference* 65

Search path: *User Guide* 34

#include: *Tools Reference* 14

configurer: *Tools Reference* 54

conventions: *Tools Reference* 326

icc: *Tools Reference* 7

SEEK_CUR: *Language Reference* 16

SEEK_END: *Language Reference* 16

SEEK_SET: *Language Reference* 16

Segment ordering: *User Guide* 73

Segment re-location: *User Guide* 73

segread: *Language Reference* 29, 282

Select process: *Tools Reference* 136

Select source file: *Tools Reference* 127

Selective linking: *Tools Reference* 228

Selective loading, libraries: *Tools Reference* 210

SemAlloc: *User Guide 62; Language Reference 25, 283*

semaphor.h: *User Guide 51, 62; Language Reference 22, 25*

Semaphore, structure type: *User Guide 50; Language Reference 25*

Semaphore: *User Guide 48*
 acquiring: *Language Reference 286*
 allocating: *Language Reference 283*
 initializing: *Language Reference 284*
 releasing: *Language Reference 285*

SEMAPHOREINIT: *Language Reference 25*

SemInit: *User Guide 62; Language Reference 25, 284*

SemSignal: *User Guide 62; Language Reference 25, 285*

SemWait: *User Guide 62; Language Reference 25, 286*

Separate compilation: *User Guide 276*

Sequential programming: *User Guide 6*

Serial links: *User Guide 3*

Server: *User Guide 26, 277*

server transaction: *Language Reference 4, 26, 287*

Session manager: *User Guide 277; Tools Reference 284, 288*
 configuration file: *Tools Reference 286*

Set file pointer: *Language Reference 155*

Set program locale: *Language Reference 9*
See also Locale

set_abort_action: *Language Reference 31, 36, 290, 427*

set_host link: *Language Reference 364*

setbuf: *Language Reference 15, 291*

setconf.inc: *User Guide 68; Tools Reference 53*

setjmp: *Language Reference 12, 292*

setjmp.h: *Language Reference 12*

setlocale: *Language Reference 9, 293*

setvbuf: *Language Reference 15, 294*

Shift right: *Tools Reference 8; Performance note 25*

short: *Language Reference 380; Performance note 17, 19*

short int, default promotion: *Language Reference 382*

Show debugging messages: *Tools Reference 139*

SHRT_MAX: *Language Reference 9*

SHRT_MIN: *Language Reference 9*

sig_atomic_t: *Language Reference 12*

SIG_DFL: *Language Reference 12*

SIG_ERR: *Language Reference 12*

SIG_IGN: *Language Reference 12*

SIGABRT: *Language Reference 12, 296, 423*

SIGALRM: *Language Reference 13, 296, 423, 424*

SIGEGV: *Language Reference 296*

SIGFPE: *Language Reference 12, 296, 423*

SIGILL: *Language Reference 12, 296, 423*

SIGINT: *Language Reference 12, 423*

SIGIO: *Language Reference 12, 296, 423, 424*

SIGLOST: *Language Reference 13, 296, 423, 424*

Signal
 handler: *Language Reference 36*
 handling: *Language Reference 295*
 constants: *Language Reference 12*
 functions: *Language Reference 12*
 macros: *Language Reference 12*
 types: *Language Reference 12*
 raise: *Language Reference 274*

signal: *Language Reference 12, 295, 423*

signal.h: *Language Reference 12*

signed: *Language Reference 379, 383*

signed char: *Language Reference 380, 395*

signed int: *Language Reference 396*

signed long: *Language Reference 396*

signed short: *Language Reference 395*

Signedness of char: *Tools Reference 8; Performance note 19, 25*

SIGPIPE: *Language Reference 12, 296, 423, 424*

SIGSEGV: *Language Reference 423, 424*

SIGSERV: *Language Reference 12*

SIGSTERM: *Language Reference 12*

SIGSYS: *Language Reference 13, 296, 423, 424*

SIGTERM: *Language Reference 296, 423, 424*

SIGURG: *Language Reference 12, 296, 423, 424*

SIGUSR1: *Language Reference 13, 296, 423, 424*

SIGUSR2: *Language Reference 13, 296, 423, 424*

SIGUSR3: *Language Reference 13, 296, 423, 424*

SIGWINCH: *Language Reference 13, 296, 423, 424*

Simulator: *User Guide 29; Tools Reference 303*
 batch command files: *Tools Reference 313*
 batch commands: *Tools Reference 313*
 batch mode: *Tools Reference 313*
 booting program: *Tools Reference 311*
 command definitions: *Tools Reference 307–313*
 summary: *Tools Reference 307*
 command line: *Tools Reference 303*
 commands: *Tools Reference 306*
 errors: *Tools Reference 314*
 options: *Tools Reference 304*
 starting a program: *Tools Reference 309*
 use in debugging: *User Guide 144*

sin: *Language Reference 11, 297*

sinf: *Language Reference 27, 298*

Single processor program, memory map: *Performance note 7*

Single step execution: *User Guide 145*

sinh: *Language Reference 11, 299*

sinhf: *Language Reference 27, 300*

size: *User Guide* 68, 90; *Tools Reference* 346, 371; *Language Reference* 391

size_t: *Language Reference* 13, 15, 19, 21

sizeof. See **size_t**

Skip load
example: *User Guide* 113
in debugging: *User Guide* 123

Skip loader: *User Guide* 26; *Tools Reference* 317
command line: *Tools Reference* 318
command line options: *Tools Reference* 318
errors: *Tools Reference* 321

Skipping channels: *Language Reference* 260

Soft channels: *User Guide* 77, 277

Software virtual routing: *User Guide* 77
control of: *User Guide* 185
disable: *User Guide* 78

Sort: *Language Reference* 273

Source character set: *Language Reference* 402

Source level debugging: *User Guide* 129

sourcefile: *Tools Reference* 346, 372

Space, optimizing compilation: *Optimizing Compiler Guide* 6

Space character
printable: *Language Reference* 189
test for: *Language Reference* 7, 191

Spanning tree, network: *User Guide* 188

sprintf: *Language Reference* 15, 17, 301

sqrt: *Language Reference* 11, 302

sqrtf: *Language Reference* 27, 303

Square root: *Language Reference* 302

srand: *Language Reference* 18, 304

srecord. See **output.format**

sscanf: *Language Reference* 15, 17, 305

Stack: *User Guide* 217
checking: *Tools Reference* 5, 16; *Performance note* 11
for dynamic code loading: *User Guide* 242
for runtime startup: *Language Reference* 363
freeing: *User Guide* 54
layout: *Performance note* 11, 20
overflow: *Language Reference* 32
overflow detection: *User Guide* 217
placing in on-chip RAM: *User Guide* 217; *Performance note* 8
position in memory: *User Guide* 71, 179; *Tools Reference* 54, 86; *Performance note* 6, 7
size: *User Guide* 71
usage: *Language Reference* 212

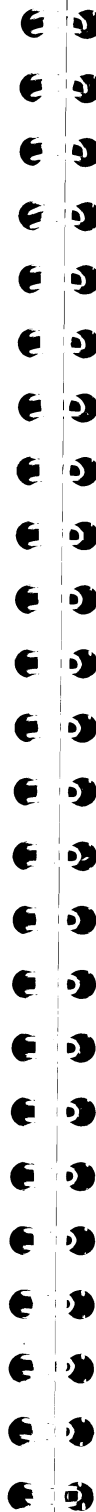
stack.buffer: *Tools Reference* 89

Standard definitions: *Language Reference* 13

Standard error: *User Guide* 277
writing error message: *Language Reference* 233

Standard input: *User Guide* 277; *Language Reference* 281

Standard memory configuration: *Tools Reference* 185



Standard output: *User Guide* 277; *Language Reference* 237, 271, 352
writing to: *Language Reference* 272

Standards, file extensions: *Tools Reference* 327

start.offset: *Tools Reference* 200

starttrd.lnk: *User Guide* xxiii

startup.h: *Language Reference* 368

startup.lnk: *User Guide* xxiii

Statements, implementation data: *Language Reference* 421

Static area: *User Guide* 205
pointer: *User Guide* 206
position in memory: *User Guide* 71, 179; *Tools Reference* 54, 86; *Performance note* 6, 7
requirement: *User Guide* 206
dynamic code loading: *User Guide* 242
runtime startup initialization: *Language Reference* 367

Static data: *Tools Reference* 87
access: *Performance note* 21
memory map: *Tools Reference* 9

Static data layout: *Language Reference* 405; *Performance note* 9
constant: *Language Reference* 406; *Performance note* 10
local: *Language Reference* 405; *Performance note* 10

Static variables, memory map: *Tools Reference* 271

stdarg.h: *Language Reference* 13

stddef.h: *Language Reference* 13

stderr: *Language Reference* 402, 416

stdin: *Language Reference* 402, 416
get character: *Language Reference* 170
read line: *Language Reference* 173

stdio.h: *Language Reference* 14

stdioe.h: *Language Reference* 17

stdlib.h: *Language Reference* 17

stdout: *Language Reference* 402, 416

STOP error mode, debugging: *User Guide* 118

strcat: *Language Reference* 20, 306

strchr: *Language Reference* 20, 307

strcmp: *Language Reference* 20, 308

strcoll: *Language Reference* 20, 309

strcpy: *Language Reference* 20, 310

strcspn: *Language Reference* 20, 311

strerror: *Language Reference* 20, 312
return values: *Language Reference* 427

strftime: *Language Reference* 21, 313

String
appending: *Language Reference* 306, 317
compare: *Language Reference* 308, 311
compare and count: *Language Reference* 322
compare characters: *Language Reference* 318
convert to double: *Language Reference* 324
convert to long int: *Language Reference* 330

- convert to tokens: *Language Reference* 326
 - copy to array: *Language Reference* 310, 319
 - handling functions: *Language Reference* 20
 - length: *Language Reference* 316
 - transform by locale: *Language Reference* 331
 - String constants, syntax: *Language Reference* 384
 - string.h: *Language Reference* 20
 - strlen: *Language Reference* 20, 316
 - strncat: *Language Reference* 20, 317
 - strncmp: *Language Reference* 20, 318
 - strncpy: *Language Reference* 20, 319
 - strpbrk: *Language Reference* 20, 320
 - strrchr: *Language Reference* 20, 321
 - strspn: *Language Reference* 20, 322
 - strstr: *Language Reference* 20, 323
 - strtod: *Language Reference* 18, 324
 - strtok: *Language Reference* 20, 326
 - strtol: *Language Reference* 18, 328
 - strtoul: *Language Reference* 18, 330
 - Structures: *Language Reference* 380
 - avoiding workspace: *Performance note* 21
 - implementation: *Language Reference* 397; *Performance note* 12
 - syntax: *Language Reference* 385
 - strxfrm: *Language Reference* 20, 331
 - Subsystem: *User Guide* 111, 277
 - connecting: *Tools Reference* 116
 - reset: *Tools Reference* 318
 - wiring: *User Guide* 111
 - Sun 4: *User Guide* 9, 33
 - SunOS: *User Guide* 9, 33
 - Switch statement
 - implementation: *Language Reference* 404
 - optimizing: *Performance note* 17
 - Symbol data, listing: *Tools Reference* 241
 - Symbolic debugging: *User Guide* 129; *Tools Reference* 142
 - See also *Debugging information: User Guide* 117
 - Synchronised communication: *User Guide* 6
 - Synchronising processes: *User Guide* 48, 58
 - Synchronous process: *User Guide* 56
 - Syntax
 - configuration language: *User Guide* 266
 - notation: *Language Reference* 413
 - system: *Language Reference* 18, 332
 - System services: *User Guide* 111
- ## T
- T-mode programs: *Tools Reference* 108
 - T4 series, configurable memory interface: *Tools Reference* 177
 - T8 series, configurable memory interface: *Tools Reference* 177
 - Tail recursion optimization: *Optimizing Compiler Guide* 47

- Tail-call optimization: *Optimizing Compiler Guide* 47
- tan: *Language Reference* 11, 333
- tanf: *Language Reference* 27, 334
- tanh: *Language Reference* 11, 335
- tanhf: *Language Reference* 27, 336
- Target, transputer: *Tools Reference* 333
- Target files, for imakef: *Tools Reference* 254
- Target transputer: *User Guide* 10, 277; *Performance note* 2
 - command line options: *Tools Reference* 339
- TCOFF: *User Guide* 11, 25
 - listing files: *Tools Reference* 250
- Temporary file: *Language Reference* 338
 - names: *Language Reference* 16
- Terminal I/O, test for: *Language Reference* 184
- Terminate: *Language Reference* 109
 - configured processes: *User Guide* 98
 - configured programs: *Language Reference* 112, 115
 - process: *User Guide* 54
 - program: *Language Reference* 36
 - See also *abort; exit*
- terminate.heap.use: *User Guide* 207
- terminate.static.use: *User Guide* 207
- terminate_server: *Language Reference* 366
- Termination, invoking function at: *Language Reference* 50
- Text files, listing: *Tools Reference* 250
- textname: *Tools Reference* 346, 373
- Through-routing: *User Guide* 190, 192
 - example: *User Guide* 106
- Time: *Language Reference* 337
 - See also *Date and time conversion, formatted: Language Reference* 313
 - difference: *Language Reference* 101
 - optimizing compilation: *Optimizing Compiler Guide* 6
 - UTC: *Language Reference* 174
- time: *Language Reference* 21, 337
- time.h: *Language Reference* 21
- time_t: *Language Reference* 21
- TIMER, parameters: *User Guide* 200
- Timer: *User Guide* 63, 134
 - See also *Clock*
- Timer queues: *User Guide* 135
 - displaying: *User Guide* 176; *Tools Reference* 139, 312
- Timing data: *Tools Reference* 186
- Tm: *Tools Reference* 184
- TMP_MAX: *Language Reference* 16
- tmpfile: *Language Reference* 15, 338
- tmpnam: *Language Reference* 15, 339
- to_host_link: *User Guide* 243; *Language Reference* 28, 340
- to86: *Language Reference* 29, 341
- TOGGLE BREAK**: *Tools Reference* 145
- TOGGLE HEX**: *Tools Reference* 147

tolerance: *User Guide* 67, 69, 186, 189

tolower: *Language Reference* 7, 342

toolname: *Tools Reference* 346, 374

Toolset
 development cycle: *User Guide* 21
 documentation: *User Guide* xviii; *Tools Reference* xx; *Language Reference* ix; *Optimizing Compiler Guide* iii; *Performance note* iii
 conventions: *User Guide* xix; *Tools Reference* xxi; *Language Reference* xi; *Optimizing Compiler Guide* v; *Performance note* v
 features: *User Guide* 9
 file extensions: *User Guide* 30
 getting started: *User Guide* 41
 list of tools: *User Guide* 19
 performance techniques: *Performance note* 1
 program development: *User Guide* 21
 running benchmarks: *Performance note* 27
 standards: *Tools Reference* 325

TOP: *Tools Reference* 122, 141, 146

TOP OF FILE: *Tools Reference* 123, 147

toupper: *Language Reference* 7, 343

Tptr0: *User Guide* 134

Tptr1: *User Guide* 134

Traceback information, in ROM: *Tools Reference* 202

TRAM: *User Guide* 68, 112, 278; *Tools Reference* 317

trams.inc: *Tools Reference* 53

TRANSPUTER: *User Guide* 35; *Tools Reference* 286, 287, 292

Transputer
 accessing: *Tools Reference* 284
 on a remote host: *Tools Reference* 293
 on the local host: *Tools Reference* 293
 architecture: *User Guide* 4
 clock: *User Guide* 134, 136
 in real-time programming: *User Guide* 5
 inline code: *Tools Reference* 21
 instructions: *User Guide* 49; *Language Reference* 389
 prefixing: *Performance note* 1
 size option: *Language Reference* 391
 introduction: *User Guide* 3
 loading: *User Guide* 109
 master: *User Guide* 111
 module: *User Guide* 278
 networks: *User Guide* 5, 49
 parallel processing: *User Guide* 48
 products: *User Guide* 6
 root: *User Guide* 276
 simulator: *Tools Reference* 303
 targets: *User Guide* 277; *Tools Reference* 6, 333; *Performance note* 2
 command line options: *Tools Reference* 339
 timer: *User Guide* 134

Tree, network topology: *User Guide* 5

Trigraphs: *Tools Reference* 24; *Language Reference* 380, 386; *Optimizing Compiler Guide* 12

Type: *Language Reference* 382
 conversion: *Language Reference* 399
 implementation: *Language Reference* 395
 in configuration language: *User Guide* 86
 nodes: *User Guide* 91

qualifiers: *Language Reference* 382

signal handling: *Language Reference* 12

specifiers: *Language Reference* 379

type: *User Guide* 67, 90

U

u, integer suffix: *Language Reference* 380, 384

UCHAR_MAX: *Language Reference* 9

uglobal.h: *Language Reference* 368

UINT_MAX: *Language Reference* 9

ULONG_MAX: *Language Reference* 9

Unary operators: *Language Reference* 380

ungetc: *Language Reference* 15, 344

Unions: *Language Reference* 380
 implementation: *Language Reference* 399
 initialization: *Language Reference* 380, 386
 syntax: *Language Reference* 385

UNIVERSAL: *User Guide* 12; *Tools Reference* 7
 debugging: *User Guide* 118

Unix: *User Guide* 33; *Tools Reference* 325, 417

unlink: *Language Reference* 26, 345

Unresolved references: *Tools Reference* 228

unsigned: *Language Reference* 384

unsigned char: *Language Reference* 380, 395

unsigned int: *Language Reference* 396

unsigned long: *Language Reference* 380, 396

unsigned short: *Language Reference* 395

Unsupported options: *User Guide* 39; *Tools Reference* 326

Unused process pointer, warnings: *User Guide* 52

Up: *User Guide* 111

Update registers: *Tools Reference* 139

Upper case
 convert to: *Language Reference* 7
 convert to lower: *Language Reference* 342
 test for: *Language Reference* 7, 192

Usage checks: *User Guide* 278

Usage files, libraries: *User Guide* 274

use: *User Guide* 78

User link: *User Guide* 278; *Tools Reference* 284, 291

USHRT_MAX: *Language Reference* 9

UTC time: *Language Reference* 174

V

va_arg: *Language Reference* 13, 346

va_end: *Language Reference* 13, 348

va_list: *Language Reference* 13

va_start: *Language Reference* 13, 349, 350

Variable argument lists: *Language Reference* 13, 346, 381
 cleaning up: *Language Reference* 348

Variables
 built-in: *Language Reference* 391
 performance considerations: *Performance note* 20

VAXVMS: *User Guide* 9, 33, 34, 35

Vector space: *User Guide* 278; *Tools Reference* 92
 in mixed language programming: *User Guide* 213
 position in memory: *User Guide* 71, 179; *Tools Reference* 54, 88

vfprintf: *Language Reference* 15, 350

Virtual channel: *User Guide* 77
 example: *User Guide* 104

Virtual link: *User Guide* 77

Virtual memory: *Tools Reference* 227

Virtual routing: *User Guide* 77, 190
 control of: *User Guide* 185
 disable: *User Guide* 78; *Tools Reference* 52
 example: *User Guide* 106
 use of memory: *User Guide* 185

VME bus, motherboard: *User Guide* 111

VMS: *User Guide* 33, 35; *Tools Reference* 325, 416, 417

void: *Language Reference* 379, 383

volatile: *Language Reference* 379, 383, 406; *Optimizing Compiler Guide* 8; *Performance note* 10
 implementation: *Language Reference* 404

vprintf: *Language Reference* 15, 352

vsprintf: *Language Reference* 15, 17, 353

W

Wait
 See also *ProcAfter*; *ProcWait*
 connection: *Tools Reference* 185
 race: *Tools Reference* 185
 error: *Tools Reference* 191
 states: *Tools Reference* 185

Warnings
 See also *Error messages*
 selective suppression, *icc*: *Tools Reference* 16

Waveform diagrams: *Tools Reference* 188

wchar_t: *Language Reference* 13, 19

wcstombs: *Language Reference* 18, 354

wctomb: *Language Reference* 18, 355

Wdesc: *User Guide* 134

WdescIntSave: *User Guide* 134

Wide characters. See *Character*

Wired down: *User Guide* 111; *Tools Reference* 116

Wired subs: *User Guide* 111; *Tools Reference* 116

word: *Tools Reference* 346, 375

Workspace: *User Guide* 278
 See also *Stack*
 allocation, optimizing: *Optimizing Compiler Guide* 49
 freeing: *User Guide* 54
 in mixed language programming: *User Guide* 213

Worm: *User Guide* 278

Write
 character, to file: *Language Reference* 138, 270
 error message, to stderr: *Language Reference* 233

line, to stdout: *Language Reference* 272
 mode: *Tools Reference* 185
 string, to stream: *Language Reference* 139
 strobe: *Tools Reference* 184
 to file: *Language Reference* 356
 to memory, in *idebug*: *Tools Reference* 140
 to stream: *Language Reference* 160

write: *Language Reference* 26, 356

Write formatted string
 to file: *Language Reference* 134, 350
 to standard output: *Language Reference* 237
 to stdout: *Language Reference* 352
 to string: *Language Reference* 301, 353

Z

z, command line option: *User Guide* 39; *Tools Reference* 326