Table of Contents

Introduction

When Steve Jobs, co-founder of Apple Computer and proponent of the Macintosh, was fired from Apple by CEO John Scully earlier this decade, he vowed to start a new company and produce a next-generation computer as revolutionary as the Mac. His new company, NeXT Inc., promised the computing world everything they could ask for, and was more than a year late in introducing their first product, named The Cube.

Those who follow the trade publications already know the machine's promising hardware specs: 25 MHz 68030 CPU, 68882 floating point math co-processor, Digital Signal Processor (which can also be used to process data as well as sound), 8MB RAM (expandable to 32 MB), the world's first embedded Read/Write optical drive, 400 dpi laser printer, built-in speakers housed in the CRT, and a packaging design that would make Darth Vader proud. All this (minus the laser printer) for academic institution price of $6500!

Plus a few fresh approaches: The underlying operating system was re-written from the ground up by Carnegie-Mellon University using Objective c, eliminating what they considered to be inconsistencies and historic anomalies of Unix and, at the same time, adding a solid foundation for a parallel-processing environment across networks. The windowing system is the first to use the Display Postscript language, thereby guaranteeing that the image on the printer will be identical to that on the screen (something Macintosh users don't always get).

Originally the Cube was to be marketed to academic institutions only. Since then deals with Businessland and Canon, Inc. have created two commercial outlets for "the rest of us" to purchase one at an inflated price.

After evaluating a machine (with 0.9 Beta software all around) on loan to us for one month, here are some general impressions (more details will follow): The unit performed a little slower than a Sun 3/260 containing a 25 MHz 68020. The operating system is a hybrid of Berkeley UNIX and an Apple Macintosh (an unlikely combination but one that was well thought-out), and the 0.9 operating system release, like all beta versions of software, was full of bugs. Innovative applications, such as Voicemail and the Interface Builder (a treat for anyone who's ever had to deal with the Macintosh's myriads of complex system calls) show what's possible with the re-designed Unix. The volumes of bundled software, reminiscent of the way Kaypro used to sell their CP/M machines, promises to give a basic tool for every area of computer-aided productivity, especially software development.

Bundled Software, bells-and-whistles

Eleven applications, including four development tools, are bundled with the NeXT computer system. They are:

1) The NeXT user interface, called NeXTStep (tm), which performs the same function as SunView, Macintosh or X windows (but is not compatible with any of them).

2) WriteNow - The Macintosh's most heralded word processor has been ported over to NeXT. (If you enjoy writing with mice, this is the program for you.) Three other editors are available if you want no frills -- Jot and Edit, for use in the windowed environment, and vi, for use anywhere EXCEPT a shell window. (See the section on Mach Operating System for an explanation.)

3) Mail - like Unix' mail program but also allows picture and voice mail to be sent just as effortlessly. Although we had slight difficulty keeping the machine from forwarding mail to itself, the idea behind it the utility is superb. Just drag icons of whatever you want to send into the mail space (programs, postscript images, sounds) and click "Deliver". No FTP -i, no binary mode, no uuencode, etc. If you're interacting with other non-NeXT machines, it can deal with text-only messages and be completely compatible.

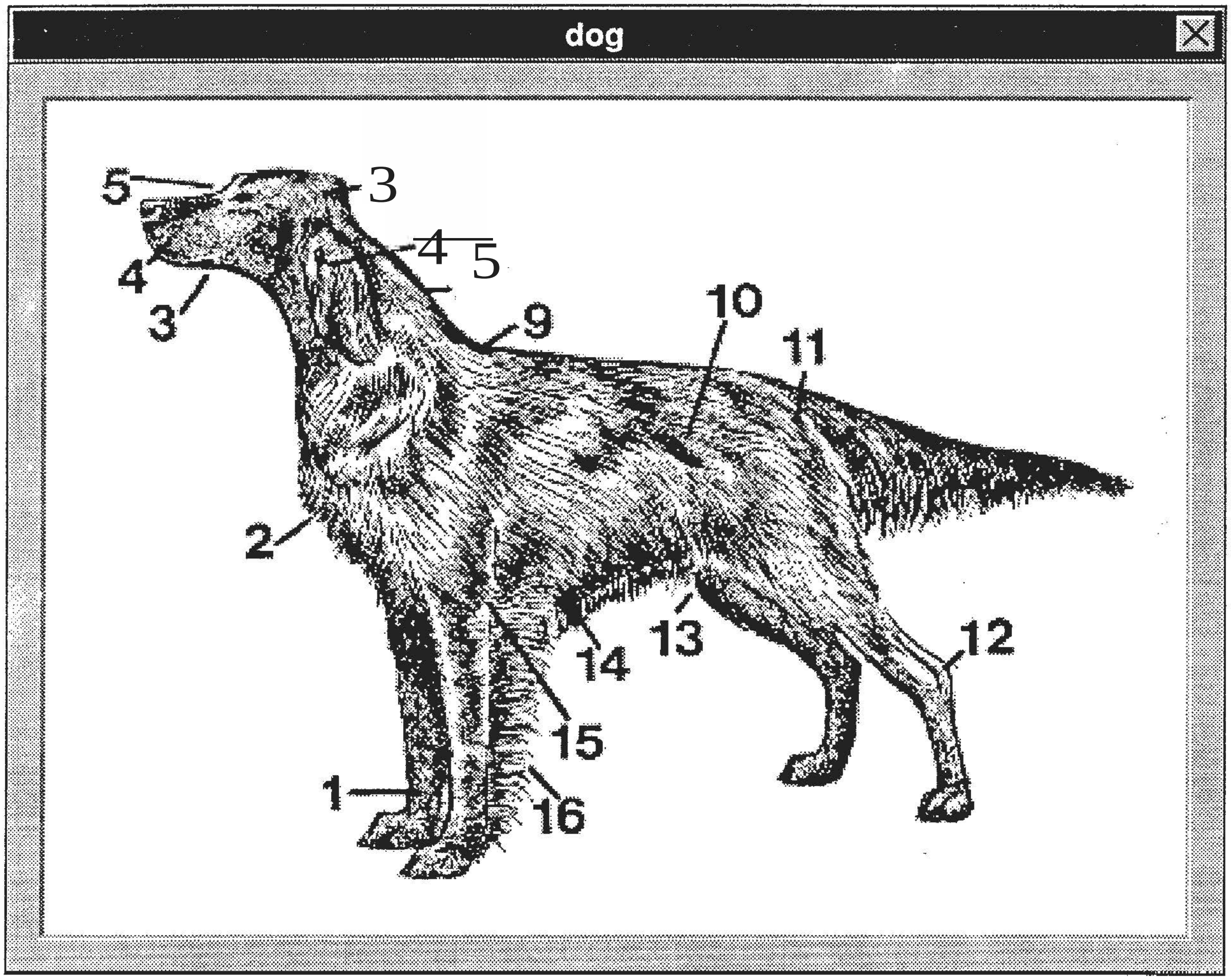
(One problem with having so many versatile tools: we were able to completely change the meaning of a voice mail message by using the sound editor tool to take out the word 'not'. Not healthy since it is the nature of Voice Mail to be trusted (i.e., not tampered with) more than text messages.)

4) The Oxford Dictionary of Quotations, and Merriam-Webster's Collegiate Dictionary and Thesaurus are both on-line and, once 'launched', are instantly available.

5) The Digital Librarian, which can search through anything quite quickly. It comes loaded with all the formatted UNIX man pages, all the NeXT documentation, and the complete works of Shakespaere. (How many times does the valiant taste of death? Gee, let's find out ...) You can also add your own documents, backlogged mail, old memos, etc. for it to search through.

6) Jot, a personal database. It's kind of like a general purpose note pad except the data is easier to extract.

7) Statlab, a general-purpose data analysis and graphing tool. We wanted it to filter and prepare all the graphs presented in the benchmark article (elsewhere in this issue), but a bug prevented it from reading an outside file.


dog
5
3
4
5
4
3
9
10
11
2
13
14
12
15
1
16

· dog \'dog\ k) l cdpk ^qqpf_
[ME, fr. OE al `d^Z
(bef. 12c)
1: CANID:
a: a highly variable domestic mammal %@^kfpc^j fif^ofp&closely related to the common wolf %@^kfp irmrp&
b: a male dog
2a: a worthless person
b: FELLOW' CHAP (a lazy al d&(you lucky al d&
3a: any of various usu. simple mechanical devices for holding, gripping, or fastening that consist of a spike, rod, or bar
b:ANDIRON
4a: SUN DOG
b:FOGBOW
5: uncharacteristic or affected stylishness or dignity (liked to put on the al d&
6 `^m7 either of the constellations Canis Major or Canis Minor
7 mi7 FEET
8 mi7 RUIN (go to the al dp&
9: one inferior of its kind: as
a: an investment (as a stock or bond) not worth its price
b: a slow-moving or undesirable piece of merchandise
10: an unattractive girl or woman
11: 1HOT DOG
- dog•like \'d6-,glik\ ^ag

/ dog ^ag
(14c)
1:CANINE
2: SPURIOUS; bpm7 unlike that used by native speakers or writers %al d Latin) %al d French)

0dog sq dogged; dog•ging
(1519)
1a: to hunt or track like a hound
b: to worry as if by pursuit with dogs: HOUND
2: to fasten with a dog
- dog it: to fail to do one's best: GOLDBRICK

8) Preferences, a utility for setting system preferences, similar to the Control Panel on the Mac. Use this to set your password instead of the /etc/passwd command! (This is an important area of incompatibility -- see **The Mach Operating System** section for more details and ramifications.) It also allows any user to change the system date and time quite accidentally, a security violation so severe that this feature is expected to go away with release 1.0.

9) NeXT Interface Builder - the most innovative aspect of the system, allowing the programmer to say, "Here's how I want my windows to work." and the Interface Builder generates the source code for you. All you have to do is program the smarts. The Interface Builder is covered in greater detail in a later section.

10) Mathematica. This product not only handles sophisticated and complex computations, it also comes (in the NeXT version only) with an "interactive textbook" user interface for development of mathematical courseware.

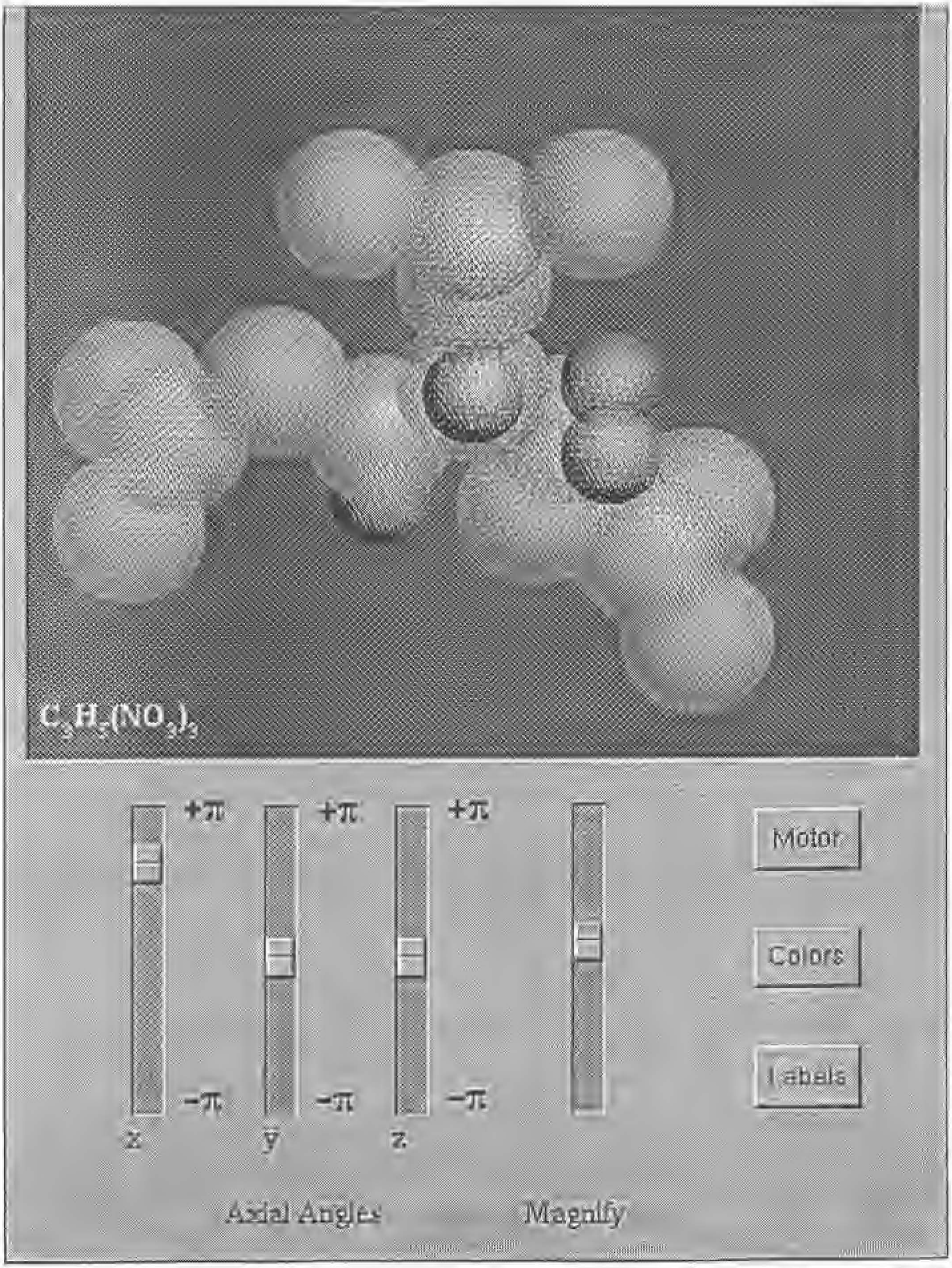
11) Sybase SQL Server database management system. (Will be bundled with O/S release 1.0.) This is completely compatible with the Sybase system that is used on the SFOC baseline. (According to a Sybase spokesperson, they don't have enough people to make the product incompatible.)

In addition, there are lots of other bells-and-whistles which enhance the immediate gratification the designers tried so hard to provide. For starters, the computer comes with its own NeXT T-shirt and stickers as standard equipment. It also has its own version of Flight Simulator, Billiards, a pseudo-3D molecule viewing window, a poker program (which some FISTers complained was written by someone who doesn't know poker), and a neural-network simulation program in which a seal learns to balance a stick on its nose.

OS overview

Introduction

The NeXT has a schizoid operating system; it tries to behave both like a Macintosh and a Berkeley Unix system. Although the two have successfully been rolled into one, thorough coverage necessitates discussing each aspect separately.


$C_3H_5(NO_3)_3$
+π
+π
+π
Motor
Colors
Labels
−π
−π
−π
x
y
z
Axial Angles
Magnify

NeXTStep (tm) User Interface

NeXTStep (tm) is Yet Another User Interface, trying to improve on the existing X Windows, SunView, Macintosh, and Presentation Manager standards. Like all these programs, NeXTStep uses a mouse to select windows, move icons, and "launch an application" (NeXT terminology for "starting a program").

Watching first-time users struggle with the "intuitive" user-interface suggests that, like the Macintosh interface, there's nothing at all intuitive about it. What is usually meant by the term 'intuitive' is that there are about seven easy-to-remember things about the interface that you have to learn; once learned they can be universally applied across all applications to facilitate easy navigation and self-help.

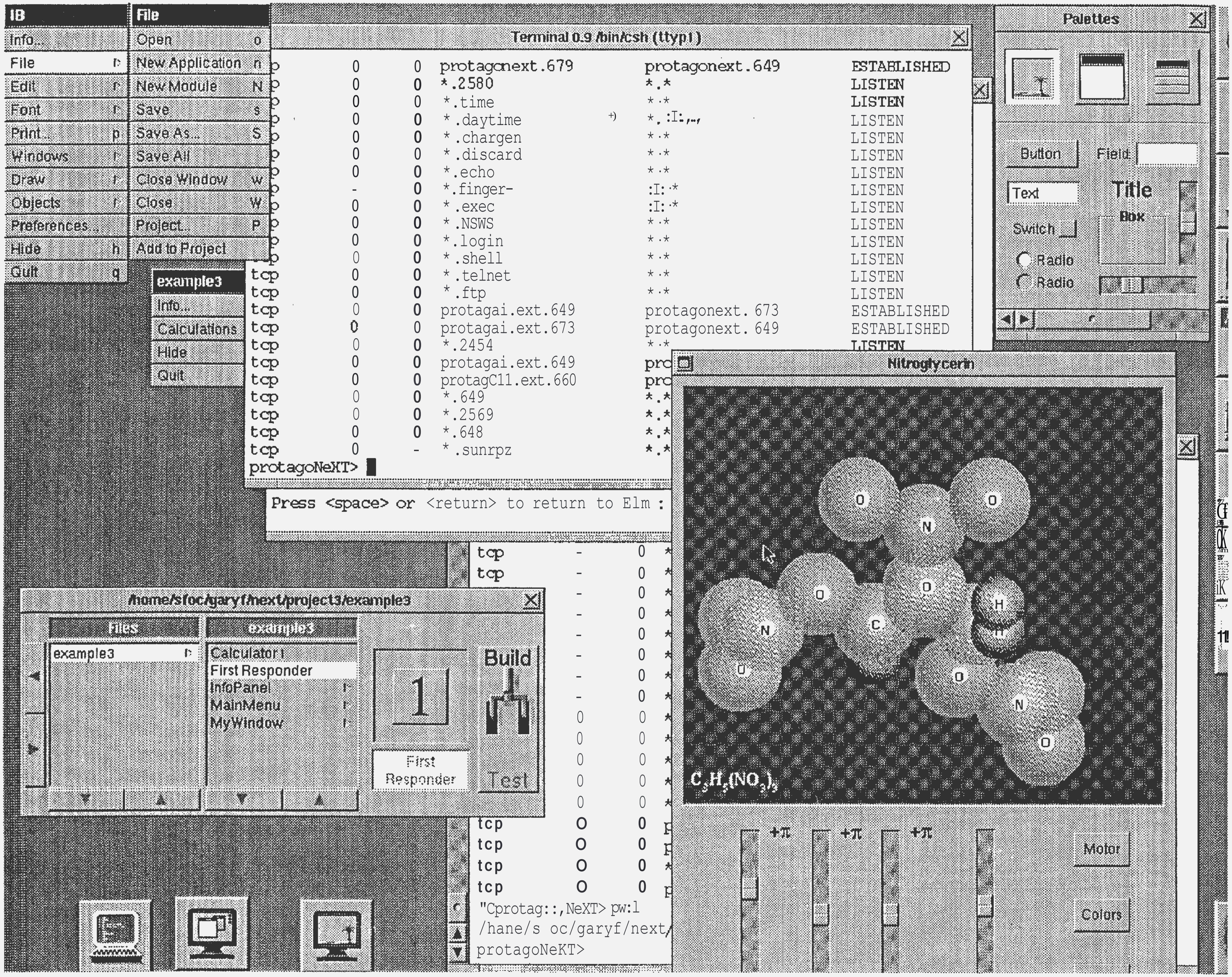
[The seven things are: mouse positioning, clicking and dragging to move an item, double-clicking to start a program, multiple-selecting using the shift key, clicking on folders to change directories, the function of the trash can, and using the point-and-click menus to display most available options.]

The machine can be configured to behave like a Macintosh, that is, it doesn't request a login name, and works stand-alone. Anybody doing anything in this mode is given the user name 'me'. Giving 'me' a password using the preferences utility (not by using the /etc/passwd utility; see **Mach** Operating System **Differences** section later for an explanation) and rebooting initiates the standard login procedure.

There are several important differences between the NeXT interface and the Mac. The most visible are the small yet intentional differences in appearance from that of the Macintosh, designed to avoid "look-and-feel" lawsuits Apple has brought against other firms in the past. The familiar trash can icon has been replaced by a black hole which becomes animated when its fed valid trash. The 'close' box associated with windows has been moved to the opposite corner, and buttons and mouse options have been given a raised 3-D appearance. The scroll bar, which appears at the left side of most windows, actually scrolls text while it is moved, a tiny but effective improvement for roaming through large files.

[Figure 1, the browser, goes here.]

Icon-based directories like the Mac's are still available, but quickly become cumbersome when used with Unix' deep hierarchical file structure. NeXT has developed a better way of navigating through deep directory structures called the browser (see Figure 1). Triangles pointing to the right indicate that this item is a directory, when highlighted the column to the right shows its contents.



```
Terminal 0.9 /bin/csh (ttyp1)
p        0      0   protagonext.679      protagonext.649      ESTABLISHED
p        0      0   *.2580               *.*                  LISTEN
p        0      0   *.time               *.*                  LISTEN
p        0      0   *.daytime            *.:I:,..,            LISTEN
p        0      0   *.chargen            *.*                  LISTEN
p        0      0   *.discard            *.*                  LISTEN
p        0      0   *.echo               *.*                  LISTEN
p        -      0   *.finger-            :I:.*                LISTEN
p        0      0   *.exec               :I:.*                LISTEN
p        0      0   *.NSWS               *.*                  LISTEN
p        0      0   *.login              *.*                  LISTEN
p        0      0   *.shell              *.*                  LISTEN
tcp      0      0   *.telnet             *.*                  LISTEN
tcp      0      0   *.ftp                *.*                  LISTEN
tcp      0      0   protagai.ext.649     protagonext. 673     ESTABLISHED
tcp      0      0   protagai.ext.673     protagonext. 649     ESTABLISHED
tcp      0      0   *.2454               *.*                  LISTEN
tcp      0      0   protagai.ext.649     pro
tcp      0      0   protagC11.ext.660    pro
tcp      0      0   *.649                *.*
tcp      0      0   *.2569               *.*
tcp      0      0   *.648                *.*
tcp      0      -   *.sunrpz             *.*
protagoNeXT>
```

```
Press <space> or <return> to return to Elm :
```

```
tcp      -      0   *
tcp      -      0   *
         -      0   *
         -      0   *
         -      0   *
         -      0   *
         -      0   *
         -      0   *
         0      0   *
         0      0   *
         0      0   *
         0      0   *
         0      0   *
tcp      O      0   p
tcp      O      0   p
tcp      O      0   *
tcp      O      0   p
"Cprotag::,NeXT> pw:l
/hane/s oc/garyf/next/
protagoNeKT>
```

Replacing the symbol for the internal disks on the right-hand edge (not visible in Figure 1) is a concept called the Dock, a handy place to keep your most commonly-used application icons. When you need to launch an application ('launching' is NeXT's terminology for starting a program), just move the mouse over and double-click; there's no need to navigate through multiple directories and clutter up the screen.

Some inconsistencies of the Mac interface have been eliminated. For example, on the Mac clicking and dragging an icon from one directory to another will move the file, whereas the same operation from one disk to another will copy it instead. Dragging things to the trash can normally 'trashes' files, except when you drag a disk icon there, in which case it only ejects the disk without erasing anything. With NeXTStep, every mouse operation is a 'move', unless you intentionally hold down the *command* key during the operation, in which case it performs a 'copy'. No more guessing what an operation is going to do.

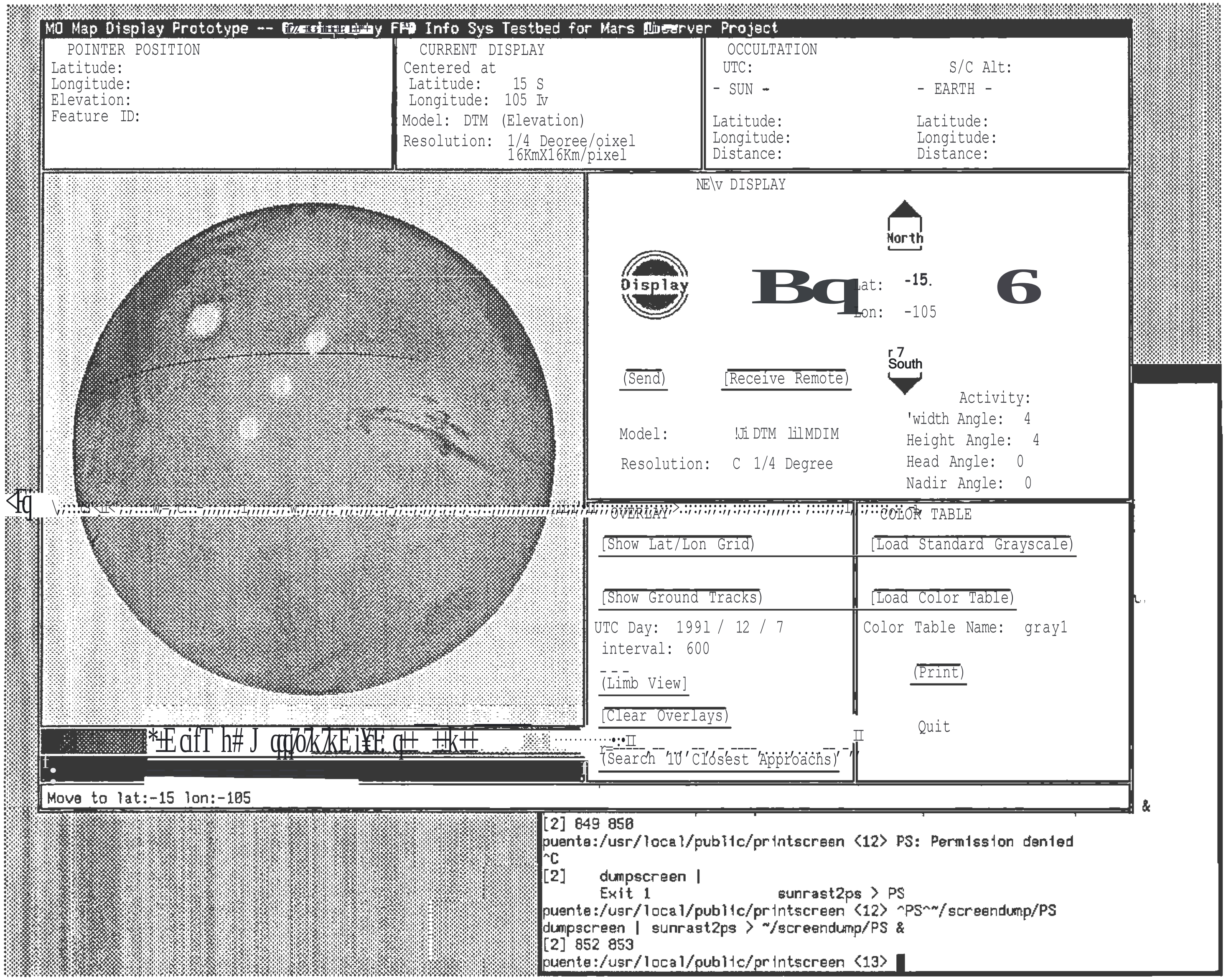
All the windowing is done using the Display Postscript language, an extension of Adobe's original Postscript page description language which was first adopted by Apple's laser printers. Improvements have been made to fill shapes with varying degrees of transparency, and to increase execution speed, all while maintaining 100% compatibility: if it prints on your old postscript printer, it will print here, usually much faster and with higher resolution.

(If you had read all the NeXT literature and had a feeling of how complex Display Postscript is and the intricacies of all the system calls, you'd watch it work and say, "Wow! It sure is quick considering all it's doing!". On the other hand, if you're simply a user who's used to working on a Mac II, you don't really care what extravagant paths it takes to get the job done, and might conclude that MacDraw and MacWrite are faster on the Mac and what's the big deal, anyway?)

This method of screen imaging via Postscript is very different than that employed by the Macintosh, with two very important benefits:

1) The printer image will be *exactly* what appears on the screen. Macintosh application software must draw the screen image one way, and then translate the image into Postscript to send to the printer. Bugs in the translation software have often resulted in the printer image being slightly different than what the screen shows.

2) Because no extra translation occurs, a printed image can appear faster. The Display Postscript image only has to be interpreted once (for the screen), then the CPU generates a 400 dpi image in memory and downloads it to the printer via



MO Map Display Prototype -- [illegible] Info Sys Testbed for Mars Observer Project

POINTER POSITION
Latitude:
Longitude:
Elevation:
Feature ID:

CURRENT DISPLAY
Centered at
Latitude: 15 S
Longitude: 105 Iv
Model: DTM (Elevation)
Resolution: 1/4 Deoree/oixel
16KmX16Km/pixel

OCCULTATION
UTC:
S/C Alt:

- SUN -
Latitude:
Longitude:
Distance:

- EARTH -
Latitude:
Longitude:
Distance:

NE\v DISPLAY

Lat: -15.
Lon: -105

Model: DTM MDIM
Resolution: C 1/4 Degree

Activity:
'width Angle: 4
Height Angle: 4
Head Angle: 0
Nadir Angle: 0

OVERLAY

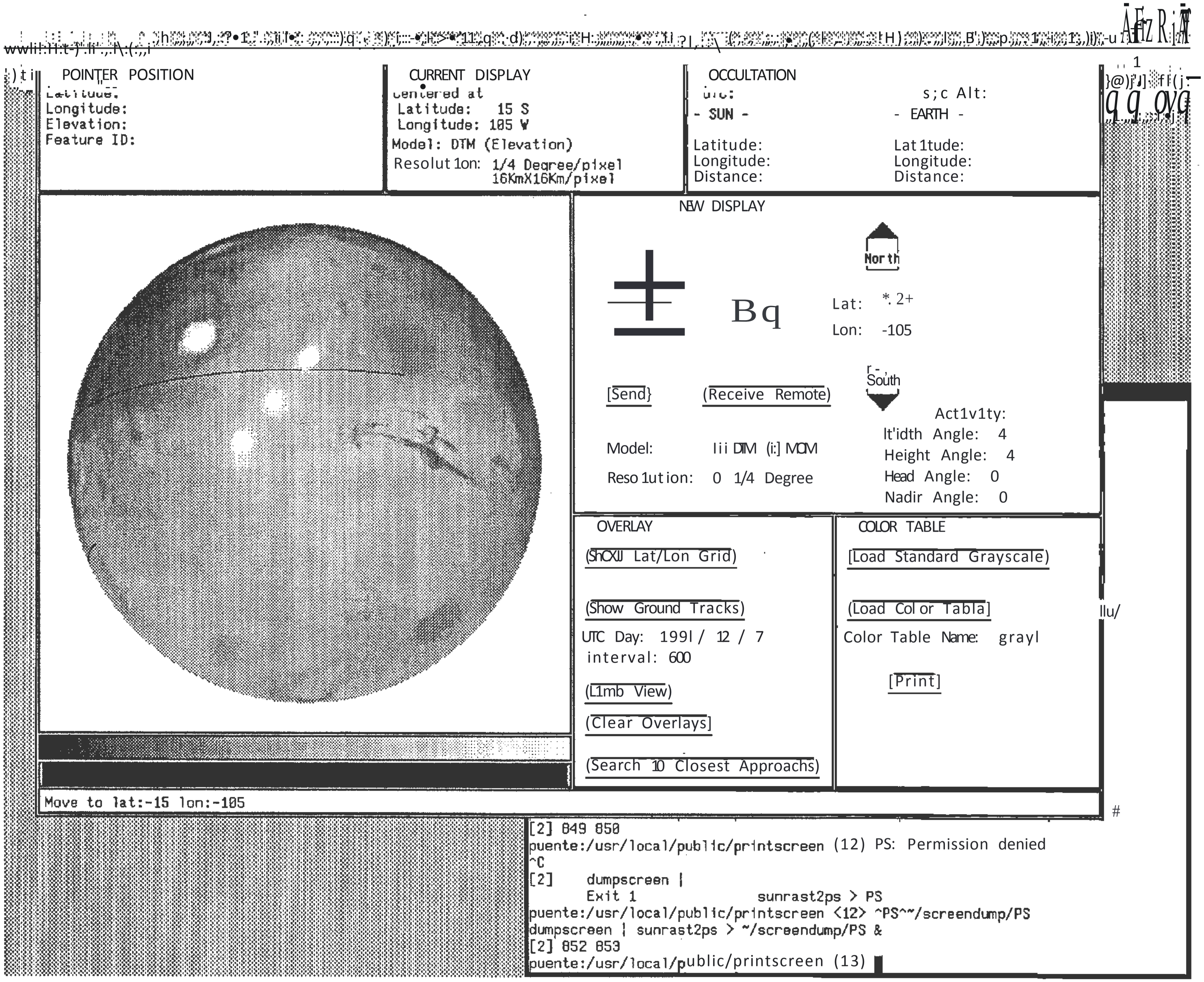
UTC Day: 1991 / 12 / 7
interval: 600

COLOR TABLE

Color Table Name: gray1

Move to lat:-15 lon:-105

```
[2] 849 850
puente:/usr/local/public/printscreen <12> PS: Permission denied
^C
[2]    dumpscreen |
       Exit 1                 sunrast2ps > PS
puente:/usr/local/public/printscreen <12> ^PS^~/screendump/PS
dumpscreen | sunrast2ps > ~/screendump/PS &
[2] 852 853
puente:/usr/local/public/printscreen <13>
```



POINTER POSITION
Latitude:
Longitude:
Elevation:
Feature ID:

CURRENT DISPLAY
centered at
Latitude: 15 S
Longitude: 105 W
Model: DTM (Elevation)
Resolut1on: 1/4 Degree/pixel
16KmX16Km/pixel

OCCULTATION
UTC:
- SUN -
Latitude:
Longitude:
Distance:

s;c Alt:
- EARTH -
Lat1tude:
Longitude:
Distance:

NEW DISPLAY

Lat: *. 2+
Lon: -105

Act1v1ty:
It'idth Angle: 4
Height Angle: 4
Head Angle: 0
Nadir Angle: 0

Model: lii DTM (i:] MDM

Reso 1ution: 0 1/4 Degree

OVERLAY

UTC Day: 199l / 12 / 7
interval: 600

COLOR TABLE

Color Table Name: grayl

Move to lat:-15 lon:-105

```
[2] 849 850
puente:/usr/local/public/printscreen (12) PS: Permission denied
^C
[2]    dumpscreen |
       Exit 1                sunrast2ps > PS
puente:/usr/local/public/printscreen <12> ^PS^~/screendump/PS
dumpscreen | sunrast2ps > ~/screendump/PS &
[2] 852 853
puente:/usr/local/public/printscreen (13)
```

a high-speed dedicated OMA channel. The trade-off is an additional CPU load; while printing a bit-image file the computer can become annoyingly slow.

[include two screendumps of Mars below.]

Figure 2a below is a 8-bit-deep bitmap image of a Sun color display that was sent to a 300 dpi Sun LaserWriter. The 2MB file took about 40 minutes to download at 9600 baud. Figure 2b is the same file routed to the 400 dpi NeXT printer; this one took 3 minutes and 45 seconds and shows much greater detail and grey level. The dramatic increase in speed is due to benefit #2 above.

The printer is so compatible that the cube can be used as a print server from a network of Suns. Files from Suns that are text only will also print at higher resolution with no additional commands. Postscript support also comes in the form of 2 different previewing programs, and there's even a C function that allows you to define your own unique Postscript routines as C function calls.

If you are an applications developer, you will relish a new developer's tool (it's so new it still has lots of bugs) called the Interface Builder. It shields the programmer from having to learn the conventions and know all the system calls associated with making the window interface conform to the guidelines. (Making the user interface 'look right' is usually half the task of writing a Macintosh application; the amount that has to be learned in order to do it right can be overwhelming.) The interface builder is covered in greater detail later.

Mach Operating System

Features

As capable as the User Interface is, developers will still have to access the Unix environment for compiling and debugging. NeXT's version of Unix, called Mach, behaved like an old friend and appeared to be the Unix we all knew and loved. (Well, maybe loved is a bit strong ...) But as this section explains, Mach is actually quite different internally.

The manual's design philosophy section tells how the original UNIX was a great idea but over the years it has grown to unmanageable proportions, and in doing so it has become inconsistent and difficult to master (I guess they mean as compared to mastering the OLD Unix). They stress that Mach returns to the original simplicity and is conceptually very different from BSD. Although the differences (outlined below) would normally mean re-writing code to run, Mach designers decided to implement all the original Berkeley functions by emulating them in software and building them on top of their new concepts. Therefore, despite internal differences, Mach is guaranteed to be upward compatible with 4.3 BSD: all one has to do is recompile.

Mach has done an excellent job at achieving this goal of compatibility. When logging in from a remote machine, it was difficult to tell (aside from the prompt) that it was a NeXT. (It even came with the 'less' text browsing program, normally a Berkeley add-on!) Within the first week, we were able to get our evaluation machine to run NFS and Yellow Pages and coexist comfortably with FIST's array of Suns, Apollos, IBM PCs, Masscomps, and DECstations.

Every program brought over that didn't try to access a windowed environment such as X or SunView would recompile with little or no trouble; and if there was trouble it was usually traced back to the Sun's compiler being lax in enforcing sloppy coding techniques. (Slight differences in obscure libraries slipped us twice, too.) Everything from benchmark programs to a VNESSA client (a program which would display live data from Voyager 2) would run.

This level of compatibility is even more impressive when one considers the very fundamental differences underlying Mach. Whereas standard UNIX uses shells, parent, and children processes, Mach employs tasks and threads. Conceptually they are very similar, but Mach claims lower system overhead, more memory allotted to each, and easy adaptation to multiple CPUs without any kernel changes. Each task can support multiple parallel threads, each inheriting attributes of its owner.

Inter-process communication is accomplished via two kernel abstractions, the port and the message, which are both similar to virtual circuit ports and inter-process pipes. The advantage here is that it works regardless of where the other process is; it could be under another task or another node. For security, a process must have the proper send and receive privileges before it can be accessed.

Exception handling has been generalized to handle both signals and exceptions easily. One clear advantage of this is that a debugger can operate on any process, parallel and/or distributed, from itself or any other remote machine rather than on only its own children.

Another automated tool is the Mach Interface Generator, which automatically generates remote procedure call code; all you have to do is specify some of the other machine's characteristics.

Mach was written in Objective C, an object-oriented language which is upward-compatible with C. Gnu has supplied an excellent compiler and debugger for this language, which is similar to but not compatible with the C++ language that is rapidly becoming an industry standard.

Differences

If it seems contradictory that an OS claiming complete compatibility can have so many differences, well, it is. Our first encounter with these "compatible differences" was with system administration.

System Administration Differences

The new 'friendly' System Administration and Network Management procedures are two examples of how NeXT plans to alter the UNIX method and show the world how they think it ought to be done. The UNIX way employs many files and bits of information scattered around and controllable only by experienced people who understand its evolved structure. Their new method is completely incompatible with the standard UNIX way; it employs a graphical interface and stores its information in different locations and in different formats.

That part isn't so bad; some think that UNIX is needlessly cryptic and any common-sense inroads into making it more usable are to be welcomed. But the user interface, like the rest of Nextstep, is far from intuitive. Compounding this was the fact that there wasn't much documentation about it in the 0.9 release. (NeXT does supply compromise programs for UNIX die-hards that will translate information from the standard format into the NeXT format.) It took a few tries before we got it to work; once the final documentation comes out we suspect it will pose fewer problems.

Shell access

Despite of the glitz of the user interface, developers still have to start shells in order to access Unix to edit, debug, or compile. Two window programs are available: Terminal (a VT-100 emulator) or Shell (a csh emulator). Oddly, the capabilities of the two do not overlap; Shell can start processes and do standard UNIX stuff but isn't powerful enough to handle a vi or an EMACS session. Terminal can; but the most of the power of the shell is missing. The manuals only hint that there are future plans to merge these two applications and make Shell behave like a fully-capable Unix window as on other computers.

.app Directories

There were areas where differences were viewed as improvements. In the past users of new applications receiving only verbal instructions would be given a pathname and "Tell me what you think." Changing to that directory revealed a myriad of source, include, and support files and no hint as to which file to run first.

This trivial problem which can be annoying to beginners was addressed by the Mach designers, and they developed the .app directory suffix. When viewed from NeXTStep, any directory ending with .app will appear as a file. When double-clicked, the OS moves into that directory and automatically starts the executable with the same name. This way, support files, source and object code won't clutter up the directory and cause extra confusion. This feature disappears when doing a conventional 'ls' from a shell window or from a remote terminal, however, so programmers won't be fooled.

As mentioned earlier, we also encountered slight differences in the standard libraries and include files on two occasions. They were easily caught and fixed, but proved to be a bother when porting software.

Dwrite, Dread

The concept of environment variables (manipulated with the setenv command) has been replaced by a non-ASCII file called ~/.NeXTdefaults that holds the environment variables. Three utilities — dread, dwrite, and dremove — are provided to alter these parameters in this file. For example, the command

```
     dwrite Terminal Fontsize 13
```

will remind the terminal program upon startup to change the default font. Most variables for 0.9 release software were undocumented, making this utility very difficult and unfriendly; it is NeXT's goal to require all applications to provide "Preferences" options in their pull-down menus which will shield the user from performing the dwrite and dread operations.

Other Differences

Every NeXT machine has a 68882 math chip installed, so they saw no need for a -f option for the Gnu (objective) cc compiler. Often Makefiles had to have this option removed for successful compilation.

Two extra serial port drivers are available in /dev in addition to the standard tty[a,b): ttyd[a,b) and ttycu[a,b). Each new driver has been optimized to interface with telephone modems that receive calls and telephone modems that initiate calls.

FIST users who tried the machine only once found a mysterious .places file and a .NeXT subdirectory in their home directories. The .places file is used to remember dock information and size and placement of windows. The .NeXT directory seems to be a place to hold all other defaults by several other applications.

The following utilities are not included in the MACH operating system, and there is no intention to include them in the future:

> adb, arff, bad144, badsect, dbx, dump.4.1, diskpart, efl, flcopy, fp, f77, gcore, learn, lint, lisp, pascal, rxforrnat, sendbug, and vlp.

Interface Builder

If you've ever developed for a Macintosh before, you're aware of the steepness of the learning curve: a few thousand system calls whose un-obvious calling order must be known. NeXT has a complex array of system calls, too, called the Application Kit. Knowing that the Mac's OS complexity has in the past driven away otherwise qualified developers, a solution called the Interface Builder was designed.

The Interface Builder (IB for brevity) is by far the most useful and innovative application in this system. It shields the programer from the complexities associated with Macintosh-like software development. Although the version tried was still full of bugs, the tool allows you to specify, using MacDraw-like techniques, what your windows, menus, icons, and mouse actions should look like. Then push a button and it generates skeletal source code and 'make' file for you! All you have to do then is add the brains to the program and you're finished.

This also represents the only area of NeXT in which it really helps to be familiar with object-oriented programming. Each item that is dragged into the window -- a text field, a menu item, a "button" (a place for a mouse to click on), -- represents an input or output object. Once added to the window the IB knows to generate an object for it in the source code, along with the color, size, position, and behavior attributes.

In another panel, the programmer makes a list of all the different mouse commands and I/O commands, and can graphically link the display object with a point in the source code. (See Figure 3.)

[Figure 3, the IB screen here.]

"My Window" in the example shown above is a simple Celsius to Fahrenheit converter program (although the "Calculations" submenu can also change its function to give square roots.) The black line shows graphically the connection between the Cale and its object, calculatorl.m, which was created earlier. After the source file is created it is then up to the user to fill in the skeletal functions "convertToSqRoot:·· and "convertToTemp:·· with useful algorithms and then compile.

That's a simple case, anyway. Even with a simple case, the IB made no less than five errors (four were easy to spot; the fifth took two telephone calls to uncover). Subsequent experiments with other aspects of IB proved so bug-ridden and frustrating (adjectives consistent with the rest of 0.9) that the tool's promise was duly noted and I went on to other things, all the while thinking what a difficult tool it must have been to build and that I'm glad I wasn't on the design team.

Network Compatibility

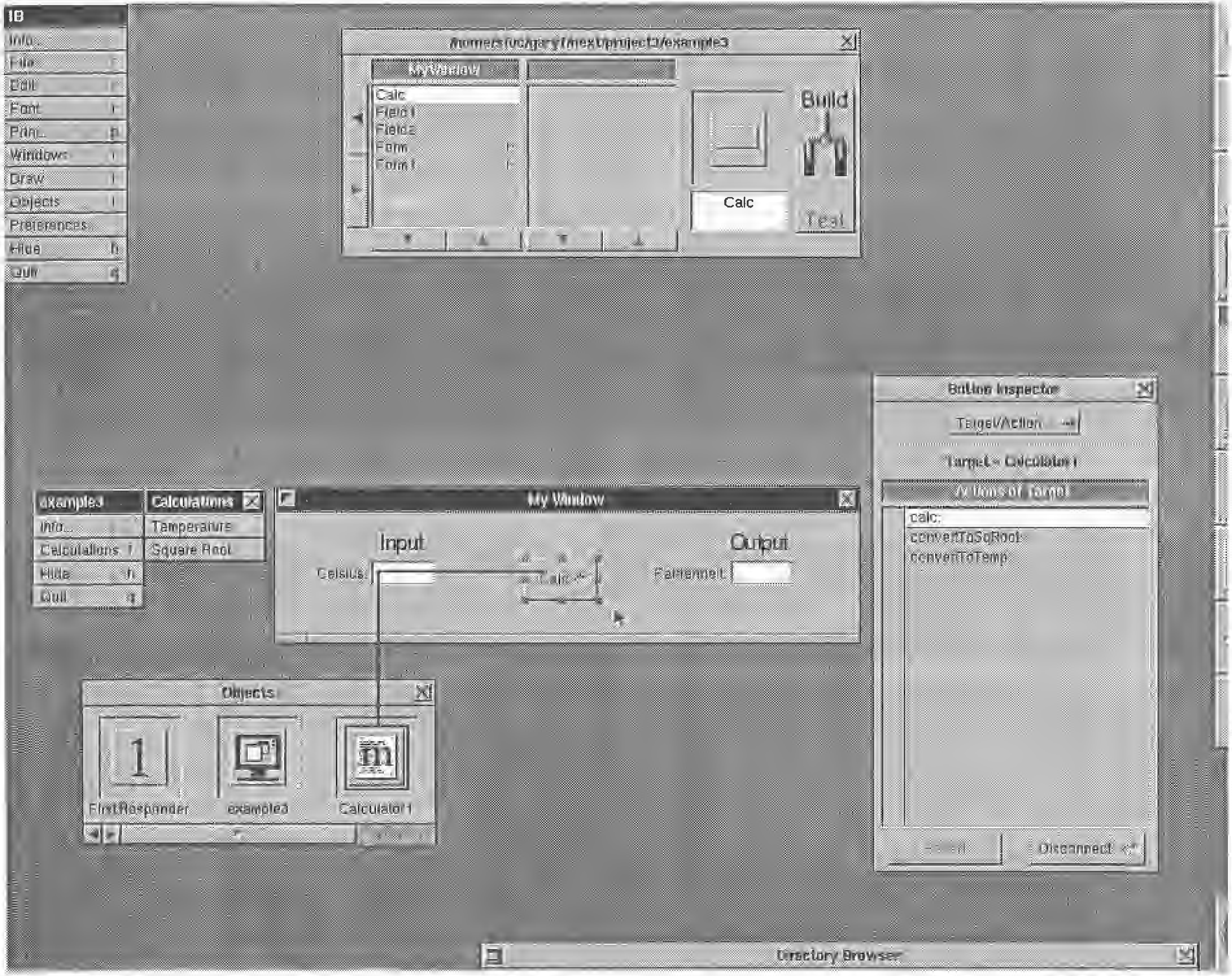
It is clear that the system designers thought that a network ought to be an integral element of the Cube; witness the ease in which objects can communicate with other processes across the network, the ability to debug anybody's code from a remote machine, and the lack of an inexpensive, low-density mass storage medium for the transference of software. A thin Ethernet (sometimes called Cheapernet) connector is provided at the back of the cube to perform these functions.

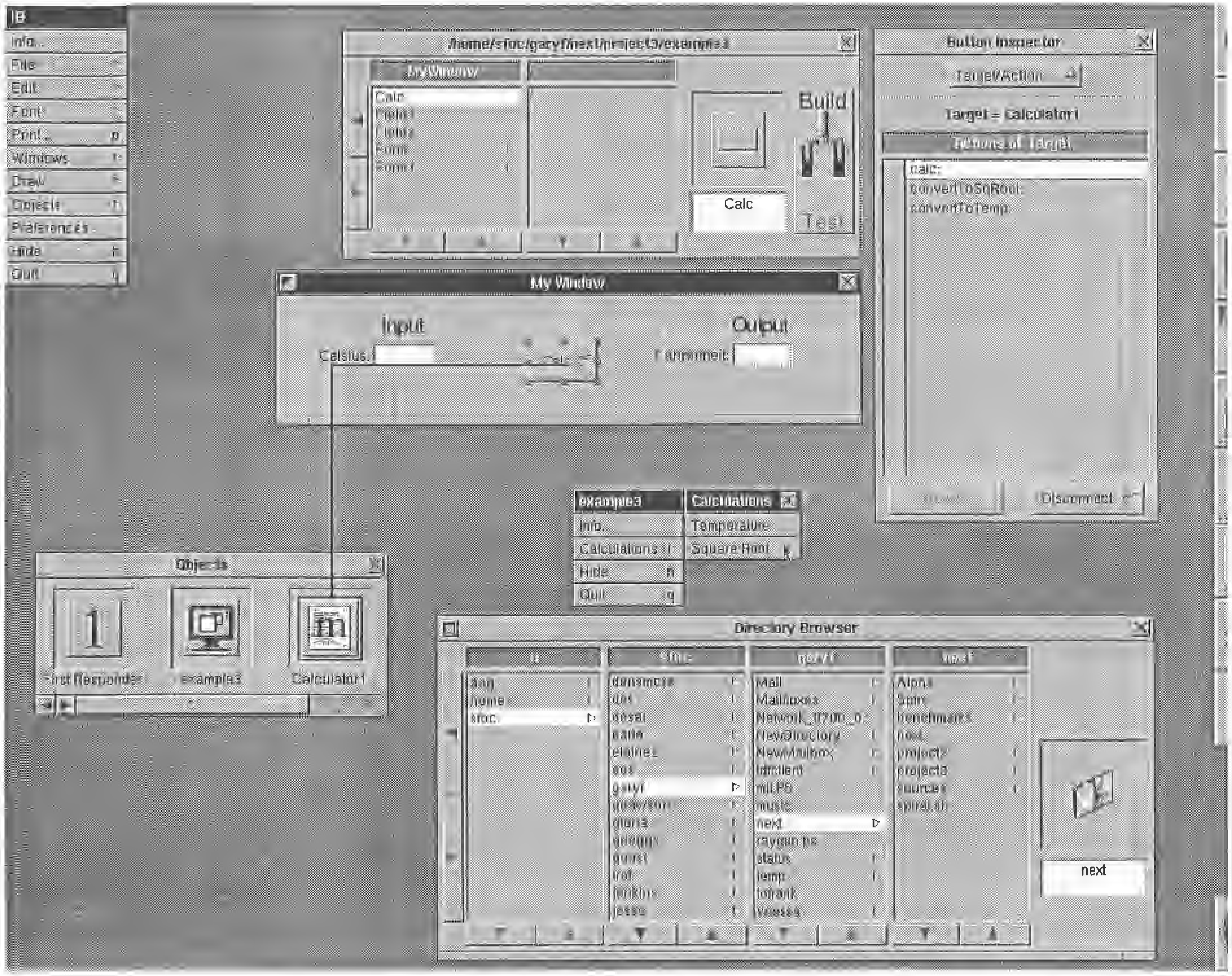
Translating the thin-wire to the FIST lab's thick-wire Ethernet was a little less than bothersome; all we did was extend the length of the existing thick wire with a thin-wire segment, and move the terminator to the new endpoint.

Because of Berkeley compatibility, we were able to get Yellow Pages and Network File System running on the NeXT within a few days, and when it ran it ran flawlessly. The big obstacle stemmed from their new method of system administration, and knowing when to use the new way and when to use the old. See the previous section, "Mach Operating System Differences" for more details.

Networking speed, as outlined in the previous section, was about 2.2 Megabits/sec, not as fast as other machines but still respectable. This was measured using the handshake-less UDP protocol; large file transfers using FTP or rep (which both worked fine) will yield a slightly lower rate.

The only network problem encountered was the failures of 'rup' and 'rwho', two programs which are supposed to tell the health of other networked machines and who is logged on to them. It is suspected that this is a NeXT <--> Sun problem.


IB
Info...
File
Edit
Font
Print...
Windows
Draw
Objects
Preferences...
Hide h
Quit q
/homes/uc/gary1/next/project3/example3
MyWindow
Calc
Field1
Field2
Form
Form1
Calc
Build
Test
example3
Info...
Calculations
Hide h
Quit q
Calculations
Temperature
Square Root
My Window
Input
Celsius:
Calc
Output
Farenheit:
Objects
FirstResponder
example3
Calculator1
Button Inspector
Target/Action
Target - Calculator1
Actions of Target
calc:
convertToSqRoot:
convertToTemp:
Select
Disconnect
Directory Browser


IB
Info...
File
Edit
Font
Print...
Windows
Draw
Objects
Preferences
Hide
Quit
/home/sfoc/garyf/next/projects/example3
MyWindow
Calc
Build
Test
Button Inspector
Target/Action
Target = Calculator1
Actions of Target
calc:
convertToSqRoot:
convertToTemp:
Disconnect
My Window
Input
Celsius:
Output
Fahrenheit:
example3
Info...
Calculations
Hide
Quit
Calculations
Temperature
Square Root
Objects
First Responder
example3
Calculator1
Directory Browser
sfoc
garyf
next
home
Mail
Mailboxes
NewDirectory
NewMailbox
music
next
status
temp
projects
sources
next

Security issues

The word 'security' was never synonymous with the words 'Unix' or 'Macintosh'. (Or the word 'network' for that matter.) There are two ways in Unix, for example, that a user can become root. The first requires that the user be trusted, by making him/her a member of the wheel group (for this you must use the Netinfo application in the new, friendly system administration interface rather than adding the name to /etc/group like the conventional way.) Once done, you can su to root without the need for a password.

The second way doesn't even require an account. Like a Sun, it is possible to reboot the system at the console by pressing two obscure keys simultaneously and boot single-user; effectively becoming root. Correspondence with NeXT personnel revealed that a you can employ a software patch similar to one that Sun uses to tackle this problem.

A security hole that appears to be inadvertently designed in occurs in the 'Preferences' panel, a program that each user can call up from the console and change things such as mouse speed, type of warning sound used, password, and some environmental variables. Some other things anyone can easily change are the system date and time, something a secure system is not supposed to allow.

The panel presents the variables as two appealing icons which casual browsers often mistake as a calendar program and randomly click on calendar dates thinking it will bring up an appointment menu. What they actually do is set the date. The same 'Preferences' panel can also allow anyone to specify the boot device.

If the System Administrator is familiar enough with the Interface Builder and proficient with .nib files, these undesirable 'Preferences' options can be removed; although it is not an obvious procedure. Many users have complained about this feature; it is expected that it will be removed in release 1.0.

Benchmarks

During our one-month evaluation we tried as many meaningful tests as we could and compared the performance data with that of other machines. This has been packaged in a separate article on benchmarking elsewhere in this issue. The initial ranking as mentioned in the introduction still stands; it's almost as fast as a Sun 3/260. It's Ethernet rate is low although its slow speed is not apparent to the user when using NFS. The read/write optical disk, although faster than the floppy disks which performs the same function, is anywhere from two to five times slower than it's internal hard disk.

If you want quantitative terms, please refer first to the benchmarking article. The last two bars in every category represent the NeXT machine using only the optical disk, and the NeXT using the internal hard disk.

Dhrystones

See FIST Quarterly Report #7 pg. 19 for background on the dhrystones. Of the six machines tested, NeXT ranked fourth and fifth (with and without the optical drive; dhrystones don't access the disk so both times as expected were equal). The average was 6350 dhrystones per second.

Read/Write Tests with and **without Optical Disks**

Several tests of disk efficiency were run, including file read/write, booting time, and multiple processes which require lots of swapping time. (Two of these are graphed in the benchmark article.) The results showed that the optical disk is anywhere from two to five times slower than the optional 660 MB **SCSI-based hard disk**.

People who look beyond the horizon at future trends (see Jose Urena's article on the future of optical storage elsewhere in this issue) may complain that the disc's 256 MB density is too low and the access time as shown above is too slow compared to next year's technology; while those who look at history recall that this is the price of being a pioneer, and besides, the whole NeXT machine costs as much as a third party's price for just a read/write optical disk peripheral. The incorporated optical disc represents a technology bargain.

Ethernet Speed

The Ethernet speed was disappointing; a mere 2.2 Mbps, roughly half the speed of our Sun 4/160. (Even the VAXstation, the machine which came in last by a large margin in every other test, was faster at 2.4 Mbps.) This may be a physical limitation of the >J Ethernet transceiver chip seated on the CPU board.

In spite of this measured speed, Ethernet-intensive operations such as NFS did not appear to be slow.

Pipes

Notice that inter-process pipes are slower than that of other machines. Keep in mind that, because of Mach's new method of inter-process communication, all standard Berkeley methods are emulated in software, thereby somewhat increasing the execution time. Regretfully, there was inadequate time to test the efficiency of the underlying Mach method for comparison purposes.

Float

In the floating point arena NeXT came in third, comparing favorably with our fastest machines. Normally this could be attributed to its next-generation 68882 math co-processor, but that does not explain why the process is so much slower when running off of the optical disk. One surprising thing was the relative slowness of the 68881 co-processor in the 3/260; it took last place. (The chip was indeed there; 8.5 seconds are required for the software routines.)

Generic Decommutation

As explained in the Benchmark article, the Generic Decom test was originally developed as a "typical JPL activity" during the days when we were procuring computers for a project which hadn't developed any software yet. It is disk intensive, as it reads a Voyager emulation data stream and writes its output to disk. The metric used is Maximum Theoretical Throughput (MTT), the highest throughput possible assuming the operating system wasn't busy with other processes with higher priorities. (Other tests were used to measure OS overhead.)

MTT was 46.1 Kbps; about 85% of that of a Sun 3/260.

The only trouble we had compiling the Generic Decom test was due to the lax nature of the Sun compiler; syntax mistakes and pointer mis-matches that had been tolerated by the Sun C compiler were rigorously flagged using the Gnu Objective C compiler.

Conclusions

Because of the similarities between the 68020 and 68030 architectures, we had fully expected the NeXT to come in somewhere between our 25 MHz 60820 Sun 3/260 and our SPARC-based Sun 4/260. Most CPU tests, however, rate it at slightly below the 3/260. There are three things working against the NeXT machine that help account for this:

The first is the fact that we are evaluating pre-release 0.9 software, which NeXT says will be slower than the official release of 1.0.

Second is the NeXTStep (tm) user interface, a high-overhead set of complex Display Postscript system calls with which every running program must share CPU resources.

Finally, it is important to keep in mind that Mach, due to its internal re-design, is not *really* compatible with BSD UNIX; rather compatibility is maintained via an extra layer of emulation software and aliases. Although it slows down the system by an unknown degree, it *does* allow it to meet an SFOC requirement to maintain complete software compatibility, something MACH is guaranteed to do throughout its lifetime of revisions.

Problems we had

Many of the problems found during evaluation were claimed by NeXT to be bugs that will be cleaned up by release 1.0. To their credit, it was a pleasure reporting bugs to an organization which was actively seeking such input; as they are fanatically working on release 1.0 and are prepared with a small army of programmers to squash the reported bugs. On public forums such as Usenet and Bitnet, NeXT representatives would respond to problems promptly with an intelligent answer, telling you where the problem lied, whether it was a bug or a feature, and how to fix it if you don't want to wait for the 1.0 release. They even built a slicker-than-ever bug reporting utility on top of their Mail utility, where you simply check off the application, give a few lines of description, and click the 'send' icon. The message then automatically gets routed to the proper team via the Internet.

As mentioned before, the whole 0.9 system seemed a bit sluggish; it is unknown how much the official 1.0 release will be tightened up to allow faster operation. The most bothersome results of the slowness is the way the whole machine (including the mouse) dies while "launching" applications and when printing bit-mapped images.

The 'talk' utility, which is supposed to divide your terminal screen into two halves and allow a full-duplex conversation with another user, didn't work. When invoked, one of two error messages appeared:

```
     talk: protagoNeXT: Can't figure out network address.
```

or

```
     You don't exist.  Go away!
```

(I feel that computers are arrogant enough nowadays without smug programmers adding to the intimidation.) A software patch was available via anonymous FTP on the many University support site bulletin boards, which I promptly downloaded and executed. No improvement.

We also had a lot of trouble with the mail routing utility, which kept trying (and failing) to route mail to itself. We finally changed the name of the machine to protagonext.jpl.nasa.gov since we discovered that was the destination of all the forwarded mail. (Before that it was trying to find that name from the non-existent name server root@jplproto (which is our domain name)). Now it works better; with the .forward files modified accordingly it will now forward mail to NeXT from everywhere except when it originates from our **VAX**, which runs Mt. Xinu.

Other problems, which should be cleaned up by the 1.0 release:

- Serial ports (which are sort of a +-5v RS422/RS232 hybrid) are rumored to not work reliably above 1200 baud.

- It's very difficult to eject an optical disk, mounted or unmounted.

- Doing a s in the VT-100 Terminal emulation window to stop program output would occasionally freeze the terminal program (not the program that was running under it) permanently. And since for some reason root owns the window process instead of the user, it was possible to kill the frozen process but you had to be root to kill the terminal program.

Finally, a complaint about industrial design. All other computers have at least one pilot light to indicate some sort of status, be it disk access or even whether the power's on. Printer manufacturers do this too, knowing full well that the feedback these LEDs provide is invaluable when setting up systems for the first time and troubleshooting them later on. NeXT, in their packaging arrogance, decided that their OS would be so solid they wouldn't need indicator lights to provide such essential feedback. (When we powered up the machine the first few times we weren't even sure it was on!) When the line printer daemon suddenly dies (as happened on many an occasion to us), the laser printer (the first thing usually consulted when troubleshooting on other systems) gives you no visual indication that it did/didn't receive data. Instead you're supposed to consult a software printer status panel which also had bugs, and would core dump when large files were in the queue.

Rumors and Promise

Rumors

If you've read this far you're probably contemplating getting a machine for yourself, although all those bugs and the lack of a large application library would make anyone justifiably hesitant. So to offset all the inherent characteristics of a new machine, allow me to mention some RUMORS (that's right, rumors, with their mandatory disclaimers: I have nothing to do with NeXT; don't own any stock, am not pushing their product; opinions are my own, etc., etc.) about future directions:

X Windows: According to PC week magazine, MIT is currently working to port X windows over to work inside the NeXTStep interface. The article says that X protocol will be emulated inside a display postscript window.

CUrveos File Security for NFS will be implemented.

Color: NeXT is developing color upgrade capability, but won't announce it until a full compliment of peripherals (like printers) become available. Canon (who makes color hardcopy printers as well as NeXT's 400 dpi laser printer engine) has recently invested $100 million in NeXT, no doubt the color printer technology will be shared to speed things along. NeXT also owns Pixar, a major graphics engine maker, and rumor has it that the 24-bit-deep color screen will utilize Pixar's Renderrnan scheme and will be available at "a very attractive price".

Many third party applications are on the horizon. NeXT's own literature says they're looking to have every major computer application (Lotus 1-2-3, Framemaker, Write Now, Mathamatica, Sybase) ported over to the machine; those we were able to try were so bug-ridden that it prevented any meaningful evaluation. One application was so bad that it displayed an unusually blatant disclaimer when started up:

> "...This is pre-alpha, unsupported demonstration software. Don't even dream of using this version for real work. Expect it to crash, do the wrong thing, confuse you, run like a snail, look ugly, clutter your disk with extraneous files, and do other nasty sorts of things pre-release software is notorious for..."

Other applications such as 'Digital Ears' promise to unleash the power of the built-in DSP chip, and 'Notater', a package that integrates the sound chip with a Musical Instrument Digital Interface (a standard method to have electronic keyboards communicate with each other) to turn the machine into a composer's workstation.

In the pre-pre-release software (0.8) it was clear that one goal was to have Sun binary files run directly on the NeXT without re-compiling. In 0.9 that didn't work, although the MAN pages mention a utility called 'atom' (g.out to Mach Object) which was supposed to help do the same thing. (Couldn't get it to work.)

And on a related note: IBM has purchased the future release 1.0 of the NeXTStep user interface; supposedly to bundle with their next generation 80486 PC boxes instead of Microsoft's Presentation Manager.

Future Promise

When the bugs are worked out and OS 1.0 is released, the NeXT machine, like all new computing platforms, will still be in short supply of applications and will have various system features un-exploited for a short while.

Their initial appeal to the hackers of academia should put the first dent into solving this problem, paving the way for essential development tools to be distributed via the university support network. At this writing there are 6 university sites acting as Next software and patch distribution sites. Their archived data, which at recent checking contained precious little, can be had by anonymous FTP across the Internet.

Some of the unique features, such as the built-in DSP, the Interface Builder, and the ease of multiprocessing make for a unique platform around which high-level data analysis applications can be quickly developed. And the DSP (in the hands of a knowledgeable programmer) can be used to speed up complex data analysis by orders of magnitude! (It also plays superb Compact Disc-quality music at your command for those long edit-compile-debug cycles.) Lastly, its Berkeley Unix compatibility also means existing software that doesn't use X windows should port easily.

To be sure, the machine we evaluated certainly shows a lot of promise. Whether it succeeds or not depends more on the market than its technical merit.