Software Product Description

PRODUCT NAME: VAX C, Version 3.2 **SPD 25.38.17**

DESCRIPTION

VAX C is an extended implementation of the C programming language originally developed at Bell™ Laboratories. The VAX C compiler runs under the VMS Operating System and generates optimized and position-independent code.

As a native-mode VAX language, VAX C is integrated into the VAX common language environment. All VAX system services are thus available to programs written in VAX C. VAX C supports VAX Record Management Services (RMS), including sequential, relative, and indexed file organizations and associated access methods, in addition to the stream file-access conventional among most C implementations. VAX C programs can invoke, as functions, modules written in other VAX languages.

VAX C provides assistance in decomposing loops. Decomposed loops run in parallel in multiple processes, reducing the total elapsed time required to run the program. This capability is most useful on multiple processor machines, such as the VAX 8800.

VAX C supports the lint-like features of the Source Code Analyzer component of VAX Language-Sensitive Editor/Source Code Analyzer (SPD 26.59.xx). The combination of using function prototypes and the Source Code Analyzer component allows the programmer to check for consistent function usage throughout a program environment. VAX C also generates complete debug and traceback records for use with the VAX Symbolic Debugger. The Debugger allows the C programmer to set breakpoints, examine and modify the contents of user variables, and selectively halt or continue program execution.

Features

- Compiler-assisted decomposition of for loops to run in parallel in multiple processes
- Function prototypes for declaring and checking function argument count and types
- Structured programming control flow constructs:
  - — if...else construct for simple selection
  - — switch statement for multi-choice selection with an arbitrary number of case statements
  - — while, do, and for statements for iterative execution
- Flow modification statements:
  - — break
  - — continue
  - — goto
- Data types for numeric, nonnumeric, and systems programming:
  - — Byte, word, and longword signed and unsigned integers
  - — Integer constants in decimal, octal, and hexadecimal radices
  - — Void data type
  - — Single-character variables and constants
  - — Single, double-precision and G_Floating point numbers
  - — Pointer variables containing the addresses of other variables
  - — Data aggregates including arrays, structures, and unions, referencing allowed
- Storage allocation using:
  - — Auto, static, register, and extern storage allocation classes for variables
  - — Keywords (globalref, globaldef, and globalvalue) for sharing data among program modules
  - — Keywords (readonly, noshare, and psect name specification) for control of data attributes and data placement
  - — Keywords (const and volatile) for specifying attributes for storage in which objects are placed
  - — Keyword (_align) for specifying the alignment boundaries of data objects

- Large set of concise arithmetic, relational, and logical operators
- Preprocessor control statements for:
  - — File inclusion
  - — Identifier substitution
  - — Conditional compilation
  - — Pragmas to control compiler options
  - — Pragmas to support compiler-assisted decomposition
- Separate compilation capabilities
- Compiler generated listing file including optional:
  - — In-line machine code
  - — Expanded preprocessor substitution listing
  - — Storage allocation map
  - — Cross referencing of variables
  - — Compilation statistics
  - — Loop decomposition report
- Access to VAX machine instructions from C source code
- Integration into the VAX Common Languages Environment:
  - — Generation of complete debug and traceback records for VAX Symbolic Debugger support
  - — Facility for interfacing with the VAX calling standard
  - — Access to the VAX Common Run-Time Library for general purpose routines and support of multi-language environments
  - — Access to the data management facilities of VAX Record Management Services (RMS) via direct calls to the Common Run-Time Library
  - — Access to data structures declared using the VAX CDD/Plus
  - — Interface to VAX DEC/Shell as an alternate command line interpreter under which the VAX C compiler and user programs may be run
  - — Support for VAX Language-Sensitive Editor/Source Code Analyzer to provide error diagnostics to the Language-Sensitive Editor component, cross reference information for the Source Code Analyzer component, and support for low-level program design, including the processing of pseudocode and the extraction of design information from comments
- Extensive global and local optimizations of generated code for increased performance and reduced size under VMS
- Interface to the curses screen manipulation package

Compatibility with Other C Implementations

VAX C represents a more current definition and implementation of the language than is described in the initial guiding document for C, *The C Programming Language*[1]. However, some incompatibilities among implementations do exist. In general, many programs written in C for other compilers can be successfully recompiled under VAX C.

The C language has been closely associated with the UNIX® Operating System, which itself is largely written in C. Over 150 UNIX-specific routines have been included in the run-time library available with the VAX C compiler. Routines selected for inclusion in this library are primarily those for which a reasonable emulation is possible under VMS. Inherent architectural differences between UNIX and VMS necessarily preclude the likelihood of every C program written for a UNIX environment compiling and executing unaltered. Programs exploiting highly machine-dependent or UNIX-specific features will require some conversion effort.

Run-Time Package for C Applications

The VAX C Run-Time Package is distributed with the VMS Operating System. The VAX C Run-Time Package provides shareable routines to perform input/output, character and string handling, mathematical computations, memory allocations, and emulation of selected UNIX features. Also supported are the 4.3BSD socket routines for interprocess communication across a network.

HARDWARE REQUIREMENTS

VAX, MicroVAX, VAXstation, or VAXserver configuration as specified in the System Support Addendum (SSA 25.38.17-x)

SOFTWARE REQUIREMENTS

For Systems Using Terminals (No DECwindows Interface):

VMS Operating System

For Workstations Running VWS:

VMS Operating System
VMS Workstation Software

[1] Brian Kernighan and Dennis M. Ritchie, Prentice-Hall, 1978.

For Workstations Running DECwindows:

VMS Operating System (and necessary components of VMS DECwindows)

VAX C supports a DECwindows compiler interface for compile qualifier selection on workstations. This interface is accessible from FileView.

Refer to the System Support Addendum (SSA 25.38.17-x) for availability and required versions of prerequisite software and for information regarding components of VMS DECwindows.

ORDERING INFORMATION

Software Licenses: QL-015A*-**
Software Media: QA-015A*-**
Software Documentation: QA-015AA-GZ
Software Product Services: QT-015A*-**

* Denotes variant fields. For additional information on available licenses, services, and media, refer to the appropriate price book.

SOFTWARE LICENSING

This software is furnished under the licensing provisions of Digital Equipment Corporation's Standard Terms and Conditions. For more information about Digital's licensing terms and policies, contact your local Digital office.

LICENSE MANAGEMENT FACILITY SUPPORT

This layered product supports the VMS License Management Facility.

License units for this product are allocated on a CPU-capacity basis.

For more information on the License Management Facility, refer to the VMS Operating System Software Product Description (SPD 25.01.xx) or the *License Management Facility* manual of the VMS Operating System documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available from Digital. For more information, contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.

® UNIX is a registered trademark of UNIX System Laboratories, Inc.

™ Bell is a trademark of Bell Telephone Companies.

™ X11 is a trademark of Massachusetts Institute of Technology.

™ The DIGITAL Logo, CI, DECwindows, MicroVAX, ULTRIX, VAX, VAX C, VAX CDD/Plus, VAX RMS, VAXcluster, VAXft, VAXserver, VAXstation, and VMS are trademarks of Digital Equipment Corporation.