

# LISA'S ALTERNATIVE OPERATING SYSTEM

The rigid command language syntax and mode based structures of conventional operating systems may be a thing of the past.

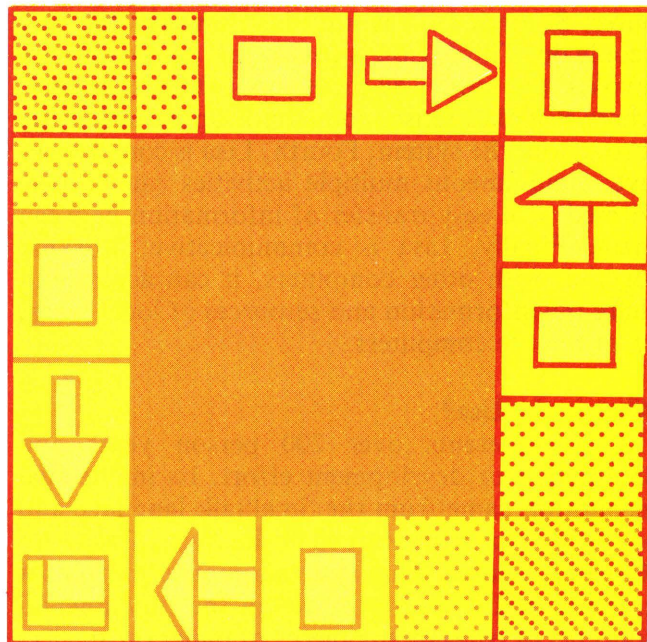
by Bruce Daniels

Lisa, an advanced personal computer system, radically changes the way people interact with a computer. In traditional microcomputer operating systems, interaction occurs strictly via a special command language. With Lisa, however, the use of a very intuitive and consistent electronic desktop model allows interaction through pointing and graphics.

Although traditional operating systems provide a raw capability for general purpose computing, they are excessively complex, arcane, and difficult to use. Thus, they do not meet the needs of nonexpert computer users. Nor do these systems provide all the required functionality since they do not support rapid switching between simultaneous activities. In addition, the convenient examination and transfer of data between these interdependent activities are not supported. Furthermore, existing systems cannot be easily modified.

The Apple's Local Integrated Software Architecture (Lisa) takes a revolutionary approach to

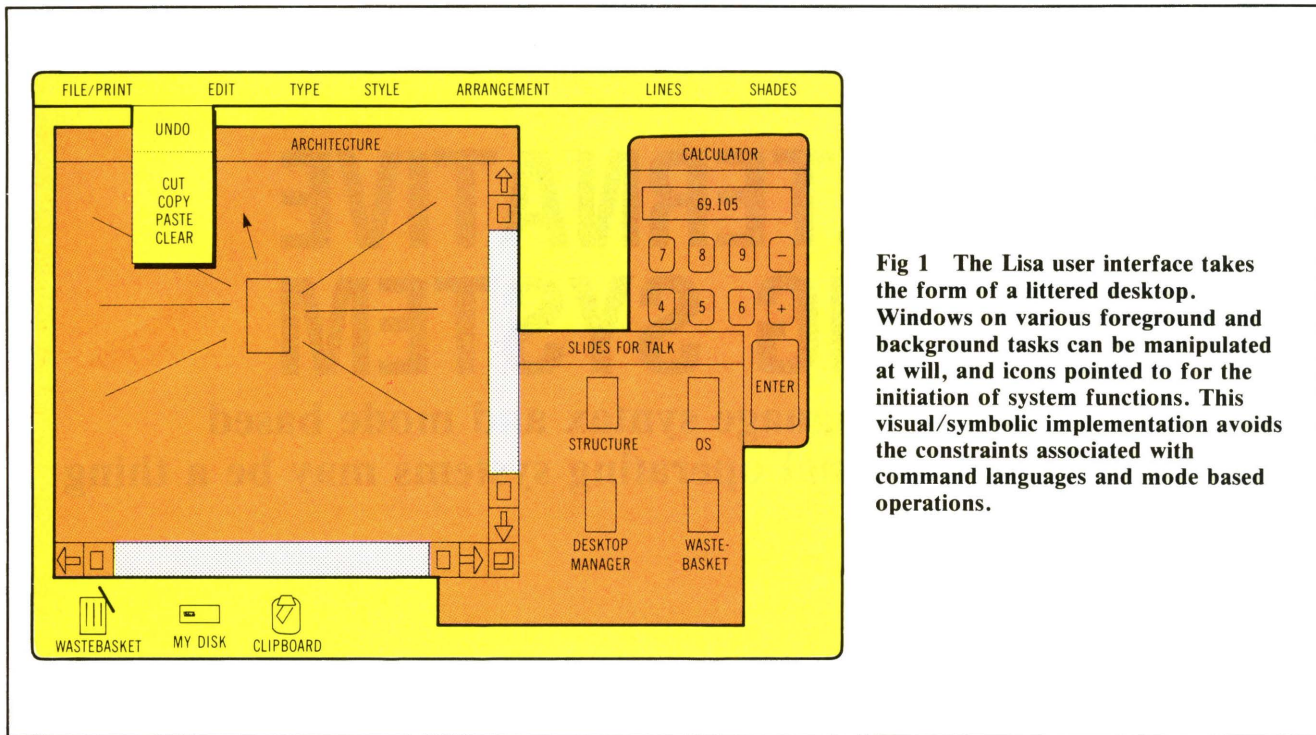
*Bruce Daniels is a consulting engineer at Apple Computer, 20525 Mariani Ave, MS 2P, Cupertino, CA 95014, where he is responsible for software architecture, integration, performance, and reliability for the Lisa Div. He has BS and MS degrees in computer science from the Massachusetts Institute of Technology.*



operating system software, resulting in an integrated system that is an order of magnitude easier to use. This new approach, not constrained by the traditional structures of existing operating systems, is concerned almost exclusively with providing features that make the computer easier or more convenient for the nonexpert user. With today's technology, the difficulty of providing such a feature is minimal.

Five concepts are implicit in Lisa's operating system. First, Lisa is a personal computer. As





**Fig 1** The Lisa user interface takes the form of a littered desktop. Windows on various foreground and background tasks can be manipulated at will, and icons pointed to for the initiation of system functions. This visual/symbolic implementation avoids the constraints associated with command languages and mode based operations.

demonstrated by the microcomputer revolution, it is economically and technically sound to dedicate a single computer to a single person, rather than forcing many users to share part of a remote computer. Second, Lisa is intuitive; it works the way people do. Third, Lisa is consistent; commands are invoked the same way throughout the system and, where it makes sense, commands apply uniformly throughout the system. Fourth, Lisa is comprehensive; simultaneous multiple activities can be performed with easy transfer of information between them. Finally, Lisa is communicative; working together with other computers, it can access and share the information and resources of local Lisas, or of remote computers.

### What is the Lisa?

Lisa represents over 200 person years and \$50 million of development effort. Its intellectual inspiration came from the Smalltalk language and operating system developed at the Xerox Palo Alto Research Center. This system's pioneering use of graphics and a mouse to present the fundamental concepts of overlapping pieces of paper on an electronic desktop marked a departure in computer operating software. Taking the Smalltalk concepts, Apple spent three years refining them, adding innovations, and reducing costs.

Hardware is based on the MC68000 microprocessor. This chip's large, uniform address space, 32-bit wide internal data paths, and powerful regular instruction set give it a clear advantage over many of today's microprocessors. Since Lisa was intended to be the first member of a powerful,

long-lived family of computer systems, this microprocessor's capability to gracefully evolve into faster and more powerful versions through the 1980s was also important.

As standard equipment, Lisa has a 1M-byte main memory—fully parity checked to ensure reliable operation. Designed to support up to 2M-byte main memory, this system also has special memory management hardware to support multiprocessing. Lisa uses a 5M-byte Winchester disk and two built-in floppy disk drives. These floppies, storing over 860K bytes of information, are specially designed and developed to provide large capacity, reliability, and foolproof operation. An IBM-like layout and numeric keypad are included on Lisa's detachable keyboard.

The most interesting feature of the hardware is the mouse—a comfortable, convenient pointing device. Lisa's mouse contains a single button in order to eliminate user confusion. Also included is a built-in 12" black and white bit-mapped display. This high resolution device allows users to simultaneously display graphics symbols, multiple fonts, and type styles on the screen.

The user interface is modeled around a desktop (see Fig 1) and contains graphical images of familiar office objects such as documents, folders, and stationery. Users control the machine by pointing at these images with the mouse. Graphics and multiple type styles, up to one-third of an inch tall, appear on the screen. These are reproduced with unparalleled quality on the printed page. A companion dot matrix printer produces high resolution text and graphics.



Standard with the Lisa system are six integrated, consistent software applications: a word processor; an electronic spreadsheet; a program that turns numbers into meaningful business charts and graphs; a graphics editor to create diagrams, drawings, and illustrations; a project scheduling program; and a database program. Also available is a seventh application—an asynchronous communication package emulating TTY, VT52, and VT100 terminals. IBM hosts can be accessed through a 3270 communications package and 3270 cluster controller.

### Existing solutions are inadequate

Although a wide variety of microcomputers are currently available, many do not satisfy the needs of the typical information user. A major deficiency in current microcomputers is the lack of suitable software.

Software consists of the operating system and applications that it supports. Currently, such popular microcomputer operating systems as CP/M, MS-DOS, Apple DOS, UCSD p-System, and Unix are fundamentally alike. Certainly, they differ in the way to use the system, but the basic manner in which users interact with the system is qualitatively similar. In reality, the computer industry is developing the same operating system over and over.

Thus, it is not surprising that traditional operating systems do not meet the needs of modern users since computer experts (ie, programmers) designed them all for other computer experts' use. Such expert users are comfortable with the artificial concepts and terms upon which these operating systems are built. They know, for instance, what a program is and what it means to execute such a program. They understand the concept of a file, a directory or catalog, and a disk volume. Most important, such experts are obviously accustomed and willing to spend many hours studying the cryptic manuals required to master such a technical subject. Soon, people like this will represent a minority of microcomputer users. The Lisa user interface offers a welcome alternative.

### The Lisa user interface

Consider the user interface of an operating system. This is the executive portion that the users see when they first turn on the machine. Through this interface a user controls the system, runs the programs, and manipulates data files. On a traditional operating system, the system executive is a command interpreter. The command interpreter interacts with the user through a predefined command language similar to conventional programming languages. It is a rigid, somewhat cryptic, and artificial syntax for the commands that the interpreter will legally accept. For example, examine the com-

mands used in several popular operating systems to delete the file containing a *Computer Design* article:

CP/M	ERA B:CmprDsgn.Tex
Apple DOS	DELETE Computer Design Article,S6,D2
MS-Dos	ERASE B:CmprDsgn.Tex
Unix	RM/USR/Bruce/Publications/ Computer Design
UCSD p-System	F R Pubs:CmptrDesgn.Text

Compare these traditional operating system commands with the Lisa user interface. Here, a document icon is simply dropped into the Wastebasket Icon. While both methods perform essentially the same function, the command syntaxes are certainly not intuitive or easy to remember. If the actual command is ERA, it then does no good to try KILL, DELETE, REMOVE, or even ERASE. Also, commands will not work if B-, B-, or B/ instead of the B= are used in the file name. And, of course, every data object in the system must have a unique file name to be used when it is accessed. Because file names must be abbreviated to fit within the maximum file name length allowed by each system, they tend to be very cryptic. Lisa allows the name of a document to be almost any length or form. Names need not even be unique.

Why use a command language to communicate with traditional operating systems? Because programmers have a lot of experience with such languages and get plenty of practice in using them. Knowing how to precisely specify syntax and knowing certain programming techniques allow them to implement languages quickly and efficiently. Therefore, the choice of a command language interface for operating systems is largely for the convenience of the designer and implementer of that system. Such a command language interface is certainly not selected for the convenience of a user who is not a computer expert.

An unfortunate side effect of a command language interface is the heavy use of modes that restrict system functions. For example, most text editors have an insert mode that allows typed text to be inserted into the file. Modes are objectionable for many reasons. The biggest reason, however, is the myriad details that users must remember. These include how to get into and out of a mode, what commands are and are not available, and most importantly, what mode they are currently in.

It is probably impossible to design a serious system that has no modes whatsoever. However, too many modes gives a user a feeling of constantly



navigating a command maze. This may be fun in a game like Adventure, but it is frustrating and tiresome in the everyday use of an operating system.

Heavy use of modes also produces an inconsistent user interface. Commands and capabilities in one mode are not usually available in another. In most systems, for example, the commands for editing text entered into a text editor are not available for editing the text entered to a file name request. Worse still, the same command may not even have the same meaning in every mode. For example, in one system, the command R is used to Run a program, Remove a file, Replace selected text, or even to insert the letter R into a text file. Conversely, two completely different commands may actually invoke similar or even identical functions; for example, lines are *deleted*, but files are *erased*.

Compounding this problem of inconsistency in the user interface are the application programs used with an operating system. Because application programs are designed and sold separately from the system itself, more differences exist between such programs and the operating system than between various parts of the system. This is particularly true of portable application programs. Clearly, such programs, designed to operate with a number of different operating systems, will not be particularly consistent with any of them. It is practically impossible to expect one conventional application to provide commands that invoke functions similar with another.

### **Operating system calls**

Although it is Lisa's user interface that one first notices, the fundamental system calls of the operating system are what make the software revolutionary. Lisa's operating system, designed specifically as a single-user system, does not need the excess capabilities built into multi-user, timesharing operating systems.

The operating system kernel provides multitasking support to the multiple, simultaneous windows displayed on the Lisa screen and the switching among them. In fact, an independent Lisa task is associated with each window. Lisa's multitasking is also used to allow such concurrent operations as foreground user interaction and background printing. Sophisticated memory management allows these multiple tasks to efficiently use and share system memory.

Specially designed to provide reliable storage of information, the Lisa File System redundantly distributes and stores critical file system data such as file directories. In the event of damage to the storage media, an operating system Scavenger can

repair damaged data and recover the user's information. The file system provides device independent access to information through a byte-stream input/output interface.

---

### *Management capabilities of the graphics screen are an integral part of the operating system design.*

---

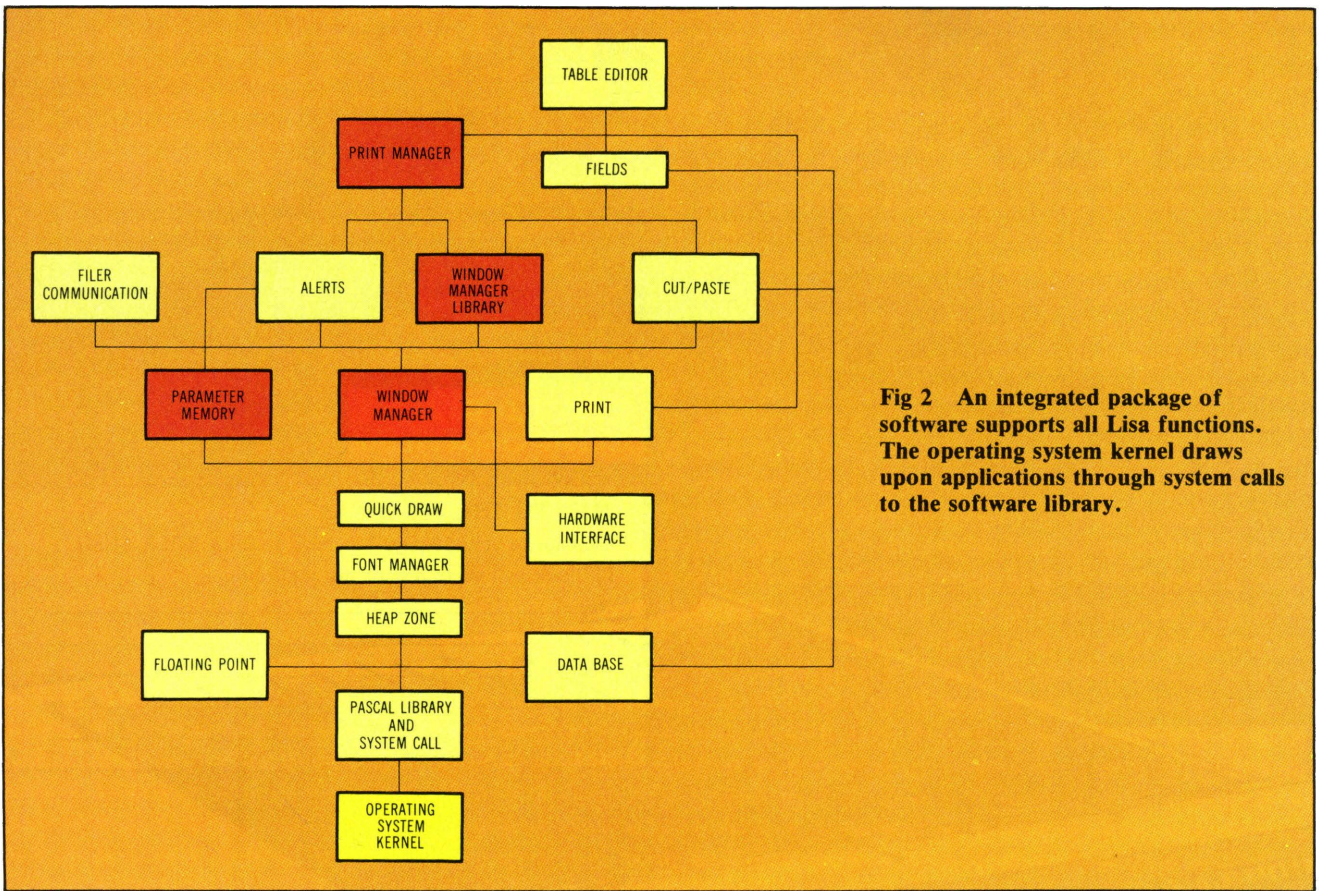
Lisa's operating system has additional capabilities that distinguish it from traditional systems. Graphics and the desktop model are not just simply an appendage to the operating system. Management capabilities of the graphics screen are an integral part of the operating system design. The Window Manager is responsible for keeping track of the number of open windows, the location of each on the screen, the size of each, and which windows are in front of or behind the other windows. For each window covered by other windows, and therefore partially obscured, this Window Manager calculates that portion currently visible. QuickDraw, a high speed, bit-map graphics package, automatically restricts or clips any output to just that portion of the visible display window. When windows are moved, resized, or otherwise changed, Window Manager redisplayes uncovered portions of windows. As a result, application designers need not worry about where on the screen windows are located, or how much is currently visible.

Similarly, mouse and keyboard support are an integral part of the operating system. Hardware interface software responds to interrupts from such input devices as the mouse or keyboard and queues input event information so that it is not lost if the application is busy. The Event Manager classifies these events and routes them to the proper window.

One of the most important aspects of Lisa is its printing technology. Superb drawings, calculations, or reports on a computer are insufficient if one cannot then print this information for others. Lisa supports both dot-matrix and daisy wheel printers. The guiding philosophy behind Lisa's advanced printing technology is, what you see on the screen is exactly what you will get on the printed page. This even extends to printing good quality graphics, as well as the usual text, on a daisy wheel printer.

Closely associated with the Lisa Operating System is a library of software modules (Fig 2). These modules perform various functions frequently needed by the Lisa application programs. The use of library modules saves an application designer the time it would take to develop similar





**Fig 2** An integrated package of software supports all Lisa functions. The operating system kernel draws upon applications through system calls to the software library.

modules; it saves space on the disk; space in memory; and, most importantly, guarantees consistency among the applications. There are modules to display and select from pulldown menus, to allow viewing of different portions of a document via scroll bars, to display alert messages informing the user of errors, to enter and edit single lines of text, to perform floating point computations, and to access database information.

The level of capability and power provided by Lisa's operating system and its library of software modules goes far beyond any traditional system. But, providing additional capability allows the Lisa system to present the user with a degree of simplicity never seen before in traditional operating systems. The high degree of integration between hardware and operating system software makes these advances possible.

*Please rate the value of this article to you by circling the appropriate number in the "Editorial Score Box" on the Inquiry Card.*

High 716                      Average 717                      Low 718