

CONVEX C Master Index
Document No. 720-002930-000

First Edition
May 1990

CONVEX Computer Corporation
Richardson, Texas USA

CONVEX C Master Index
First Edition

© 1990 CONVEX Computer Corporation
All rights reserved.

This document is copyrighted. This document may not, in whole or part, be copied, duplicated, reproduced, translated, stored electronically, or reduced to machine-readable form without prior written consent from CONVEX Computer Corporation.

Although the material contained herein has been carefully reviewed, CONVEX Computer Corporation (CONVEX) does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE PROGRAM DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS PROGRAM. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

CONVEX and the CONVEX logo ("C") are registered trademarks of CONVEX
Computer Corporation.

Printed in the United States of America

Index

- * (pointer) operator og-9-1
- `_convex_` macro ug-2-12
- `_convexc_` macro ug-2-12
- `_DATE_` macro acc-4-6, ug-2-12
- `_FILE_` macro acc-A-2, ug-2-12
- `_LINE_` macro acc-A-2, ug-2-12
- `_STDC_` macro ug-2-12
- `_TIME_` macro acc-4-6, ug-2-12
- `_unix_` macro ug-2-12
- `_assert` function lrm-5-3
- `_CONVEX_FLOAT_` macro ug-2-12
- `_CONVEX_SOURCE`
 - use lrm-1-2, lrm-1-3
- `_CONVEX_SOURCE` macro ug-2-12
- `_convexc` macro ug-2-12
- `_IEEE_FLOAT_` macro ug-2-12
- `_IOFBF` macro lrm-5-19
- `_IOLBF` macro lrm-5-19
- `_IONBF` macro lrm-5-19
- `_POSIX_ARG_MAX` macro lrm-5-9
- `_POSIX_CHILD_MAX` macro lrm-5-9
- `_POSIX_LINK_MAX` macro lrm-5-9
- `_POSIX_MAX_CANON` macro lrm-5-9
- `_POSIX_MAX_INPUT` macro lrm-5-9
- `_POSIX_NAME_MAX` macro lrm-5-9
- `_POSIX_NGROUPS_MAX` macro lrm-5-9
- `_POSIX_OPEN_MAX` macro lrm-5-9
- `_POSIX_PATH_MAX` macro lrm-5-9
- `_POSIX_PIPE_BUF` macro lrm-5-9
- `_POSIX_SOURCE` macro lrm-1-2, lrm-1-3
- `_POSIX_SOURCE` macro ug-2-12
- `_setjmp` function
 - optimization with ug-A-30
- `_tolower`
 - function acc-A-4
 - macro lrm-5-5
- `_toupper`
 - function acc-A-4
 - macro lrm-5-5
- `#` (stringizing) operator
 - description acc-1-2
 - example acc-1-2
- `##` (pasting) operator
 - description acc-1-2
 - example acc-1-2
- `#define` lrm-6-1
- `#elif` acc-1-2
- `#include`
 - meaning of delimiters acc-4-5
 - statement lrm-6-3
- `#pragma`, causes no actions acc-4-6
- `#undef`, use of lrm-5-2
- `&` (address) operator og-9-1
- `()` (function) operator, operand of ug-A-29
- `+` (unary plus) operator ug-A-28
- `-` (unary minus) operator ug-A-28
- `-`, in `%[` scanlist, no special meaning acc-4-10
- `->` (structure pointer) operator og-9-1
- `-alias array_args` option og-9-1
- `-alias worst` compiler option acc-1-4
- `-asm`, compiler option lrm-7-1

- `-Enposix`, linker option ug-2-12
- `-ep` option og-9-9
- `-Eposix`, linker option ug-2-12
- `-no` option og-1-1, og-2-1
- `-O` option og-1-1
- `-OO` option og-1-1
- `-O1` option og-1-1, og-2-9
- `-O2` option og-1-1, og-1-2, og-3-1
- `-O3` option og-1-1, og-1-2, og-4-1
- `-p` option ug-5-2
- `-pb` option ug-5-2
- `-pg` option ug-5-2
- `-re` option og-4-5, og-4-6
- `-uo` option og-A-1
- .contact file, skipping first prompt by using
 - ug-B-3, acc-C-3, lrm-B-3, og-C-3
- `/**/` (pasting) operator acc-A-3
- `<<` (shift left) operator og-2-8
- `??!` trigraph of `|` acc-A-2
- `??'` trigraph of `^` acc-A-2
- `??(` trigraph of `[` acc-A-2
- `??)` trigraph of `]` acc-A-2
- `??-` trigraph of `~` acc-A-2
- `??/` trigraph of `backslash` acc-A-2
- `??<` trigraph of `{` acc-A-2
- `??=` trigraph of `#` acc-A-2
- `??>` trigraph of `}` acc-A-2

A

- abort function
 - action on open files acc-4-13
 - action on temporary files acc-4-13
 - description lrm-5-27
- abs function lrm-5-28
- acos function
 - description lrm-5-11
 - returns `acos(1)` on domain error acc-4-7
- Ada routines, calling lrm-3-9
- adb debugger ug-4-4
- adding automatic vector optimization og-5-3
- adding parallelization og-5-6
- adding vectorization og-5-3
- address operator og-9-1
- aggregate
 - definition ug-A-6
 - initialization of automatic storage class ug-A-6
- Alaska
 - technical assistance for, how to obtain acc-viii, lrm-viii, ug-viii
- algebraic simplification og-2-5
- alias og-7-1, og-B-1
- aliasing acc-1-4, og-6-4, og-7-1, og-9-1
- alignment of data types acc-4-4
- alternate
 - entries og-3-6, og-4-4
 - exits og-3-6, og-4-4, og-6-5
- ANSI C
 - aliasing acc-1-4
 - arithmetic conversions acc-1-3
 - beneficial aspects acc-1-1

ANSI C (cont)

- changes preventing compilation acc-A-1
- compatibility mode features lrm-1-3
- converting to acc-3-1
- function prototype
 - diagnostics acc-1-2
- future directions acc-A-5
- header file changes acc-A-4
- history acc-1-1
- improved diagnostics acc-1-2
- motivation for standard acc-1-1
- preprocessor enhancements acc-1-2
- semantic changes acc-A-2
- stricter type checking acc-1-3
- apparent recurrence og-3-10
- append mode
 - location of file position indicator acc-4-10
- arg_ptr_qual
 - compatibility mode actions ug-A-3
 - error message ug-A-44
- arg_ptr_ref
 - compatibility mode actions ug-A-3
 - error message ug-A-4
- argc, argument of function main acc-4-1
- argument pointer lrm-3-1
- argv, argument of function main acc-4-1
- arithmetic
 - conversions acc-A-2
 - expressions with float acc-A-3
- array
 - addresses lrm-2-11
 - aliasing og-6-4
 - alignment acc-4-4
 - character array initialization ug-A-27, ug-A-43
 - data type lrm-2-11
 - data type of index acc-4-4
 - dimensions lrm-2-11
 - efficient accesses og-3-4
 - initializing og-4-7
 - optimizing access og-6-9
 - oversubscripting og-9-3
 - parameters og-7-7
 - promoting to higher dimension og-8-3
 - storage of og-6-9
 - zero-length acc-A-2
- ASAP og-1-2, og-B-1
- asctime function lrm-5-33
- asin function
 - description lrm-5-11
 - returns asin(1) on domain error acc-4-7
- asm keyword lrm-7-1
- asm statement acc-A-2, lrm-7-1
- assembly-language debugger ug-4-4
- assembly-language statements lrm-7-1
- assert function
 - description lrm-5-3
- assert macro
 - form of message displayed acc-4-6
 - terminated by abort function acc-4-6
- assert.h contents lrm-5-3

- assignment substitution og-2-6
- associated documents
 - acc-vii, lrm-vii, ug-vii
 - how to order acc-viii, lrm-vii, ug-viii
- atan function
 - description lrm-5-11
 - returns $\pi/2$ on domain error acc-4-7
- atan2 function
 - description lrm-5-12
 - returns $\pi/2$ on domain error acc-4-7
- atexit function lrm-5-27
- atof function lrm-5-13, lrm-5-26
- atoi function lrm-5-26
- atol function lrm-5-26
- automatic self-allocating processors og-1-2

B

- backup C compiler acc-2-1, acc-3-1
- backward-compatible mode
 - converting to acc-B-1
 - defined acc-2-2
 - example lrm-1-3
 - long long int lrm-2-4
 - overview lrm-1-1
 - porting to acc-3-1
- balancing trees og-2-3
- bank conflict og-B-1
- base level og-2-1
- basic block
 - defined og-2-1, og-B-1
 - optimization og-2-5, ug-6-3
- begin_tasks directive lrm-A-4, og-4-7
- bibliography acc-vii, lrm-vii, ug-vii
- bit field
 - alignment acc-4-5
 - allocation acc-4-5
 - data types ug-A-19
 - initialization ug-A-21
 - maximum size ug-A-14
 - order of allocation acc-4-5
 - straddling byte boundaries acc-4-5
 - unqualified, sign of acc-4-5
 - zero width ug-A-48
- bit-shift operators og-2-8
- bitwise operation
 - on a signed integer acc-4-3
 - on an unsigned integer acc-4-3
- block-level optimization og-2-1
- blocking of signals acc-4-9
- break statement og-6-3, og-6-5
- bsearch function lrm-5-27
- BUFSIZ macro lrm-5-19

C

- cabs function lrm-5-13
- calling
 - Ada routines lrm-3-9
 - FORTTRAN functions lrm-3-7
 - runtime functions lrm-5-3

- calling sequence
 - standard lrm-3-2
- calloc function
 - description lrm-5-27
 - request size zero acc-4-13
- calls, standard function lrm-3-4
- Canada
 - technical assistance for, how to obtain acc-viii, lrm-viii, ug-viii
- case, maximum number acc-4-5
- casting, pointers and integers acc-4-4
- caution
 - improper use of `-alias array_args` leads to disaster og-7-7
 - no loop counters larger than 32 bits og-6-3
 - on `force_parallel` og-4-5
 - on `no_recurrence` og-3-11
 - on `no_side_effects` og-2-13
 - on reentrance og-4-5
- cc command line acc-3-1
- CCLIBS environment variable ug-2-13
- CCOPTIONS environment variable ug-2-13
- ceil function lrm-5-12
- chaining og-B-1
- char data type
 - alignment acc-4-4
 - description lrm-2-1, lrm-2-12
- CHAR_BIT macro lrm-5-8
- CHAR_MAX macro lrm-5-8
- CHAR_MIN macro lrm-5-8
- character
 - eight bits acc-4-2
- character array
 - initialization ug-A-27, ug-A-43
- character constant
 - initialization ug-A-9
 - maximum length ug-A-9
 - sign in preprocessor conditionals acc-4-5
 - value acc-4-2
 - wide ug-A-22
 - zero length ug-A-48
- character data lrm-2-12
- character input and output functions lrm-5-21
- character set
 - execution acc-4-2
 - illegal characters ug-A-17
 - source acc-4-2
- CHILD_MAX macro lrm-5-9
- chime og-B-1
- class_ignored
 - compatibility mode actions ug-A-3
 - error message ug-A-38
- clearerr function lrm-5-24
- CLK_TCK macro lrm-5-33
- clock function lrm-5-33
- clock_t type lrm-5-32
- CLOCKS_PER_SEC macro lrm-5-32
- code motion og-2-10, og-A-1
- Common C compiler
 - converting from acc-3-1
 - defined acc-2-1
- Common C compiler (cont)
 - incompatibilities acc-B-1
 - permits illegal code acc-B-1
- common subexpressions
 - elimination of og-2-7, og-2-11
- communication register og-B-1
- communication with the environment lrm-5-27
- comparison functions lrm-5-30
- compatibility
 - object file ug-2-15
- compatibility mode features lrm-1-3
- compatibility modes
 - backward-compatible acc-2-2, lrm-1-1
 - conforming acc-2-2, lrm-1-1
 - default acc-2-2, lrm-1-1
 - described acc-2-1
 - description lrm-1-1
 - differences lrm-1-1
 - examples lrm-1-2
 - extended acc-2-2, lrm-1-1
 - strict acc-2-2, lrm-1-1
- compatible type
 - description ug-A-30
 - pointer ug-A-29
- compiler
 - backup C acc-2-1, acc-3-1
 - Common C acc-2-1, acc-3-1
 - Common C, converting from acc-3-1
 - CONVEX C V3.0 acc-2-1
 - CONVEX C V4.0 acc-2-1
 - definition ug-1-1
 - mixed compatibility modes ug-2-15
 - Portable C acc-2-1
- compiler directives lrm-A-1
- compiler error
 - example ug-2-14
 - internal compiler error ug-A-10
 - machine serial number mismatch ug-A-9
 - no available memory ug-A-25
 - parser stack exceeded ug-A-42
 - program is too complex ug-A-32
- compiler limitations og-9-8
- compiler limits
 - CPU time limit ug-A-11
 - file size limit ug-A-14
 - maximum number of scalars in function ug-A-24
- compiling
 - conditional lrm-6-4
 - introduction ug-1-1
 - language specifications lrm-1-3
 - library systems lrm-1-3
 - multiple files ug-1-2
 - multiple modes lrm-1-3
 - one source file, example ug-1-2
 - simple program ug-1-1
 - single mode lrm-1-2
- compiling examples
 - backward-compatible mode lrm-1-3
 - conforming mode lrm-1-2
 - default mode lrm-1-2

- compiling examples (cont)
 - extended mode lrm-1-2
 - mixed mode lrm-1-4
 - strict mode lrm-1-2
- complex conditionals og-9-9
- concatenation functions lrm-5-29
- concurrent og-B-2
- concurrent operation og-2-2
- conditional compilation lrm-6-4
- conditional induction variables og-3-5
- conditionals
 - complex og-9-9
 - description og-3-6, og-5-4
 - in loop control statements og-6-4
- conforming compatibility mode
 - description acc-2-2, lrm-1-1
 - example lrm-1-2
- `const` type acc-1-4
- `const_not_init`
 - compatibility mode actions ug-A-3
 - error message ug-A-10
- constant
 - backslash-a* acc-A-3
 - backslash-x* acc-A-3
- constant propagation and folding og-2-6, og-2-9
- constants
 - integer acc-A-2
 - octal acc-A-3
- constructs
 - program, efficient og-6-1
- contact**
 - aborting the report ug-B-3, ug-B-7, acc-C-3, acc-C-7, lrm-B-3, lrm-B-7, og-C-3, og-C-7
 - editing the report ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - ending a response ug-B-4, acc-C-4, lrm-B-4, og-C-4
 - ending the report ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - including files in the report ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - invoking ug-B-1, ug-B-4, acc-C-1, acc-C-4, lrm-B-1, lrm-B-4, og-C-1, og-C-4
 - prerequisites ug-B-1, acc-C-1, lrm-B-1, og-C-1
 - prompts ug-B-4, acc-C-4, lrm-B-4, og-C-4
 - reporting problems ug-B-1, acc-C-1, lrm-B-1, og-C-1
 - restrictions on tilde-escape sequences ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - reviewing the report ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - skipping first prompt by using `.contact` file ug-B-3, acc-C-3, lrm-B-3, og-C-3
 - step-by-step discussion of prompts ug-B-4, acc-C-4, lrm-B-4, og-C-4
 - submitting dead.report file ug-B-3, acc-C-3, lrm-B-3, og-C-3
- contact** (cont)
 - submitting the report ug-B-6, acc-C-6, lrm-B-6, og-C-6
 - suspending the report ug-B-3, acc-C-3, lrm-B-3, og-C-3
 - tilde-escape sequences ug-B-4, acc-C-4, lrm-B-4, og-C-4
 - tips on using ug-B-3, acc-C-3, lrm-B-3, og-C-3
- conversion elimination og-A-2
- conversions
 - `float` truncation acc-4-3
 - `int` to `float` acc-4-3
 - integral acc-4-3
- converting
 - function pointers acc-A-1
 - object pointers acc-A-1
 - to ANSI C acc-3-1
- CONVEX consultant debugger ug-4-1
- CONVEX Extensions
 - `assert.h` lrm-5-3
 - `ctype.h` lrm-5-5
 - `math.h` lrm-5-13
 - `signal.h` lrm-5-16
 - `stdio.h` lrm-5-24
 - `stdlib.h` lrm-5-31
- convex macro ug-2-12
- CONVEX symbolic debugger ug-4-1
- ConvexOS considerations ug-2-13
- copy propagation og-2-10
- copying functions lrm-5-29
- `cos` function
 - description lrm-5-12
 - returns `cos(0)` on domain error acc-4-7
- `cosh` function
 - description lrm-5-12
 - returns `HUGE_VAL` on domain error acc-4-7
- counted loops og-6-2, og-6-4, og-6-7, og-7-4, og-7-5, og-7-6
- CPU
 - defined og-B-2
 - self-allocating og-1-2
- `cref` program ug-4-3
- critical region og-B-2
- cross-reference generator ug-4-3
- `csd` debugger og-1-3, ug-4-1
- `ctime` function lrm-5-33
- `ctype.h`
 - ANSI C changes acc-A-4
 - contents lrm-5-4
 - future changes acc-A-5
- `cuserid` function lrm-5-25
- customer support
 - telephone number for acc-viii, lrm-viii, ug-viii
- CXpa profiler ug-5-2

D

data representations
 FORTRAN lrm-3-5

data type
 array lrm-2-11
 char lrm-2-1, lrm-2-12
 double lrm-2-5
 enum lrm-2-4
 float lrm-2-5
 int lrm-2-1
 long lrm-2-1
 long double lrm-2-5
 long float lrm-2-7
 long int lrm-2-1
 long long lrm-2-3
 long long int lrm-2-3
 pointer lrm-2-8
 short lrm-2-1
 short int lrm-2-1
 string lrm-2-12
 void lrm-2-8

DBL_DIG macro lrm-5-6
 DBL_EPSILON macro lrm-5-7
 DBL_MANT_DIG macro lrm-5-6
 DBL_MAX macro acc-A-4, lrm-5-7
 DBL_MAX_10_EXP macro lrm-5-7
 DBL_MAX_EXP macro lrm-5-6
 DBL_MIN macro lrm-5-7
 DBL_MIN_10_EXP macro lrm-5-7
 DBL_MIN_EXP macro lrm-5-6
 devtid function lrm-5-13
 dead code elimination og-2-10
 dead.report file
 submitting ug-B-3, acc-C-3, lrm-B-3,
 og-C-3
 using -r option to submit ug-B-3,
 acc-C-3, lrm-B-3, og-C-3

debugger
 adb ug-4-4
 assembly-language ug-4-4
 consultant package ug-4-1
 csd ug-4-1
 symbolic ug-4-1
 tools ug-4-1
 use og-1-3

debugging optimized code og-5-1

declarators
 minimum number of acc-4-5

default compatibility mode
 description acc-2-2, lrm-1-1
 example lrm-1-2

dependency
 backward og-4-4
 defined og-3-6, og-B-2
 forward og-4-4
 loop-carried og-4-4
 loop-independent og-3-9, og-4-4

diagnostic messages ug-2-14
 difftime function lrm-5-33
 direct input and output functions lrm-5-22

directives

compiler lrm-A-1
 force_parallel og-4-5, og-4-6
 lint utility ug-3-2
 max_trips og-8-2
 no_recurrence og-3-10, og-3-11, og-6-8
 no_side_effects og-2-12
 optimization ug-6-8
 preprocessor lrm-6-1
 pstrip og-9-9
 scalar og-5-4, og-8-2, og-9-9
 select og-8-2, og-9-9
 synch_parallel og-4-4, og-4-5, og-4-6
 synch_parallel og-9-9
 unroll og-8-3
 vstrip og-9-9

distribution, loop og-3-3
 div function lrm-5-28
 div_t type lrm-5-26
 division
 integer, sign or remainder acc-4-3
 division_by_zero
 compatibility mode actions ug-A-3
 error message ug-A-10

do-while loops og-6-7

documentation
 ordering acc-viii, lrm-vii, ug-viii
 subscription service, how to apply acc-viii,
 lrm-vii, ug-viii

dollar_names
 compatibility mode actions ug-A-3
 error message ug-A-12

domain error
 math function return values acc-4-7

double
 alignment acc-4-4
 data type lrm-2-5, lrm-2-6
 double versus float og-6-1
 double-precision floating point lrm-2-6

dump
 post-mortem ug-4-2

dynamic selection og-9-9

E

EDOM
 macro lrm-5-5
 use lrm-5-11

efficient
 loops og-6-2
 program constructs og-6-1

elaborate conditionals og-9-9

elimination of common subexpressions og-2-7,
 og-2-11

empty declarations
 struct acc-A-2, acc-A-3
 union acc-A-2

end_tasks directive lrm-A-4, og-4-7

enhancing
 parallelization og-5-7
 vectorization og-5-4

- entry statement acc-A-2
 - enum
 - data type lrm-2-4
 - representation of acc-4-5
 - environment variables
 - CCLIBS ug-2-13
 - CCOPTIONS ug-2-13
 - environment, ConvexOS ug-2-13
 - envp, argument of function main acc-4-1
 - EOF macro lrm-5-19
 - ERANGE
 - macro lrm-5-5
 - use lrm-5-11
 - errno
 - use lrm-5-13
 - values of fgetpos and ftell acc-4-10
 - variable lrm-5-5
 - errno.h
 - contents lrm-5-5
 - future changes acc-A-5
 - error handling functions lrm-5-24
 - error message
 - d options ug-A-2
 - catalog ug-A-4
 - control ug-A-1
 - diagnostic options ug-A-1
 - overflow og-2-6
 - underflow og-2-6
 - error messages, controlled
 - arg_ptr_qual ug-A-3
 - arg_ptr_ref ug-A-3
 - class_ignored ug-A-3
 - const_not_init ug-A-3
 - division_by_zero ug-A-3
 - dollar_names ug-A-3
 - escape_range_sequence ug-A-3
 - float_suffix ug-A-3
 - func_ptr_math ug-A-3
 - function_storage_class ug-A-3
 - implicit_decl ug-A-3
 - incomplete_enum ug-A-3
 - integer_overflow ug-A-3
 - long_long_suffix ug-A-3
 - no_external_declaration ug-A-3
 - nothing_declared ug-A-3
 - null_effect_expression ug-A-3
 - obsolescent ug-A-3
 - old_form_assign ug-A-3
 - old_octal_digits ug-A-3
 - sizeof_bitfield ug-A-3
 - strict_syntax ug-A-3
 - error reporting ug-B-1, acc-C-1, lrm-B-1, og-C-1
 - error utility ug-3-4
 - escape sequences
 - description ug-A-46
 - value of undefined acc-4-2
 - escape_range_sequence
 - compatibility mode actions ug-A-3
 - error message ug-A-13
 - evaluation order og-2-3
 - example
 - array of pointers to functions ug-A-6
 - const-qualifier ug-A-44
 - function parameter is pointer to function ug-A-15
 - function returns pointer to array of int ug-A-16
 - function returns pointer to functions returning int ug-A-16
 - incomplete type ug-A-5
 - tasking directives ug-A-6
 - execution stream og-B-2
 - exit function lrm-5-27
 - EXIT_FAILURE macro lrm-5-25
 - EXIT_SUCCESS macro lrm-5-25
 - exp function lrm-5-12
 - expression evaluation order acc-A-2, og-2-3
 - extended compatibility mode
 - description lrm-1-1
 - differences with Common C acc-A-1
 - example lrm-1-2
 - porting to acc-3-2
 - extern storage class
 - block scope acc-A-1
 - changing from extern to static ug-A-35
 - changing to static from extern ug-A-33, ug-A-34
 - initialization in function ug-A-11
- F**
- fabs function lrm-5-12
 - fastmath.h header og-2-6
 - fclose function lrm-5-20
 - fdopen function lrm-5-25
 - feof function lrm-5-24
 - ferror function lrm-5-24
 - fflush function lrm-5-20
 - fgetc function lrm-5-21
 - fgetpos function lrm-5-23
 - fgets function lrm-5-21
 - figure
 - char representation lrm-2-2
 - character data representation lrm-2-12
 - character string representation lrm-2-13
 - compiler and linker interactions ug-1-4
 - double-precision floating representation lrm-2-7
 - int representation lrm-2-3
 - linking library routines ug-1-5
 - long long representation lrm-2-4
 - multiple source files ug-1-3
 - pointer representation lrm-2-8
 - role of the compiler ug-1-1
 - sample program, file2.c ug-1-3
 - sample program, prog2.c ug-1-3
 - short int representation lrm-2-2
 - single-precision floating representation lrm-2-6
 - stack layout lrm-3-3
 - top of the runtime stack lrm-3-2

FILE

- description of lrm-4-2
- type lrm-5-20
- file access functions lrm-5-20
- file input and output
 - concepts lrm-4-1
 - support for fully buffered acc-4-10
 - support for line buffered acc-4-10
 - support for unbuffered acc-4-10
- file location
 - meaning of #include delimiters acc-4-5
- file manipulation paradigm lrm-4-1
- file name
 - length of acc-4-10
 - naming conventions ug-2-2
- file positioning functions lrm-5-23
- file types and access modes lrm-4-2
- FILENAME_MAX macro lrm-5-19
- fileno function lrm-5-25
- flags
 - code generation ug-2-4
 - compatibility ug-2-6
 - diagnostic ug-2-6
 - miscellaneous ug-2-11
 - optimization ug-2-9
 - preprocessor ug-2-3
 - utility support ug-2-7
- float
 - alignment acc-4-4
 - data type lrm-2-5
 - promotion to double acc-A-3
- float versus double og-6-1
- float_suffix
 - compatibility mode actions ug-A-3
 - error message ug-A-14
- float.h contents lrm-5-6
- floating-point
 - 32-bit lrm-2-5
 - 64-bit lrm-2-5
 - data types lrm-2-5
 - IEEE format lrm-2-5
 - native format lrm-2-5
 - overflow og-2-6
 - roundoff og-9-4
 - underflow og-2-6
 - versus integer og-6-1
- floating-point range
 - double lrm-2-5
 - float lrm-2-5
 - long double lrm-2-5
 - long float lrm-2-8
- floor function lrm-5-12
- FLT_DIG macro lrm-5-6
- FLT_EPSILON macro lrm-5-7
- FLT_MANT_DIG macro lrm-5-6
- FLT_MAX macro lrm-5-7
- FLT_MAX_10_EXP macro lrm-5-7
- FLT_MAX_EXP macro lrm-5-6
- FLT_MIN macro lrm-5-7
- FLT_MIN_10_EXP macro lrm-5-7
- FLT_MIN_EXP macro lrm-5-6

- FLT_RADIX macro lrm-5-6
- FLT_ROUNDS macro lrm-5-6
- fmod function
 - description lrm-5-12
 - domain error acc-4-7
- folding, constant og-2-6, og-2-9
- fopen function lrm-5-20
- FOPEN_MAX macro lrm-5-19
- for loops og-6-2
- force directives lrm-A-6
- force_parallel directive lrm-A-6, og-4-5, og-4-6
- force_vector directive lrm-A-6
- formal parameters
 - typedef names acc-A-2
- formatted input and output functions lrm-5-21
- FORTRAN
 - mixing IO with C lrm-3-9
- fpos_t type lrm-5-20
- fprintf function
 - description lrm-5-21
 - example lrm-4-1
 - meaning of %p acc-4-10
- fputc function lrm-5-21
- fputs function lrm-5-22
- frame pointer lrm-3-1
- fread function lrm-5-22
- free function lrm-5-27
- freopen function lrm-5-20
- frexp function lrm-5-12
- fscanf function
 - description lrm-5-21
 - meaning of %p acc-4-10
- fseek function
 - associated macros lrm-5-19
 - description lrm-5-23
- fsetpos function lrm-5-23
- ftell function lrm-5-23
- func_ptr_math
 - compatibility mode actions ug-A-3
 - error message ug-A-32
- function
 - arguments lrm-3-4
 - definition ug-1-1
 - incomplete return type ug-A-15
 - initialization ug-A-16
 - parameter storage class ug-A-19
 - stack layout lrm-3-1
 - static storage class ug-A-15
 - storage class ug-A-18
- function calls
 - in loops og-3-6
 - within loops og-4-4, og-4-5
- function names
 - lrm-3-4
 - accessing function names from C lrm-3-5
- function pointers
 - converting acc-A-1
 - different types acc-A-1
- function prototype
 - definition ug-A-16

function prototype (cont)
 diagnostics acc-1-2
 efficiency acc-1-3
 function_storage_class
 compatibility mode actions ug-A-3
 error message ug-A-15
 function-level optimization og-2-9, ug-6-3
 function-like macros versus functions lrm-5-2
 functional unit og-B-2
 functions
 character input and output lrm-5-21
 close acc-A-4
 communication with the environment
 lrm-5-27
 compared to function-like macros lrm-5-2
 comparison lrm-5-30
 concatenation lrm-5-29
 copying lrm-5-29
 creat acc-A-4
 direct input and output lrm-5-22
 error-handling lrm-5-24
 fdopen acc-A-4
 file access lrm-5-20
 file positioning lrm-5-23
 fileno acc-A-4
 formatted input and output lrm-5-21
 implicit declaration lrm-5-3
 integer arithmetic lrm-5-28
 lseek acc-A-4
 main acc-4-1
 memory management lrm-5-27
 miscellaneous lrm-5-31
 multibyte character lrm-5-28
 multibyte string lrm-5-28
 open acc-A-4
 operations on files lrm-5-20
 pseudo-random sequence generation
 lrm-5-26
 read acc-A-4
 sacos acc-A-4
 sasin acc-A-4
 satan acc-A-4
 satan2 acc-A-4
 scabs acc-A-4
 scos acc-A-4
 scosh acc-A-4
 search lrm-5-30
 searching and sorting lrm-5-27
 sexp acc-A-4
 sfabs acc-A-4
 shypot acc-A-4
 signal acc-A-4
 signal handlers acc-A-4
 SIGNALS_IN_PROG acc-A-4
 slog acc-A-4
 spow acc-A-4
 ssin acc-A-4
 ssinh acc-A-4
 ssqrt acc-A-4
 stan acc-A-4
 stanh acc-A-4

functions (cont)
 string conversion lrm-5-26
 string handling lrm-5-29
 time lrm-5-32
 time conversion lrm-5-33
 time manipulation lrm-5-33
 unlink acc-A-4
 variable number of parameters acc-A-1
 write acc-A-4
 further reference acc-vii, lrm-vii, ug-vii
 fwrite function lrm-5-22

G

gamma function lrm-5-13
 generic pointer lrm-2-8
 getc function lrm-5-22
 getchar function lrm-5-22
 getenv function
 description lrm-5-27
 set of environment names acc-4-13
 gets function lrm-5-22
 Global Optimization og-2-9
 global variables og-2-12, og-4-5
 gmtime function lrm-5-33
 granularity og-4-1, og-B-2

H

Hawaii
 technical assistance for, how to obtain
 acc-viii, lrm-viii, ug-viii
 header files
 assert.h lrm-5-3
 ctype.h acc-A-4, lrm-5-4
 errno.h lrm-5-5
 float.h lrm-5-6
 limits.h lrm-5-8
 locale.h lrm-5-10
 math.h acc-A-4, lrm-5-11
 setjmp.h lrm-5-14
 signal.h acc-A-4, lrm-5-15
 stdarg.h lrm-5-18
 stddef.h lrm-5-18
 stdio.h acc-A-4, lrm-5-19
 stdlib.h lrm-5-25
 string.h lrm-5-29
 strings.h acc-A-4
 time.h lrm-5-32
 use of lrm-5-3
 hidden aliasing og-9-1
 hoisting og-2-11, og-4-3, og-B-2
 hoisting and sinking og-3-4
 HUGE macro acc-A-4
 HUGE_VAL macro lrm-5-11
 HUGE_I macro acc-A-4
 hypot function lrm-5-13

I

- idcvtd function lrm-5-13
- identifiers
 - external acc-4-2
 - internal acc-4-2
 - significant characters acc-4-2
- IEEE format
 - double-precision lrm-2-7
 - single-precision lrm-2-6
- if else statements og-9-9
- if statements og-3-6, og-5-4
- illegal subscripts og-9-3
- implicit function declaration lrm-5-3
- implicit_decl
 - compatibility mode actions ug-A-3
 - error message ug-A-21
- incompatibilities of common C acc-B-1
- incomplete type
 - definition ug-A-5
 - example ug-A-5, ug-A-45
- incomplete_enum
 - (See also: incomplete type) ug-A-13
 - compatibility mode actions ug-A-3
 - error message ug-A-13
- indent utility ug-3-3
- index function lrm-5-31
- induction variables og-6-2, og-6-3, og-6-4, og-6-7, og-7-4, og-7-5
- information directives lrm-A-2
- inhibitors of parallelization og-4-4, ug-6-5
- inhibitors of vectorization og-3-6, ug-6-4
- initialization
 - character array ug-A-27
 - character string ug-A-43
 - function ug-A-16
 - legal initializations ug-A-18
 - struct acc-A-3
 - typedef variables ug-A-45
- initializers
 - bit field ug-A-21
- input and output
 - mixing C and FORTRAN lrm-3-9
 - program lrm-4-3
- instruction scheduling og-2-2, og-2-7
- int
 - alignment acc-4-4
 - data type lrm-2-1
- INT_MAX macro lrm-5-8
- INT_MIN macro lrm-5-8
- integer arithmetic functions lrm-5-28
- integer constants acc-A-2
- integer division, sign of remainder acc-4-3
- integer overflow og-2-6
- integer range
 - char lrm-2-2
 - enum lrm-2-4
 - int lrm-2-2
 - long lrm-2-2
 - long int lrm-2-2
 - long long lrm-2-3
- integer range (cont)
 - long long int lrm-2-3
 - short lrm-2-2
 - short int lrm-2-2
- integer representation
 - 16-bit lrm-2-1
 - 32-bit lrm-2-1
 - 8-bit lrm-2-1
 - two's complement acc-4-3
- integer type lrm-2-1
- integer vs. floating-point og-6-1
- integer_overflow
 - compatibility mode actions ug-A-3
 - error message ug-A-10
- integral constant expressions, defined ug-A-40
- integral conversions acc-4-3
- interactive devices, location acc-4-1
- interchange, loop og-3-4
- invariant computation og-2-10
- ipc/network operational error messages
 - acc-4-13
- ipow function lrm-5-13
- ircvtr function lrm-5-14
- isalnum function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
 - returns true for acc-4-6
- isalpha function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
 - returns true for acc-4-6
- isascii
 - function acc-A-4
 - macro lrm-5-5
- isctrl function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
 - returns true for acc-4-6
- isdigit function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
- isgraph function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
- islower function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
 - returns true for acc-4-6
- isprint function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
 - returns true for acc-4-6
- ispunct function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
- isspace function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10
- isupper function
 - description lrm-5-4
 - LC_CTYPE lrm-5-10

isupper function (cont)
 returns true for acc-4-6
 isxdigit function
 description lrm-5-4
 LC_CTYPE lrm-5-10
 iterating by zero og-9-4
 iteration count og-3-2

J

j0 function lrm-5-14
 j1 function lrm-5-14
 jmp_buf type lrm-5-14
 jn function lrm-5-14

K

kill function lrm-5-17

L

L_ctermid macro lrm-5-25
 L_cuserid macro lrm-5-25
 L_tmpnam macro lrm-5-19
 label, scope of ug-A-12
 labs function lrm-5-28
 LC_ALL macro lrm-5-10
 LC_COLLATE macro lrm-5-10
 LC_CTYPE macro lrm-5-10
 LC_MONETARY macro lrm-5-10
 LC_NUMERIC macro lrm-5-10
 LC_TIME macro lrm-5-10
 LCD og-B-3
 lconv structure lrm-5-10
 LDBL_DIG macro lrm-5-6
 LDBL_EPSILON macro lrm-5-7
 LDBL_MANT_DIG macro lrm-5-6
 LDBL_MAX macro lrm-5-7
 LDBL_MAX_10_EXP macro lrm-5-7
 LDBL_MAX_EXP macro lrm-5-7
 LDBL_MIN macro lrm-5-7
 LDBL_MIN_10_EXP macro lrm-5-7
 LDBL_MIN_EXP macro lrm-5-6
 ldexp function lrm-5-12
 ldiv function lrm-5-28
 ldiv_t type lrm-5-26
 libraries, general ug-1-5
 LID og-B-3
 limitations of optimization og-9-1
 limiting directives lrm-A-6
 limits.h contents lrm-5-8
 LINK_MAX macro lrm-5-9
 linker
 description ug-1-4
 specifying libraries ug-2-12
 usage ug-2-12
 linker options
 -Enposix ug-2-12
 -Eposix ug-2-12
 on cc command line ug-2-12
 lint utility

lint utility (cont)
 converting to backward-compatible mode
 acc-3-2
 description ug-3-1
 directives ug-3-2
 options ug-3-2
 locale
 available acc-4-2
 locale.h
 contents lrm-5-10
 future changes acc-A-5
 localeconv function
 description lrm-5-11
 LC_MONETARY lrm-5-10
 LC_NUMERIC lrm-5-10
 localtime function lrm-5-33
 log function
 description lrm-5-12
 returns log(|x|) on domain error acc-4-7
 log10 function
 description lrm-5-12
 returns log10(|x|) on domain error acc-4-7
 long
 alignment acc-4-4
 bit shift operand acc-A-3
 data type lrm-2-1
 long double
 alignment acc-4-4
 data type lrm-2-5
 long float
 data type acc-A-1, lrm-2-7
 long int data type lrm-2-1
 long long data type lrm-2-3
 long long int
 data type lrm-2-3
 function parameter, backward-compatible
 mode lrm-2-4
 long_long_suffix
 compatibility mode actions ug-A-3
 error message ug-A-22
 LONG_MAX macro lrm-5-8
 LONG_MIN macro lrm-5-8
 longjmp function
 description lrm-5-14
 optimization with ug-A-30
 loop
 constant og-B-3
 distribution og-3-3, og-B-3
 induction variable og-B-3
 interchange og-3-4, og-B-3
 invariant og-4-2, og-B-3
 unrolling og-6-2, og-8-2
 with function calls og-4-5
 loop exits
 multiple og-6-5
 loop-carried dependency og-3-6, og-4-4, og-B-3,
 ug-6-5
 loop-independent dependency og-3-6, og-3-9,
 og-B-3
 lowest optimization level og-2-1
 lpow function lrm-5-14

lvalue, definition of ug-A-23

M

machine-dependent optimizations og-1-1,
og-2-1, og-2-2, ug-6-1, ug-6-2
machine-independent optimizations og-1-1,
og-2-1
macro
 parameter substitution acc-A-3
 recursive acc-A-3
main function acc-4-1
make utility ug-3-3
malloc function
 description lrm-5-27
 request size zero acc-4-13
math domain error return values lrm-5-13
math functions
 setting of `errno` acc-4-7
 underflow acc-4-7
 values returned on domain errors acc-4-7
math.h
 ANSI C changes acc-A-4
 contents lrm-5-11
 future changes acc-A-5
MAX_CANON macro lrm-5-9
MAX_INPUT macro lrm-5-9
max_trips directive lrm-A-2, og-8-2
MB_CUR_MAX macro lrm-5-25
MB_LEN_MAX macro lrm-5-8
mbilen function lrm-5-28
mbstowcs function lrm-5-28
mbtowc function lrm-5-28
memchr function lrm-5-30
memcmp function lrm-5-30
memcpy function lrm-5-29
memmove function lrm-5-29
memory accesses, efficient og-3-4
memory management functions lrm-5-27
memset function lrm-5-31
messages
 compiler ug-2-14
 diagnostic ug-2-14
miscellaneous functions lrm-5-31
misused
 directives og-9-6, og-9-9
 options og-9-6
mixed-mode
 example lrm-1-4
 operations og-6-1
mktime function lrm-5-33
modf function lrm-5-12
multibyte
 character functions lrm-5-28
 characters acc-4-2
 string functions lrm-5-28
multidimensional arrays og-6-7
multiple entries og-3-6, og-4-4
multiple exits og-3-6, og-4-4, og-6-5
multiple mode compiling lrm-1-3
multiprocessor og-B-4

mutual exclusion og-B-4

N

name space
 ordinary identifiers ug-A-36
 struct, union, and enum ug-A-35
 tags ug-A-41
NAME_MAX macro lrm-5-9
native format
 double-precision lrm-2-7
 single-precision lrm-2-6
next_task directive lrm-A-4, og-4-7
no_external_declaration
 compatibility mode actions ug-A-3
 error message ug-A-43
no_recurrence directive lrm-A-2, og-3-10,
og-3-11, og-6-8
no_side_effects directive lrm-A-3, og-2-12
nondeterminism in parallel code og-9-8
note
 -d option ug-A-2
 converting error messages ug-A-2
 converting errors to warnings ug-A-2
 IEEE 754 specifications lrm-2-5
 LCDs can prevent vectorization og-3-9
 long float is a CONVEX extension
 lrm-2-7
 long long int is a CONVEX extension
 lrm-2-3
 no_recurrence directive lrm-A-2
 on -parens og-3-2, og-9-6
 on -parens implicit og-2-4
 on basic blocks og-2-1
 restrictions on tilde-escape sequences with
 contact ug-B-6, acc-C-6, lrm-B-6,
 og-C-6
 verify a program for each optimization
 level og-5-1
nothing_declared
 compatibility mode actions ug-A-3
 error message ug-A-26
null characters
 appending to stream acc-4-10
NULL macro
 expansion acc-4-6
 use lrm-5-18, lrm-5-19, lrm-5-25, lrm-5-29,
 lrm-5-32
null_effect_expression
 compatibility mode actions ug-A-3
 error message ug-A-13

O
object file compatibility ug-2-15
object pointers, converting acc-A-1
obsolescent
 compatibility mode actions ug-A-3
 error message ug-A-27
octal constants acc-A-3
octal digits, 8 and 9 acc-A-1

- offsetof macro lrm-5-18
- old_form_assign
 - compatibility mode actions ug-A-3
 - error message ug-A-28
- old_octal_digits
 - compatibility mode actions ug-A-3
 - error message ug-A-27
- operation
 - bitwise, on a signed integer acc-4-3
 - bitwise, on an unsigned integer acc-4-3
 - right shift, negative signed integer acc-4-3
- operations on files lrm-5-20
- operators
 - old-style (=+), etc. acc-A-3
 - pasting (##) lrm-6-3
 - stringizing (#) lrm-6-3
- optimization
 - basic block og-2-5
 - basic-block ug-6-3
 - directives lrm-A-4, ug-6-8
 - function-level og-2-9, ug-6-3
 - global og-2-9
 - machine-dependent og-2-2, ug-6-1, ug-6-2
 - machine-independent og-1-1
 - options ug-6-7
 - parallel og-1-2, og-4-1, ug-6-2, ug-6-5
 - program-unit og-2-9
 - report ug-2-14, ug-6-6
 - scalar ug-6-1, ug-6-2
 - strategy og-5-1
 - tricks and tips og-8-1
 - user-directed ug-6-7
 - vector og-1-2, og-3-1, ug-6-1, ug-6-4
- Optimization Report og-3-12
- optimizing
 - array accesses og-6-9
 - C applications og-5-1
- options
 - code generation ug-2-4
 - compatibility ug-2-6
 - compatibility with other compilers ug-2-11
 - diagnostic ug-2-6
 - introduction ug-1-2
 - miscellaneous ug-2-11
 - optimization ug-2-9
 - preprocessor ug-2-3
 - utility support ug-2-7
- options
 - alias array_args ug-2-9
 - alias cautious ug-2-9
 - alias ptr_args ug-2-9
 - alias standard ug-2-9
 - alias worst acc-1-4, ug-2-9
 - asm ug-2-4
 - B ug-2-11
 - C ug-2-3
 - c ug-2-4
 - D ug-2-3
 - d ug-2-6, ug-A-1
 - db ug-2-8
 - ds ug-2-9

- options (cont)
 - E ug-2-3
 - ep ug-2-9
 - ext ug-2-7
 - extern distinct ug-2-4
 - extern same ug-2-4
 - fd ug-2-4
 - fi ug-2-4
 - float dp_const ug-2-4
 - float dp_ops ug-2-4
 - float sp_const ug-2-4
 - float sp_ops ug-2-4
 - fn ug-2-5
 - I ug-2-3
 - k ug-2-3
 - no ug-2-9
 - nw ug-2-6
 - O ug-2-9
 - o ug-2-11
 - O1 ug-2-10
 - O2 ug-2-10
 - O3 ug-2-10
 - O4 ug-2-10
 - or ug-2-6
 - p ug-2-8
 - pa ug-2-8
 - pab ug-2-8
 - par ug-2-8
 - parens explicit ug-2-5
 - parens ignore ug-2-5
 - parens implicit ug-2-5
 - pb ug-2-8
 - pcc ug-2-7
 - pg ug-2-8
 - re ug-2-5
 - r1 ug-2-10
 - S ug-2-5
 - sc ug-2-6
 - std lrm-1-2, ug-2-7
 - str ug-2-7
 - string read_only ug-2-5
 - string read_write ug-2-5
 - t1 ug-2-11
 - tm ug-2-5
 - U ug-2-3
 - uo ug-2-10
 - ur ug-2-10
 - va ug-2-10
 - vn ug-2-11
- ordering documentation
 - how to acc-viii, lrm-vii, ug-viii
- overflow og-2-6

P

- padding
 - data types acc-4-4
 - structures lrm-2-11
- paired hoist and sink og-3-4
- parallel
 - optimization og-1-2, og-4-1, ug-6-2, ug-6-5

- parallel (cont)
 - strip length og-B-4
 - strip-mining og-4-2
 - parallelization
 - defined og-B-4
 - description ug-6-2, ug-6-5
 - directives lrm-A-4
 - inhibitors ug-6-5
 - parallelizing
 - code outside of loops og-4-7
 - loops og-4-1
 - parameter passing to FORTRAN lrm-3-6
 - partial vectorization og-3-8
 - pasting operator
 - ## lrm-6-3
 - /**/ acc-A-3
 - pattern matching og-A-1
 - pcc command line acc-3-1
 - pclose function lrm-5-24
 - perror function lrm-5-24
 - pipelining og-2-2
 - pmd utility ug-4-2
 - pointer
 - aliasing og-7-1
 - data type lrm-2-8
 - generic lrm-2-8
 - pointers
 - aliasing og-6-4
 - converting function pointers acc-A-1
 - converting object pointers acc-A-1
 - to arrays lrm-2-12
 - popen function lrm-5-24
 - Portable C compiler acc-2-1
 - porting to backward-compatible mode acc-3-1
 - POSIX
 - conforming applications lrm-1-2
 - defined acc-2-2
 - definition lrm-1-1, lrm-5-1, ug-2-6
 - POSIX Extensions
 - limits.h lrm-5-9
 - setjmp.h lrm-5-15
 - signal.h lrm-5-16
 - stdio.h lrm-5-25
 - time.h lrm-5-33
 - POSIX functions
 - close acc-A-4
 - creat acc-A-4
 - fseek acc-A-4
 - open acc-A-4
 - read acc-A-4
 - unlink acc-A-4
 - write acc-A-4
 - post-mortem dump ug-4-2
 - potential aliasing og-7-1, og-7-2
 - pow function lrm-5-12
 - pow function
 - returns pow(|x|) on domain error acc-4-7
 - precision
 - float operations acc-A-3
 - floating-point operations acc-A-3
 - predefined macro names acc-A-2
 - predefined symbols ug-2-12
 - preference directives lrm-A-7
 - preprocessor
 - description of lrm-6-1
 - directives lrm-6-1
 - operators lrm-6-3
 - preprocessor directive
 - # define lrm-6-1
 - # elif lrm-6-4
 - # else lrm-6-4
 - # endif lrm-6-4
 - # error lrm-6-5
 - # if lrm-6-4
 - # ifdef lrm-6-4
 - # ifndef lrm-6-4
 - # line lrm-6-5
 - # pragma lrm-6-5
 - # undef lrm-6-3
 - printf function lrm-5-21
 - process og-B-4
 - processor status word lrm-3-2
 - processors, self-allocating og-1-2
 - profiler, CXpa ug-5-2
 - program constructs, efficient og-6-1
 - program counter lrm-3-2
 - program input and output lrm-4-3
 - program-unit optimization og-2-9
 - programming practices og-6-1
 - promotion
 - arrays og-8-3
 - function parameter acc-A-3
 - propagation, constant og-2-6, og-2-9
 - prototype, definition ug-A-16
 - pseudo-random sequence generation functions
 - lrm-5-26
 - pstrip directive lrm-A-8, og-9-9
 - ptrdiff_t type acc-4-4, lrm-5-18
 - putc function lrm-5-22
 - putchar function lrm-5-22
 - puts function lrm-5-22
- Q**
- qsort function lrm-5-27
- R**
- raise function lrm-5-15
 - rand function
 - return value lrm-5-26
 - with RAND_MAX lrm-5-25
 - RAND_MAX macro lrm-5-25
 - range errors, on arrays og-9-3
 - range of values, unqualified char acc-4-2
 - recv function lrm-5-14
 - Reader's Forum acc-viii, lrm-viii, ug-viii
 - realloc function
 - description lrm-5-27
 - request size zero acc-4-13
 - recurrence
 - apparent og-3-10

- recurrence (cont)
 - defined og-3-6, og-B-4
 - inhibitor of vectorization og-3-6
 - reduction of strength og-2-8
 - reductions og-3-11
 - redundant-assignment elimination og-2-8, og-2-12
 - redundant-use elimination og-2-8
 - reentrant
 - code og-4-5
 - defined og-B-4
 - use with `-re` ug-6-5
 - register allocation og-2-2
 - register storage class
 - casting acc-A-2
 - impact acc-4-4
 - legal uses ug-A-47
 - remainder, sign in integer division acc-4-3
 - remove function
 - description lrm-5-20
 - execution on open file acc-4-10
 - rename function
 - action on an existing file acc-4-10
 - description lrm-5-20
 - reordering expressions acc-A-2
 - report, optimization ug-2-14
 - reporting problems acc-viii, lrm-viii, ug-viii
 - representation
 - char lrm-2-2
 - character data lrm-2-12
 - double lrm-2-7
 - enum lrm-2-4
 - float lrm-2-6
 - int lrm-2-2
 - long lrm-2-2
 - long double lrm-2-7
 - long int lrm-2-2
 - long long lrm-2-4
 - long long int lrm-2-4
 - pointer lrm-2-8
 - short lrm-2-2
 - short int lrm-2-2
 - string lrm-2-12, lrm-2-13
 - return statement og-2-12
 - rewind function lrm-5-23
 - right shift of a negative integer acc-4-3
 - rindex function lrm-5-31
 - roundoff error og-9-4
 - runtime
 - functions, calling lrm-5-3
 - library lrm-5-1
 - messages ug-2-14
 - stack lrm-3-1
- S**
- scalar optimization og-2-1, ug-6-1, ug-6-2
 - scalar, optimization directive lrm-A-5, og-5-4, og-8-2, og-9-9
 - scanf function lrm-5-21
 - SCHAR_MAX macro lrm-5-8
 - SCHAR_MIN macro lrm-5-8
 - scope, external declarations acc-A-2
 - search functions lrm-5-30
 - searching and sorting functions lrm-5-27
 - SEEK_CUR macro lrm-5-19
 - SEEK_END macro lrm-5-19
 - SEEK_SET macro lrm-5-19
 - select, optimization directive lrm-A-8, og-8-2, og-9-9
 - setbuf function lrm-5-20
 - setjmp function
 - description lrm-5-14
 - optimization with ug-A-30
 - setjmp.h contents lrm-5-14
 - setlocale function lrm-5-11
 - setvbuf function
 - associated macros lrm-5-19
 - description lrm-5-20
 - shift operators og-2-8
 - short
 - alignment acc-4-4
 - data type lrm-2-1
 - short int data type lrm-2-1
 - short vector length og-5-4, og-9-9
 - SHRT_MAX macro lrm-5-8
 - SHRT_MIN macro lrm-5-8
 - sig_atomic_t type lrm-5-15
 - SIG_BLOCK macro lrm-5-16
 - SIG_CATCH acc-A-4
 - SIG_HOLD acc-A-4
 - SIG_SETMASK macro lrm-5-16
 - SIG_UNBLOCK macro lrm-5-16
 - sigaction function lrm-5-17
 - sigaction structure lrm-5-16
 - sigaddset function lrm-5-17
 - SIGCLD macro acc-A-4
 - sigcontext structure lrm-5-16
 - sigdelset function lrm-5-17
 - sigemptyset function lrm-5-17
 - sigfillset function lrm-5-17
 - SIGILL macro, reset of default handling acc-4-9
 - sigismember function lrm-5-17
 - sigjmp_buf type lrm-5-15
 - siglongjmp function lrm-5-15
 - sigmask macro lrm-5-16
 - signal function lrm-5-15
 - signal handler functions acc-A-4
 - signal.h
 - ANSI C changes acc-A-4
 - contents lrm-5-15
 - future changes acc-A-5
 - signals
 - blocking of acc-4-9
 - default actions acc-4-9
 - semantics acc-4-8
 - SIGILL, reset of default handling acc-4-9
 - what signals are available acc-4-8
 - SIGNALS_IN_PROG function acc-A-4
 - sigpending function lrm-5-17
 - sigprocmask function lrm-5-17

- sigprocmask, associated macros lrm-5-16
- sigsetjmp function
 - description lrm-5-15
 - optimization with ug-A-30
- sigsuspend function lrm-5-17
- sigvec structure lrm-5-16
- simple strength reduction og-A-1
- simplifying subscripts og-5-4
- sin function
 - description lrm-5-12
 - returns $\sin(0)$ on domain error acc-4-7
- single mode compiling lrm-1-2
- single-precision
 - IEEE format lrm-2-6
 - native format lrm-2-6
- sinh function
 - description lrm-5-12
 - returns `-HUGE_VAL` on domain error acc-4-7
- sinking
 - defined og-B-4
 - example og-4-3
 - hoisting and og-3-4
- size_t type lrm-5-18, lrm-5-20, lrm-5-26, lrm-5-29, lrm-5-32
- sizeof operator
 - bit field as argument ug-A-37
 - function as argument ug-A-37
 - void as argument ug-A-38
- sizeof_bitfield
 - compatibility mode actions ug-A-3
 - error message ug-A-37
- slower code og-9-9
- small trip counts og-5-4
- space characters
 - elimination text stream file acc-4-9
- span-dependent instructions og-2-3
- sprintf function lrm-5-21
- sqrt function
 - description lrm-5-12
 - returns $\sqrt{|x|}$ on domain error acc-4-7
- srand function lrm-5-26
- sscanf function lrm-5-21
- stack frame lrm-3-2
- stack pointer lrm-3-1
- static storage class
 - changing from extern to static ug-A-35
 - changing to static from extern ug-A-33, ug-A-34
- static variables og-4-5
- stdarg.h contents lrm-5-18
- stddef.h contents lrm-5-18
- stderr device lrm-5-19
- stdin device lrm-5-19
- stdio.h
 - ANSI C changes acc-A-4
 - contents lrm-5-19
 - future changes acc-A-5
- stdlib.h
 - contents lrm-5-25
 - future changes acc-A-5
- stdout device lrm-5-19
- stop values og-6-2, og-6-4, og-6-7, og-7-4, og-7-5, og-7-6, og-7-7
- storage class
 - function ug-A-18
 - function parameter ug-A-19
 - register acc-4-4, ug-A-47
 - struct or union ug-A-38
- strategy, optimization og-5-1
- strcat function lrm-5-29
- strchr function lrm-5-30
- strcmp function lrm-5-30
- strcoll function lrm-5-30
- strcpy function lrm-5-29
- strncpy function lrm-5-31
- strength reduction
 - and loops og-2-13
 - description og-2-8
- strerror function
 - description lrm-5-31
 - error message strings acc-4-14
- strftime function
 - description lrm-5-33
 - LC_TIME lrm-5-10
- strict compatibility mode
 - description acc-2-2, lrm-1-1
 - example lrm-1-2
- strict_syntax
 - compatibility mode actions ug-A-3
 - error message ug-A-26
- string concatenation operator (#) lrm-6-3
- string constant, definition ug-A-38
- string conversion functions lrm-5-26
- string handling functions lrm-5-29
- string literals, identical acc-A-3
- string.h
 - contents lrm-5-29
 - future changes acc-A-5
- stringizing operator (#) lrm-6-3
- strings.h, ANSI C changes acc-A-4
- strip distribution og-4-2
- strip length
 - parallel og-B-4
 - vector og-B-4
- strip-mining
 - by hand og-8-10
 - defined og-3-1, og-B-4
 - description og-3-2, og-4-1, og-5-4
 - directives lrm-A-7
 - optimization tip og-8-1
 - parallel lrm-A-7, og-4-2
- strlen function lrm-5-31
- strncat function lrm-5-29
- strncmp function lrm-5-30
- strncpy function lrm-5-29
- strpbrk function lrm-5-31
- strrchr function lrm-5-31
- strspn function lrm-5-31
- strstr function lrm-5-31
- strtod function lrm-5-26
- strtok function lrm-5-31

strtol function lrm-5-26
 strtoul function lrm-5-26
 struct
 alignment acc-4-5
 assignments ug-A-19
 data type lrm-2-10
 empty declaration acc-A-3
 empty declarations acc-A-2
 representation lrm-2-10
 structure
 alignment acc-4-5
 member alignment lrm-2-10
 padding lrm-2-11
 strxfrm function lrm-5-30
 subexpressions, elimination of common og-2-7,
 og-2-11
 subscripts, simplifying og-5-4
 switch case label, data type ug-A-40
 switch control expression, data type ug-A-40
 symbolic debugger ug-4-1
 synch_parallel directive lrm-A-9, og-4-4,
 og-4-5, og-4-6, og-9-9
 synchronization og-B-4
 system function
 description lrm-5-27
 valid argument acc-4-14
 system functions and ANSI functions lrm-4-2

T

table
 available signals and their semantics
 acc-4-8
 character constant representation acc-4-2
 characters checked by ctype.h functions
 acc-4-6
 compatibility modes lrm-1-1, lrm-5-1
 compiler options for profiling ug-5-2
 default actions for signals acc-4-9
 diagnostic condition default settings ug-A-3
 errno values of fgetpos and ftell acc-4-10
 errno values of fgetpos, fsetpos, and ftell
 lrm-5-23
 error messages acc-4-11
 floating-point bit length lrm-2-5
 FORTRAN and C declarations lrm-3-6
 index to diagnostic messages ug-A-3
 integer conversion acc-4-3
 integral ranges lrm-2-2
 integral type bit length lrm-2-1
 ipc/network argument error messages
 acc-4-12
 long float range: native and IEEE
 lrm-2-8
 long long data type range lrm-2-3
 math error messages acc-4-11
 math function return values acc-4-7,
 lrm-5-13
 maximum strip-mine lengths lrm-A-7
 native and IEEE floating-point ranges
 lrm-2-5

table (cont)
 nfs error messages acc-4-12
 nonblocking and interrupt IO error mes-
 sages acc-4-12
 optimization directives ug-6-9
 optimization options ug-6-7
 restrictions on optimization directive use
 ug-6-10
 restrictions on optimization directives
 lrm-A-1
 SystemV record locking error messages
 acc-4-12
 tape system error messages acc-4-13
 trigraph representations acc-A-2
 type of post-mortem dump contents ug-4-2
 TAC (Technical Assistance Center) ug-B-1,
 acc-viii, acc-C-1, lrm-viii, lrm-B-1,
 og-C-1, ug-viii
 tan function lrm-5-12
 tanh function lrm-5-13
 tasking directives
 lrm-A-4
 example ug-A-7
 listed lrm-A-4
 maximum number ug-A-17
 use og-4-7
 technical assistance
 obtaining acc-viii, lrm-viii, ug-viii
 technical assistance center
 telephone number for acc-viii, lrm-viii,
 ug-viii
 technical assistance center (TAC) ug-B-1,
 acc-viii, acc-C-1, lrm-viii, lrm-B-1,
 og-C-1, ug-viii
 text stream
 terminating new-line character acc-4-9
 truncation acc-4-10
 The Basics og-1-1
 thread
 defined og-B-4
 description og-4-1
 thread-private og-B-4
 thread-specific og-B-4
 tilde-escape sequences
 restrictions on use ug-B-6, acc-C-6,
 lrm-B-6, og-C-6
 use in contact ug-B-4, acc-C-4, lrm-B-4,
 og-C-4
 time
 conversion functions lrm-5-33
 functions lrm-5-32
 manipulation functions lrm-5-33
 time function lrm-5-33
 time_t type lrm-5-32
 time.h contents lrm-5-32
 tm structure lrm-5-32
 TMP_MAX macro lrm-5-19
 tmpfile function lrm-5-20
 tmpnam function
 description lrm-5-20
 L_tmpnam lrm-5-19

tmpnam function (cont)
 TMP_MAX lrm-5-19
 toascii
 function acc-A-4
 macro lrm-5-5
 tolower function lrm-5-4
 toupper function lrm-5-5
 tree-height reduction og-2-3
 trees, balancing og-2-3
 tricks and tips og-8-1
 trigonometric simplification og-2-5
 trigraphs acc-A-2
 trip count og-3-2, og-5-4, og-8-1, og-8-2, og-9-9
 trouble reports ug-B-1, acc-viii, acc-C-1,
 lrm-viii, lrm-B-1, og-C-1, ug-viii
 truncation of text streams acc-4-10
 type conversion from FORTRAN lrm-3-5
 typedef names in formal parameters acc-A-2
 tzname variable lrm-5-33

U

UCHAR_MAX macro lrm-5-8
 UINT_MAX macro lrm-5-8
 unary minus
 operands ug-A-28
 unary plus
 operands ug-A-28
 ungetc function lrm-5-22
 union
 alignment acc-4-5
 assignments ug-A-19
 data type lrm-2-9
 empty declarations acc-A-2
 order of accessing members acc-4-4
 unions og-7-1
 unix macro ug-2-12
 UNIX-to-UNIX
 Communication Protocol ug-B-1, acc-C-1,
 lrm-B-1, og-C-1
 copy command, uucp ug-B-1, acc-C-1,
 lrm-B-1, og-C-1
 unqualified char
 range of values acc-4-2
 unroll directive lrm-A-9, og-8-3
 unrolling loops og-6-2, og-8-2
 unsafe optimizations og-A-1
 unsigned_suffix
 compatibility mode actions ug-A-3
 error message ug-A-22
 user-directed optimization ug-6-7
 USHRT_MAX macro lrm-5-8
 UUCP
 connection to TAC ug-B-1, acc-C-1,
 lrm-B-1, og-C-1
 uucp
 UNIX-to-UNIX copy command ug-B-1,
 acc-C-1, lrm-B-1, og-C-1

V

va_arg macro lrm-5-18
 va_end macro lrm-5-18
 va_list type lrm-5-18
 va_start macro lrm-5-18
 variables
 global og-2-12, og-4-5
 static og-4-5
 vector
 chaining, defined og-B-4
 optimization og-1-2, og-3-1, ug-6-1, ug-6-4
 registers og-4-3
 spill og-B-4
 strip length og-B-4
 transformations og-3-2
 vectorization
 defined og-3-1
 description ug-6-1, ug-6-4
 directives lrm-A-4
 inhibitors ug-6-4
 partial og-3-8
 vers command
 using to find program version number
 ug-B-2, acc-C-2, lrm-B-2, og-C-2
 vfprintf function lrm-5-21
 void data type
 description lrm-2-8
 typical uses ug-A-47
 volatile
 data type acc-1-4
 what is an access acc-4-5
 vprintf function lrm-5-21
 vsprintf function lrm-5-21
 vstrip directive lrm-A-8, og-9-9

W

wall-clock time og-4-1
 wchar_t type lrm-5-18, lrm-5-26
 wctombs function lrm-5-28
 wctomb function lrm-5-28
 whence command
 using to find program path name ug-B-2,
 acc-C-2, lrm-B-2, og-C-2
 which command
 using to find program path name ug-B-2,
 acc-C-2, lrm-B-2, og-C-2
 while loops og-6-7
 wide character constant
 definition of ug-A-22
 wrong answers og-9-1

Y

y0 function lrm-5-14
 y1 function lrm-5-14
 yn function lrm-5-14

Z

zero-length arrays acc-A-2
zero-length files acc-4-10