Chez Scheme Version 5

from Cadence Research Systems is a complete high-performance implementation of IEEE/ANSI standard Scheme, with a fast incremental optimizing compiler, an efficient run-time system, an extensive run-time library, and a productivity-enhancing program development environment.

FEATURES

Uniformly high performance. The new *Chez Scheme Version 5* compiler achieves a level of performance previously unheard of in Scheme and Lisp systems. Many Scheme and Lisp compilers are good at certain optimizations, support only “common” programming styles, or generate reasonable code only with type and bounds checking disabled. *Chez Scheme* has been written to provide outstanding performance across the board. And with its advanced program analysis, code transformation, register allocation, and code generation strategies, there is no longer a significant “performance gap” between Scheme and low-level languages such as C or C++.

Sophisticated storage management. *Chez Scheme's* generation-based garbage collector segregates objects according to age and access patterns and considers older objects for collection less frequently than newer objects. This results in reduced garbage collection overhead and more efficient use of your computer's memory. Furthermore, garbage collection overhead does not increase significantly as the size of your problem increases, as with most other garbage collection strategies. With automatic storage management, programs are free of the dangerous storage leaks and dangling pointers common in low-level languages. Programs making heavy use of dynamic allocation are usually much faster in *Chez Scheme* as well, due to fast in-line object allocation and low-overhead garbage collection.

Efficient use of resources. *Chez Scheme's* efficient use of the most critical resources in your computer: memory, disk, and cpu, may literally double or triple the effectiveness of your workstation or timeshared computer. This can save money that would otherwise have to be spent on expensive hardware upgrades.

No limitations. You should never have to change your programming style simply to avoid an arbitrary system limitation. *Chez Scheme* places no limits on the size of heap, stack, program, or data structures. No limits of any kind are placed on source expressions such as the length of identifier names, the size of parameter lists, or the nesting depth of code. Furthermore, our compiler, runtime libraries, and storage manager have been carefully coded to avoid abnormal performance penalties for the use of large data structures. *Chez Scheme* dynamically resizes the heap, stack, and various internal data structures to fit the size of your program and data. This means that small programs use only what they need, while available virtual memory is all that limits the size of your application.

Fast incremental compilation. In a competitive marketplace, the time it takes to develop an application can be as important as how fast the application itself runs. And a responsive development system can result in the development of superior application code. Because of this, we have focused on the speed of *Chez Scheme's* compiler with as much energy as on the speed of generated code.

Debugging compiled code. A fast development system is not worth much without good debugging tools. *Chez Scheme* provides a sophisticated trace facility along with an interactive inspector that helps the programmer quickly pinpoint the source of bugs, significantly reducing program development time. The inspector is used to inspect the current continuation (control stack), procedures, and other objects whenever an error, interrupt, or explicit break occurs. Source code and variable names

are displayed along with the current values of both free and bound variables. The inspector works for code compiled at any level of optimization, so there is no need to sacrifice development time or code speed for debugability.

Foreign procedure interface. *Chez Scheme* provides an advanced foreign procedure interface that protects your investment in C code and allows the use of C libraries. The interface allows C object code to be linked into the system and invoked directly from your Scheme program. Automatic datatype conversions are performed for arguments and return values, so there is no need to be conscious of representation differences on either end. Code written in other languages can be hooked in via C interface routines.

Complete high-quality numeric code and libraries. *Chez Scheme* supports the entire hierarchy of exact and inexact numbers described by the IEEE/ANSI Standard: integer, rational, real, and complex. We have gone to great lengths to insure accuracy of all of our numeric operations, with special attention to avoiding unnecessary loss of precision, underflow, or overflow in floating point computations. Type-specific numeric operators for fixed-precision integer, floating point, and complex operations are provided for greater efficiency. Floating point numbers are read and written with full input/output consistency, using the fewest number of digits necessary, as required by the standard.

Advanced I/O. *Chez Scheme* provides a variety of advanced operations on ports, including character-level, block-level, and expression-level input and output. Complete facilities for creating, opening, closing, and positioning files as well as operations on string-based ports are provided. The sophisticated pretty-printer may be tailored through several control parameters. Generic ports permit the construction of ports with arbitrary read/write semantics.

Reliability. Reliability is one of our top priorities, so you can trust *Chez Scheme* to run your programs correctly. We have an extensive test suite that includes complete application programs as well as thousands of lines of tests for specific features. And each major distribution is extensively field tested before general distribution. Furthermore, we have a "drop everything" policy of tracking down bugs and eliminating them as soon as they are found. But reliability doesn't result from testing alone: reliability must be built into a system from the ground up. Reliability is the first consideration whenever any enhancement is made, as it has been since the first version of *Chez Scheme* was developed in 1984.

PLATFORMS SUPPORTED

- Sun Sparc SunOS 4.X & Solaris 2.X
- DEC Alpha AXP OSF/1 V2.X
- Silicon Graphics IRIX 5.X
- Motorola MC88000 SVR3 & SVR4
- Intel 80x86 NeXT Mach 3.2
- Decstation/Decsystem Ultrix 4.3A
- Intel 80x86 BSDI BSD/386 1.1
- Intel 80x86 Linux

FOR MORE INFORMATION CONTACT

Cadence Research Systems
3814 Devonshire South
Bloomington, IN 47408
USA

Phone: (812) 333-9269
Fax: (812) 332-4688
Email: sales@cadence.bloomington.in.us

©1994 Cadence Research Systems. *Chez Scheme* is a trademark of Cadence Research Systems.