Programming Environments

John Foderaro

We continue our survey of Common Lisps with a report on the Lucid Common Lisp Development Environment. Our previous surveys were for Lisps on dedicated Lisp Machines and for Franz Inc.'s Extended Common Lisp.

Background:

Company: Lucid, Inc.
Product Name: Lucid Common Lisp (also called DOMAIN/CommonLISP, Common Lisp for the IBM RT PC, PRIME Common LISP, and Sun Common Lisp)
Version of Product Described: Lisp 2.1
Hardware Available On: Apollo, Sun, VAX (VMS, ULTRIX, BSD 4.2 & 4.3), IBM RT PC, NCR, Prime, Intel 386-based machines Available in early '88 on IBM 370 (MVS) and HP9000 Series 300
This Version Available When: Currently on Apollo and through early Fall on Sun and VAX (VMS, ULTRIX, BSD 4.2 & 4.3)

Lucid Common Lisp is a full implementation of Common Lisp that runs on a range of general-purpose computers under a variety of operating systems, including VMS, MVS, UNIX 4.2, and UNIX System V.

The Lucid Common Lisp Development Environment includes, in addition to a Lisp interpreter and a compiler, the Lisp Editor, the Window Tool Kit, the Flavor System, and a complete debugging environment. The Editor is an EMACS-style graphics-based editor that combines the ease of working with EMACS-like commands with commands specifically tailored to Lisp. On a graphics terminal, the Editor windows may be moved and reshaped at the user's preference. On a non-bitmapped terminal, the user works in a screen-based editor that has all of the EMACS-like functionality of the graphics-based editor. The Editor has a programmable interface for applications which may wish to incorporate it as a component.

Primarily Residential or File Based: file based.

Components of the Programming Environment: The Lucid Common Lisp Development Environment consists of the Lucid Common Lisp Editor, the Window Tool Kit, a complete debugging environment, a foreign function interface to C, Pascal, and FORTRAN, and the Flavor System.

The Editor is integrated into the Common Lisp environment and runs in the same address space as Lisp. Specific editor commands that work with Lisp include commands for compiling a region of code, compiling an editor buffer, evaluating an expression, and displaying the arguments to the current function. Users can customize the Editor by developing their own key bindings and definitions. In addition, a user can incorporate the Editor into an application program. Some of the hardware vendors that support Lucid Common Lisp provide an interface from Lisp into a standard EMACS that is available on their product.

The Lucid Common Lisp Editor is window-based. However, Lucid Common Lisp can be used, from within the Editor, on either a system console or from an ASCII terminal. On an ASCII terminal, the Editor is a screen-based editor with fixed window positions; all of the Lisp-based EMACS commands and facilities are, however, still available.

The Window Tool Kit is a collection of building blocks with which the user can build a graphics display through Lisp function calls. The Window Tool Kit allows the user to customize windows facilities for use with specialized applications. It provides data structures such as viewports and windows, which specify screen areas for the display of character and graphic output; it also provides operations for defining fonts, making pop-up menus, and accessing the mouse. Additionally, the user can define keyboard interrupt characters that invoke a user-specified function.

The Window Tool Kit is designed to allow the building of graphic system interfaces that are portable across all operating systems and hardware supported by Lucid Common Lisp.

The Debugging Environment consists of several program debugging tools. The Debugger allows interactive examination and modification of stack frames (activation records). While within the context of the Debugger, the user still has access to the same interpretive facilities that are available at the top level of Lisp. The debugging tools also include the Trace facility, the Stepper, and the Inspector.

The Foreign Function Interface provides the user with the ability to make calls to the routines written in languages other than Lisp (C, Pascal, FORTRAN). With this interface, the user can also create Lisp streams and extract file descriptors associated with Lisp streams so that I/O can be performed with files.

The Flavor System is an implementation of MIT Flavors.

Lucid Common Lisp is available with two documentation manuals. Each gives extensive explanations and examples of the code being described.

The Reference Manual contains a full description of Common Lisp, with many examples of each function.

The User's Guide describes all of the language extensions and utilities provided by Lucid. An important section of the User's Guide is the section on "Compiling Lisp Programs" that contains in-depth advice on how a user can best tune a program for optimum performance and memory use.