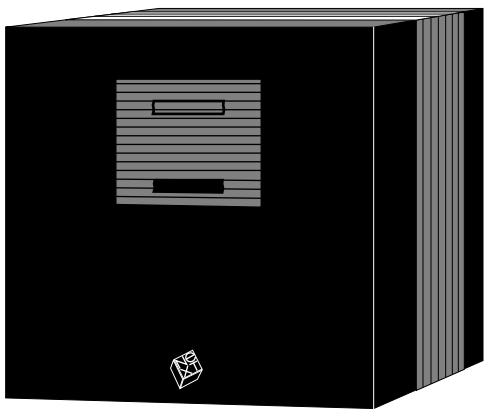
rmNUG

Rocky Mountain NeXT Users' Group Newsletter

February, 1991



In This Issue!

WordPerfect 1
Lotus Ships 2
Diagram! 3
AppleTalk/Unix 5
BaNG Meeting 6
NeXTstations Ship 8
340 HD's Upgraded 8
NeXT New Hire 8
Q & A 8
NeXT On Campus 8
NeXT v Mac 9
Job Listing 14
NeXTWORLD 15
Tech Alert 91-1 15
Member Profiles
Al Steele 15
Demetrius Johnson 16
January Meeting 16
From the Board 17
February Meeting 18
March Meeting 18
USENIX/Uniforum 18
ACM Meeting 21
Connectity Project 22
Software Program 22
rmNUG FTP Access 22
From The President 22
rmNUG Membership 23
NeXT Directory 24
UseNet Highlights!
Lighthouse 24
Fax Fact's 24
Fonts 25
SIMMS & HD's 25
From The Editor 28
NeXT Benchmarks 28
"Make" 29

WORDPERFECT FOR NeXT SHIPS!

Orem Utah, January 18, 1991

WordPerfect for NeXT began shipping today. This product is comparable in features to WordPerfect 5.0 for DOS, but will take full advantage of the Nextstep graphical user interface (GUI).

"WordPerfect for NeXT is designed for the NeXT user who is looking for a powerful but easy-to-use GUI word processor," said Royce Bybee, NeXT product manager at WordPerfect Corporation. "With full WYSIWYG, WordPerfect for NeXT users can display and edit headers, footers, and footnotes on the screen.

In addition, on-line documentation is included for easy access to the complete WordPerfect reference manual."

Graphics

Graphic placement in WordPerfect for NeXT is simple. By selecting a graphic file in the file viewer, you can drag it directly into your document. A graphic image may be placed anywhere on the page, scaled, moved, and rotated, and text will automatically wrap around it.

Ruler

The ruler lets you format and modify newspaper or parallel columns of any size. Justification, tab settings, line spacing, and margin settings can an be controlled on the ruler. Moving and deleting tabs or adjusting margins is as easy as clicking and dragging your mouse.

Printing

WordPerfect for NeXT will print to all Postscript printers supported by NeXT. In addition, WordPerfect for NeXT ships with all NeXT-supported fonts as well as fonts that display all of the characters in WordPerfect's character set.

Macros and Merge

The Macros feature lets you define and automatically execute a series of user actions. A macro editor is also included. The merge capabilities in WordPerfect for NeXT allow you to combine data from two or more different sources into a single document. This feature can be used for personalized memos, forms, and letters.

Compatibility

WordPerfect for NeXT files will be compatible with WordPerfect 5.0 for DOS. They will also be compatible with WordPerfect 5.1 for DOS files, with the exception of the new features in version 5.1 such as Tables and Equation Editor.

When a file is transferred from WordPerfect 5.1 for DOS to WordPerfect for NeXT the codes will be ignored; when the file is retrieved on DOS the features will appear and remain unchanged.

System Requirements

WordPerfect for NeXT runs on NeXT Computers using NeXTstep Operating System 2.0. WordPerfect for NeXT will list for $495.US/$645.CN. For more information, call WPCorp's Information Services Department at (801) 225-5000.

Lotus Announces Shipment Of Lotus Improv

Early Users Respond Enthusiastically To Spreadsheet's Innovation

CAMBRIDGE, Mass., February 4, 1991 --- Lotus Development Corp. today announced it has begun shipping Lotus Improv, its new spreadsheet for the NeXT computer, on schedule. Improv introduces a new approach to spreadsheets combining traditional spreadsheet metaphors, such as rows and columns, with new, breakthrough features, such as dynamic spreadsheet views, plain English formulas, and presentation graphics incorporating data, text, graphics, images and sound.

"Reaction to Lotus Improv has been enthusiastic," said Don Casey, vice president, Lotus Spreadsheet Division. "Early users have found that Improv enables them to view and analyze information faster and more dynamically than ever before, and agree Improv represents an exciting and significant step forward in spreadsheet technology."

Improv has been instrumental in driving customer interest in NeXT's new 68040-based computers. According to Todd Rulon-Miller, vice president of sales for NeXT, "Improv's unique approach to analyzing and viewing data offers our customers powerful, new capabilities not available on any other platform. It provides the level of innovation our customers need and expect from Lotus and NeXT."

Improv has been well received among spreadsheet users in corporations and higher education who rely on spreadsheets for financial planning, merger and acquisition analysis, market analysis, sales forecasting, data management, course administration, and grants management.

According to Eric Spahr, vice president at Shearson Lehman Bros., "In my opinion, Lotus has truly broken new ground with Improv. I need a spreadsheet that allows me to build models for forecasting and accounting and to restructure my data in any way I want. Improv enables me to do this quickly and easily, and the result is a spreadsheet my traders and customers can understand."

New Spreadsheet Capabilities For Improved Data Manipulation & Analysis

Improv enables users for the first time to quickly reformat the spreadsheet by creating multiple views of the same spreadsheet without re-keying information. With Improv's dynamic views feature, data can be rearranged to compare and gain greater insight into otherwise hidden data relationships; expanded to show more detail; hidden or grouped for clarity; and formatted for presentation needs. Any changes made to data in one view are automatically reflected in all related views.

Improv also allows users to name data cells and formulas using English words instead of letters and numbers. With English formulas and cell names, the structure and logic of the spreadsheet are immediately understood, while data accuracy and the ability to spot errors are enhanced significantly. Formulas are displayed separately in a window below the spreadsheet for easy viewing and modification and increased spreadsheet maintainability. Improv formulas can be applied to many cells at once reducing a spreadsheet's complexity. If there are errors in the spreadsheet formula, Improv immediately identifies them and, where possible, suggests solutions.

Lotus Improv helps users create advanced presentations combining data, text, graphics, images and sound for voice annotations. For building presentations, Improv offers full-featured drawing and charting capabilities. Charting options include 3D bar, stack, line, bar, pie, scatter, and area graphs. Users can rotate 3D bar graphs to show the bars from any perspective, and easily change font types, background colors and fill-in patterns of their graphs. Users also can add TIFF or EPS images such as those provided in a sampler of ClickArt illustration from T/Maker Company to customize and enhance presentations. Improv spreadsheets and graphs are linked dynamically so that any changes made to a worksheet are automatically reflected in the associated graph.

Improv imports and exports Lotus 1-2-3 spreadsheets and reads any other spreadsheet that writes.WK1,.WK3, or ASCII files, enabling users to make use of existing 1-2-3 data, exchange data, and combine efforts with users working on other platforms.

The rmNUG Newsletter is published monthly by the Rocky Mountain NeXT Users Group.

Readers are encouraged to send their comments or contributions to:

David Bowdish
73340.2146@compuserve.com

Any submissions of letters, artwork, articles, etc. will constitute implied permission for rmNUG to publish (in whole or in part) in print or electronically.

Sorry, but with our budget ($0) we can only afford to offer our sincerest thanks for any contributions you may send.

Special thanks to all who contributed to this newsletter.

Editor:
David J. Bowdish

Contributing Editors:
David R. Hieb and Brad Green

System Requirements, Pricing And Promotion Extension

Lotus Improv ships on 3.5 inch diskettes and runs on all NeXT computers with the NeXTstep 2.0 system software. Improv supports the NeXT 400 dots per inch PostScript Laser Printer and the Apple LaserWriter.

Lotus Improv has a suggested retail price of $695. However, **Lotus and NeXT have extended a worldwide promotion through March 31, 1991,** offering a free copy of Lotus Improv to customers who buy and register any of NeXT's new 68040 computers or who upgrade their current NeXT computer with the 68040 upgrade kit. Customers may request their free copy of Lotus Improv when they register with NeXT by calling 1-800-848-NeXT. Customers will receive the product directly from NeXT. After the worldwide promotion expires, Improv will be available through resellers and VARs that carry NeXT hardware and other NeXT products. Further details on these plans will be available at the end of March. For additional product information, customers in the U.S. can call 1-800-343-5414.

Lotus Development Corp., founded in 1982, develops, markets and supports business software and CD-ROM databases that help users access, analyze, communicate and share information. The company's first product, Lotus 1-2-3, is the most popular software program in the world with more than 14 million users. Lotus markets its broad range of products in more than 65 countries and offers users comprehensive support options, including 24-hour-a-day,7-day-a-week support in the U.S. for PC versions of its 1-2-3 spreadsheet.

Submission Policy

If you wish to submit an article or letter to the editor we ask that it be sent by email in the following format:

ASCII format.

Headline words should be capitalized.

No paragraph indents.

No Tabs or side by side columns.

Empty line between paragraphs.

Make sure spelling is correct, especially names.

Send To:

73340.2146
@compuserve.com

Introducing Diagram! for NeXT Computers

"A graphics tool for people who think and draw at the same time."

Lighthouse Design, Ltd., a developer of modern productivity software for NeXT computers, is pleased to introduce Diagram!, an innovative graphics program for NeXTstep release 2.0.

Diagram! is a general purpose drawing program that automates the more mundane tasks typically associated with drawing charts and diagrams. In so doing, it frees you to concentrate on design, experimentation and revision--not the tedious mechanics of drawing and redrawing.

Some of Diagram!'s time-saving features include:

Rubberbanding lines and curves

Rubberbanding lines and curves spare you the tedium of having to manually maintain connections and relationships in their drawings. Move an object, and Diagram! automatically moves all the lines drawn to it.

If you've ever experienced the frustration of trying to use a free-hand illustration tool to draw structure diagrams (CASE or flow diagrams, network or hierarchy charts, decision trees, etc.), you'll understand Diagram!'s immediate value: its ability to maintain structure and relationships, so you don't have to.

File linking

File linking allows you to use graphics practically as they're used conceptually: to represent other information--information stored in source code, spreadsheets or spec files. To set a link between your Diagram! drawing and any other such file or program, just drag and drop files into it. Diagram! automatically establishes a link.

To view the newly linked file, just double-click, and Diagram! starts up any other NeXTstep or UNIX application to open it.

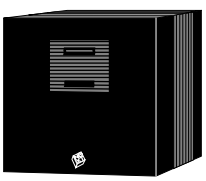
Create hierarchical diagrams by linking a series of Diagram! files together. Diagram! can even act as a coarse-grained hypertext tool.

Symbol palettes

Symbol palettes allow you to customize Diagram! by organizing frequently used symbols and images for easy retrieval. Collections of symbols can be quickly viewed and incorporated with a simple point and click.

Symbol palettes eliminate awkward drawing tool "modes"--to draw a palette shape, just point to it and "tear off" a copy into your drawing. No more guessing about your program's default styles.

“Sound adds another layer of depth to your work”



Future commercial Diagram! palette collections will include:

CASE and design
Electronics and engineering
Communications and networking
Medicine and dentistry
Desktop publishing
General business

Voice annotation

Voice annotation permits you to give a presentation once, and let others listen to it when they have the time. Just double-click the lips to hear sound recorded with the NeXT’s built-in microphone or from other sources.

Sound adds another layer of depth to your work, while saving you the time and effort of repetitious presentations. And your audience can easily send back verbal commentary and feedback.

Smart undo and undelete commands

Smart undo allows you to specify the objects to which “undo” should apply. Smart undelete even remembers to recreate connections to deleted objects--even if the objects to which the deleted object was originally attached have been moved or changed.

Remembering open files

By remembering open files, Diagram! can start up where you left off by opening all the palettes and documents open the last time you quit.

Checking this preference means you no longer have to go hunting through directories to find the last set of palettes and documents you were working on.

Standard drawing tool features

Of course, Diagram! offers you the features of a standard drawing tool, too, such as gridding, grouping, layering, zooming, and alignment.

Diagram! also supports color for fill, outline, and shadows, as well as offering several arrowheads and dash patterns.

OTHER INFORMATION

Of course, Diagram! is fully integrated with all standard NeXT software. You can copy from Diagram! and paste EPS or TIFF images into any other NeXT program.

Diagram!’s list price in the United States is $399. A steep introductory discount is available through NeXTConnection (call 1-800-800-NeXT), until March 1, 1990. Site licensing and discounts are also available.

Lighthouse Design’s University Reseller Program allows accredited academic institutions to distribute Diagram! to students on their own media for as little as $10, and to faculty/staff for $75. Have your school contact Lighthouse Design, Ltd., for more information.

Lighthouse Design products carry a guarantee of 100% satisfaction-- purchasers may return it for any reason.

A portion of Lighthouse Design’s profits is contributed to charity annually. We are an environmentally responsible company, and do not use shrink wrap in our packaging.

EXECUTIVE SUMMARY

Diagram! is an innovative drawing program for NeXT computers.

IT’S FAST - Customizable symbol palettes for fast retrieval and non-modal drawing - Complete set of on-line documentation, indexed for Digital Librarian

IT’S EFFICIENT - Rubberbanding straight and curved lines to eliminate drawing tedium - Automatic label maintenance and placement

IT PROVIDES A GRAPHICAL BACKBONE FOR ALL INFORMATION - Easy linking of Diagram! drawings any other file or application

OTHER - Voice annotation for electronic presentations and verbal feedback and commentary - Smart undo and undelete - Remembers open files to be reopened during next session - Full color support - Priced thousands of dollars below comparable products for other platforms (and for students, it couldn’t be cheaper: $10!).

“Start thinking about your drawing, and stop thinking about your drawing program.”

For more information, please contact...

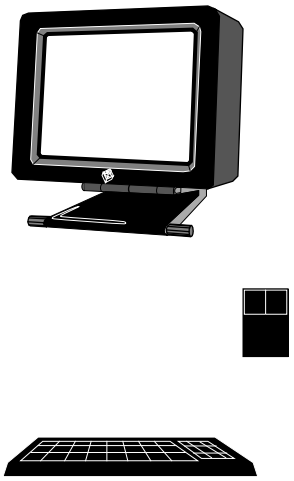
Lighthouse Design, Ltd.
Western Avenue
Chevy Chase, MD 20815-3212
301-913-2806

800-FOO-BAR9

diagram@lighthouse.com
!uunet!lighthouse!diagram

AT&T and Apple Announce a Specification for Accessing AppleTalk from UNIX

UNIFORUM, Dallas, Texas
January 22, 1991



AT&T Computer Systems and Apple Computer, Inc. today announced they have developed a specification that will standardize the access of AppleTalk networking protocols from AT&T's UNIX System V Release 4 operating system. Using the specification, software developers will be able to create AppleTalk networking application programs that run on UNIX systems and perform file sharing, printer sharing, electronic mail, or client-server applications.

Application programs written according to the new specification can easily be ported by a developer from one vendor's UNIX system to another vendor's UNIX system. According to Randy Battat, Apple's vice president of worldwide product marketing, "This AppleTalk specification is important to both Macintosh users, as well as users and vendors of UNIX systems. The specification will promote the development of AppleTalk application programs, which will allow Apple's customers to access information on UNIX systems in the same consistent manner in which they access information on servers in other environments."

AT&T has already implemented the AppleTalk application programming interfaces (APIs) described in the new specification in its StarGROUP Server for Macintosh Clients, which is part of AT&T's StarGROUP LAN Manager Server system. The StarGROUP Server for Macintosh Clients will begin shipping in February, 1991. Bill O'Shea, AT&T Computer Systems', vice president of systems marketing and development said, "We worked with Apple to develop these APIs because they are another essential step toward meeting the customer's need for interoperability. Our commitment to open, standards-based networked computing, gives the customer freedom on the desktop." O'Shea continued, "For example, by using our StarGROUP LAN Manager Server, a Macintosh user can transparently share files, peripherals and mail with other computer users running DOS, UNIX, or OS/2 operating systems -- without sacrificing their personal preference for Macintosh."

Apple will implement the AppleTalk APIs in their next release of the Portable AppleTalk source code, which Apple licenses to developers under the AppleTalk Licensing Program. Existing AppleTalk source code licensees will automatically receive the updated source code. The AppleTalk Licensing Program, established in June, 1990, is part of Apple's goal to extend AppleTalk to all major computing environments. The AppleTalk Licensing Program makes it easier for mainframe and minicomputer manufacturers, as well as networking companies, to offer a comprehensive set of network services that are compatible with the millions of Macintosh personal computers networked today.

Pacer Software, Inc. is an AppleTalk networking software developer that has completed ports of the AppleTalk protocols to several UNIX systems including those from Hewlett-Packard, Digital Equipment Corp., Sun Microsystems, and Santa Cruz Operations. Pacer's first release of the new UNIX System V Release 4 AppleTalk APIs will be offered on the NCR System 3000. Pacer also develops and licenses AppleShare-compatible file and print servers for UNIX systems. Garth Conboy, president of Pacer Software, stated, "We are committed to the new APIs because they make it so much easier for networking applications to be supported across various UNIX platforms. Our current System V Release 4 development incorporates these APIs and will result in common portable AppleTalk software that will be available from both Pacer and Apple."

AT&T and Apple will continue to work together on enhancing the APIs described in the specification while ensuring their consistency. The AppleTalk specification will be distributed by both companies beginning in second quarter of 1991.

Apple, AppleTalk and the Macintosh are registered trademarks of Apple Computer, Inc. UNIX is a registered trademark of UNIX System Laboratories Inc. OS/2 is a registered trademark of IBM. StarGROUP is a registered trademark of AT&T.

BANG Meeting

16-Jan-91 BANG Meeting Summary
Dick Benster, La Honda Software

>from: joeba@mindcraft.com (Joe Barello)

BANG director Rick Reynolds called our meeting to order at 7pm on this somber evening of the first day of the Gulf War. Approximately 45 people were in attendance for a highly informative gathering.

"Major efforts will continue in database access technology"

Prior to the presentations, the following information was offered:

1) A call for more volunteers was immediately made - we need more help in all aspects of running BANG. Please lend a hand!

2) Note the new BANG contact numbers: Rick Reynolds 415-512-5646 Joe Barello 415-327-BANG.

3) Note that the NeXT developers SIG which is co-sponsored by BMUG (Berkeley) and BANG is currently rotating meeting sites. This notice will be delivered too late for the Monday 21-Jan-91 meeting: please feel free to use the BANG contact numbers for information regarding the February meeting which should be held on our side of the Bay.

4) The next BANG meeting will be the third Wednesday of the month, (as usual), meaning 20-Feb-91 at Termin auditorium at Stanford.
5) As of this meeting there is a job opening at NeXT in the Software Release Control group - those interested should call NeXT and ask for Brad Devino (sp?).

6) There is an error in the 68040 upgrade procedure, whereby the boot block is not rewritten for the 68040. Please contact NeXT technical support ("disk -b /dev/rsd0a" is the magical incantation for writing the 68040 boot block).

Our first speaker was Ron Weissman, VP of Educational Markets at NeXT. Mr. Weissman, a former Associate Professor at Brown University, was one of the twenty-plus academic advisors that helped NeXT specify the original machine. He stressed that computing in higher education was THE reason for creating NeXT, and to this day the largest NeXT support group is for the education market and customers. NeXT strategy for the educational market is to avoid the give-aways used by competitors, and instead to focus on bringing fundamental, outstanding value-added software at very affordable prices.

High school education will be a new focus this year, with continued emphasis of expanding Mathematica in teaching, as well as using the digital voice capabilities of the machines in teaching English as a second language. Major efforts will continue in data base access technology with the intent of establishing NeXT machines as great information stations.

NeXT is very interested in user input on the importance of hypercard functionality and compatibility (i.e., do you want/require similar functionality but within NeXT software philosophies or simply strict compatibility - please feel free to contact Mr. Weissman at NeXT regarding your needs).

Finally, our speaker emphasized the need to change people's perspectives on software towards a "reusable palettes" approach. It is conceivable that in the future, customers may receive 50 - 75 objects that are carefully designed to cover the complete functionality of current "do-it-all" data processing and retrieval monoliths (like Microsoft Works), and much more. These palettes would have the enormous advantages of being much more modular, robust, and reusable in custom apps, as well as extensible via subclassing for further customization.

SLAC Physicist Paul Kuntz then gave one of the first user presentations to BANG. Mr. Kuntz explained that the goal of his group was to deliver custom statistical analysis tools, meaning quality GUIs on top of existing as well as new algorithms. The REASON package (Realtime Event AnalysiS On Next) was a superb demonstration of the power the machine brings to developers.

The SLAC group consisted of FORTRAN programs with no prior C, Object Oriented software, nor GUI experience. In less than five months his group was able to put together a premier package that allows users to graphically set up a statistical analysis via an "analysis chain," including the functions of pulling data of the disk, browsing the data, performing 2-D and 3-D histograms, and

looping through data. A major use is doing data fitting, in which starting parameters are giving and then the program converges parameters to a "statistical best fit" of data to function.

All the while the fitting is occurring, the screen shows the evolving data/plot relationship. In one demonstration, a 17,000 double floating-point data set was plotted in near real-time, with "rebinning" (the size of and offset of each histogram's bins) demonstrating the power of real-time analysis to determine what is statistically significant versus simply sampling artifact.

"clearly a new approach was needed"

One of the most important contributions of his group was to take a 20 year old FORTRAN-based statistical package which is considered THE standard in the physics community for data fitting, and by placing a quality NeXT GUI on top of the unchanged FORTRAN code, realize a new tool which exceeded the original author's expectations of both its behavior and power.

Mr. Kuntz's low-keyed and humorous approach in presenting his "buggy software by a bunch of academics" (this was quality stuff!) served as a powerful example of the extraordinary productivity and capabilities provided to NeXT developers for both breathing new life into ported code, as well as developing new approaches to old problems.

Our final speaker was Eric Bergerson of Objective Technologies, a Manhattan, NY company. Mr. Bergerson had worked on Suns on Wall Street, and understanding the value of writing reusable C code, proceeded to discover that the reusable code was never really as reusable as it should be. A closer look further revealed that much more time was being spent on the Sun View GUI code than on the required algorithms - clearly a new approach was needed.

Mr. Bergerson was immediately impressed early on by NeXT's Interface Builder's ability to greatly reduce the time spent on GUIs, and on its built-in ability to provide dynamic control objects a natural test environment (you could test the slider and output value without coding or compiling!). Another major plus was the Objective C approach which would allow the building of code "upon each others shoulders."

Mr. Bergerson's team has created a group of four different custom palettes which are sold separately. The palettes are groups of objects that are immediately accessible under the Interface Builder, and greatly reduce apps development time for text, math, and graphing. They consist of:

1) a browser palette
2) a formatted text palette
3) a graphing palette
4) a Mathematica front-end palette

Mr. Bergerson concentrated primarily on the "MathPalette," showing components including a communications channel to the math kernal (Mathematica), the MathView Postscript rendering window for graphing output, and the inspector used for accessing and editing the non-graphical attributes of objects.

A particularly simple but interested example of the MathPalette's power referred to the exercise NeXT developers see in class whereby a simple Fahrenheit-to/from-Celsius converter is created with the Interface Builder. Additional Objective C code is required to do the conversions. By using MathPalette and Mathematica, however, no code was needed to do this - the correct formula's were simply passed on to Mathematica, and the results displayed.

The GraphPalatte was the other product focused on. It lets the user do axis labeling, bar or standard plots, and "is probably good for 85% of all your graphing needs." Additionally, it may be subclassed to provide additional, custom plotting support.

Mr. Bergerson is committed to creating and selling highly reusable code. Existing palettes like MathPalete and GraphPalette, and apps like Mathematica can provide powerful tools as the foundation for new custom applications. Please note that students are currently being offered 50% discounts on the palettes, which list from about $499. to $1599. depending on the palette. Site licensing is also available for even lower costs.

“We . . . were able to put together a great listing “

200MB NeXTstation Now Shipping

The 200MB configuration is available for all systems including NeXTstation, NeXTstation Color and NeXTcube computers. This configuration ships with a preloaded version of Release 2.0. (Note that this configuration does not include a preloaded version of Release 2.0 Extended). This drive, with the preloaded software, provides the customer with over 125MB of available user space.

NeXTstation 340MB Orders Converted to 400MB

NeXT has announced that all customer orders of 340MB NeXTstation computers have been converted to ship with 400MB hard drives at no extra cost. Part number s and pricing will remain the same and resellers need not contact NeXT to verify t hat their orders have been converted. NeXTstations with 400MB hard drives will begin shipping in early February.

NeXT Hires Stanford University Bookstore Computer Manager

Richard Marquez, Manager of the Stanford University computer resale program for the past five years, has joined NeXT as a member of the Higher Education team. Richard will be working directly with Kathi Kaplan to champion the cause of the campus reseller from within NeXT. Richard will be focusing his attention on helping create the programs necessary for campus resellers to do business successfully with NeXT.

Q&A

1) How much room is left on the hard drive after the software is loaded?

A. Drive Release Approx. User Space
105 MB HDRelease 2.0 - 25 MB
200 MB HD Release 2.0 - 107 MB
400 MB HD Release 2.0 Extended - 180 MB
660 MB HD Release 2.0 Extended - 405 MB

2) Can you use a MegaPixel monochrome display with NeXTstation Color?

A. No.

3) How many floppies come with both 2.0 and 2.0 extended?

A: There will be approximately 15 and 20 floppies respectively for the Release 2.0 and Release 2.0 Extended backup products.

NeXT On Campus

Thanks to all who responded to my previous posting requesting information for the Winter 1991 issue of NeXT on Campus. We received more than 50 project descriptions and were able to put together a great listing of NeXT Academic Projects. The issue is now being printed and will be distributed nationwide during the first week of February. It will also be available at the public archives (from the sonata.cc.purdue.edu archive) shortly after publication.

For those of you new to the NeXT world, NeXT on Campus is a journal produced by the Higher Education Group at NeXT. The purpose of the publication is to highlight uses of NeXT computers in higher education (both teaching and research), to foster communication among educational users of NeXT technology, and to provide useful resources to educational users. The journal is published three times a year (January, April, and August) and is distributed to approximately 40,000 people and the public archive sites.

We are currently compiling the Academic Projects list for the Spring 1991 issue of NeXT on Campus. If you are working on a project you feel will be of interest to others in the NeXT community and would like it to be listed in NeXT on Campus, please fill out the attached information request form and return it via e-mail to: next_on_campus@next.com

An “academic project” is any work you are doing that involves a NeXT computer. Projects listed in previous issues of NeXT on Campus include:

The Genome Machine, a genetic information access and analysis tool. A video-tape training seminar. MidasPlus, an interactive molecular design and simulation system. A

"This is not possible in the MultiFinder environment"

music composition laboratory. Bessie, an interactive tutorial for teaching FM music synthesis. A series of modules to teach history and geography. DBA, a database management research project.

Here is a sample listing from the Fall90 issue:

Text Search Engine for the Humanities--A client application that allows quick and powerful searches through massive texts stored using the PAT text search engine

George Drapeau
Workstation Environments Specialist
Academic Information Resources
Stanford University
drapeau@jessica.stanford.edu

IMPORTANT: In order to include your project in the Spring issue, we need to receive your response by Friday, February 15.

If you do not have any project information to report but would like a free subscription to NeXT on Campus, fill out the request form and return it to: next_on_campus@next.com

Thank you,

Jeff Wishnie Technical Editor,
NeXT on Campus

(Use Following Format)

Information Request Form

Full Name:
Title:
Institution/Organization:
Department:
Address, phone, and e-mail address:
Which contact information you would like listed (i.e. e-mail only)?
Number of NeXT computers in department:
Project title and discipline:
Project status (i.e. completed, being coded, prototyped, proposed):
Briefly describe your project:

A Comparison of the Apple MacIntosh Environment and NeXT

MultiFinder vs. Multitasking

MultiFinder is built into Apple's standard system software as an extension of the original single-tasking Finder application. Therefore, multitasking was not designed into the original Apple OS. MultiFinder is better described as a "task switcher" than a true multitasker. In other words, MultiFinder allows several applications to be open, but only one can be active. It provides for an environment whereby the user can access one application, then switch to another application without being forced to close the original application, as they were in the original Finder environment.

However, once the user switches to the new application, the original application goes into a hold state, and is left inactive. This prevents the user from being able to allow an application to continue processing while they switch to another application to work with interactively. For example, if the user wishes to include data from their database into a report that is due that morning, they would typically start a query into the database in one application, and then start editing the report inside of their word processor while the database application is performing the query. This is not possible in the MultiFinder environment, since the word processor would take over leaving the other application in a hold state.

In an environment such as NeXT, with a UNIX operating system, true multitasking is the standard. Therefore, the above scenario would be possible, even to an extreme in that the user can have several programs running simultaneously, giving the effect that each program is running with its own processor. This is completely transparent to the end user, they don't have to think about how to do their work, they just perform the activities as comes natural.

The design of MultiFinder is based on the concept that each application thinks it is running in a typical Finder environment. Each application thinks it has access to all of the memory it needs, and to all of the resources of the computer. MultiFinder simply keeps track of who is loaded in which part of RAM, and informs applications of how much RAM is available when they are launched. Thus, each application runs in what they consider to be an exclusive mode, or non-sharing environment. This is true in UNIX as well, however, this is because UNIX provides a virtual

“This overall design dramatically increases the effective throughput of the system”

environment to each application, and handles the conflicts of resources between applications in the operating system. For example, in UNIX each application thinks it has an abundance of memory available to it, regardless of how much RAM is available on the system, or how many other applications are running. This is called “virtual memory”. This allows the end user to run several applications simultaneously, even though there is not enough RAM to hold these applications. Thus, the operating system takes care of paging to and from the disk, providing for the virtual memory environment, and making it appear as though there was enough RAM to hold everybody.

The MacIntosh does not support this virtual memory concept. Therefore, all of the applications that the user wants to run must fit within the limits of physical RAM on the machine. This prevents the user from being able to launch several programs at once. Depending on the configuration of the machine, the user can be limited to what combination of applications can run, because of the size requirements for the applications. It is quite common to launch an application in MultiFinder, and have the system come back and say there is not enough memory left for this application. The user must then choose another running application, and close it, in order to bring up the desired application. This is a scenario that is not run across in the NeXT environment.

Another fundamental distinction between the environments is in the way they switch from one task to another. Since there is physically one processor in both the Mac and the NeXT, the CPU must give out time slices to each process to keep the appearance of multitasking applications. There are two ways of determining how often each task gets a time slice, and how long each slice lasts: preemptive multitasking and cooperative (or non-preemptive) multitasking. In preemptive multitasking, which is utilized by UNIX, the operating system can shift the CPU resources to another application without the current application even knowing. Therefore, a single application cannot take control of the CPU, and halt everything else on the system.

With cooperative multitasking, each application determines for itself when it will give up the CPU to another program. The application must then save its environment before giving up the CPU. This means that a single application may never give up the

CPU once it starts running. This is the case with many of the applications in the MultiFinder environment, which take control of the CPU and stop all background processing. This also leaves the entire environment exposed, such that if the currently running application crashes, typically the whole system comes down. This is because there is not enough isolation between applications, as there is in the UNIX environment. Another advantage of preemptive multitasking is the ability for the operating system to adjust the time slices when an application is blocked, waiting for a particular device that is not currently accessible. Therefore, the OS is much more efficient in its use of the CPU. In fact, with the NeXT environment, there are special DMA (Direct Memory Access) channels included in the hardware architecture, which will allow a process to access things like memory, and the disk drives without the need for the CPU to be involved. This overall design dramatically increases the effective throughput of the system, adding to the efficiency of CPU utilization.

Interprocess Communication

Since the MacIntosh environment was not designed to be multitasking, it does not include facilities for interprocess communication. Therefore, the only way for two applications to communicate with one another would be through either shared memory (the “clipboard”) or a shared file (the “scrapbook”). This would require a formal understanding between the two applications, that must be designed and developed within each. The NeXT environment provides for standard interprocess communication facilities, which have been made into standard application objects (speaker/listener objects), which allow for simple and consistent access to inter-application communication. NeXT has even gone one step further with its “Services” facility, which is a common menu item object that allows for generic communication to outside applications from within an application,

“the environment is truly multitasking”

without the application having to know who it is talking to. This allows an application access to generic functions or services provided by other applications, that may be relevant to the current task at hand. All of this is made possible from NeXT’s utilization of the Mach kernel, which supports a common mechanism for interprocess communication (Mach ports), and the fact that NeXT is an object-oriented environment.

The most common way in the MacIntosh environment to transfer data between applications is the “clipboard”. NeXT has included this support in its environment as well, giving a consistent method for a user to transfer data from one application to the next, or even within an application. In the MacIntosh environment, the clipboard is based in RAM, and therefore is limited by the space available in RAM after the other applications have taken their chunk. In the NeXT environment, the pasteboard can grow larger, due to the virtual memory environment. In addition, only the NeXT environment can support the direct dragging of an object in one application, into another application running on the display. For example, a NeXT user can drag an image directly out of an email message and into their word processing document, to use in the report they are currently working on. They do this by selecting the graphic with the mouse, and dragging over to their word processing document, and releasing.

No menus needed, no “Copy” and “Paste” needed, and no switching between applications, since the environment is truly multitasking.

In the NeXT environment, applications coexist with each other and work together to help the user solve a problem. In the Apple MacIntosh environment, the applications exist independent of one another, and the user is responsible for tieing together the necessary functions to solve a problem.

Security

One deficiency of the Apple environment, is the security issue. The MacIntosh was designed to be a single-user system. This means that anyone who has access to the power switch, has access to all of the data that resides on the disk. There is no inherent security features. NeXT however, utilizes the multi-user environment of UNIX, allowing for different people to have different access to the files on the system. Depending on the needs of the environment, the access can be opened up or severely restricted to the users of the system.

File Sharing

Another advantage that NeXT has over the Apple MacIntosh, is the sharing of files between machines. NeXT utilizes standard facilities, like NFS, for file sharing. This allows file systems on each machine, to be accessible to any other machine on the network - if so desired. Each machine on the network can in effect be a file server.

It is this capability that allows the NeXT environment to be configured such that a user can log into any machine on the network, and see his/her own custom environment, as though they had logged into the system on their own desk. The only way to do this on the MacIntosh, is to replicate the environment on every machine, for every user.

Imaging Models

The core technology of all GUI’s is the imaging model - the language and paradigm used to put images on the screen and the printer. The NeXTstep window server does what no GUI has ever done before: unifies the imaging model for the screen and the printer. This is made possible by NeXT’s decision to utilize Display Postscript for the screen, and Postscript for the output language. In addition, NeXT worked in combination with Adobe to define and enhance the Display Postscript environment, in order to maintain compatibility with the evolving Postscript standard.

The Mac OS draws to a computer display with an entirely different set of instructions (QuickDraw) than is used to draw on the printed page (Postscript). This dichotomy causes inconsistency between what you see on the screen and what is seen on the printed page. Often times it will take several print jobs from a MacIntosh to get the printed page to look as desired. Apple has recently decided to try and correct this situation by converting to a unified imaging model as well. However, Apple’s approach is to standardize on their

"HyperCard applications are not truly stand alone applications"

own proprietary QuickDraw language. This moves Apple away from the industry standard Postscript output language. Therefore, they must continue to support Postscript in order to be compatible with the existing installed base of Apple laser printers, and the tremendous amount of Postscript based third party output devices in the industry. This means that applications must then be able to support the industry standard Postscript, as well as Apple's own QuickDraw.

With the NeXT unified imaging model, the developer only has to write one set of instructions to display the application's data, and that will take care of the display on the screen, as well as output to a hardcopy device. This reduces the amount of application code, and the time to develop the application. This also keeps the application consistent with the defacto industry standard output language.

Development Environment:

The standard application that people think of when it comes to doing development in the Apple MacIntosh environment, is HyperCard. HyperCard has been successful on the MacIntosh for a couple of reasons. First, it was bundled with every system.

Second, it introduced the concept of an authoring system where programming is an assembly process, rather than strictly code generation. However, once the user must go beyond the simplicity of a HyperCard application, the user is left on his/her own to develop from the ground up. HyperCard does not scale with the application, and is not extensible to allow for the creation of complex custom applications. HyperCard is limited to its basic elements of cards, stacks, buttons, fields, and backgrounds. The programming language, HyperScript, is very limited in its capabilities, and falls far short of a flexible programming language.

The NeXTstep environment includes the tools for the assembly of applications as well. However, these tools will work for simple assembly, as well as complex custom development. They work in conjunction with standard higher-level programming languages, such as Objective-C, C++, and ANSI C. The NeXTstep Interface Builder, which allows for the interactive assembly of an application, works in combination with the user's choice of programming languages, and is completely extensible to whatever degree of complexity is necessary for the application. The user is never left to start from the ground up to develop the application.

The other advantage the NeXTstep environment has over HyperCard, is that it is truly object-oriented. NeXT is committed to the object-oriented paradigm, and has designed the environment from the ground up to be consistent with this. The end result is a set of tools, and object kits, that directly support the object-oriented paradigm. It is this implementation that gives the user complete flexibility through the subclassing of existing objects, and complete inheritance from the existing toolkits.

This is not supported in HyperCard, because of its lack of a true object-orientation.

The user is relegated to the existing 5 entities provided by HyperCard.

Hypercard also lacks polymorphism (polymorphism: In object-oriented programming, the ability of different objects to respond, each in their own way, to the same message). This is the ability in the NeXT environment to have multiple objects, with the exact same method names implemented within each. Therefore, due to the run-time binding provided by Objective-C, an application can send a message to an ambiguous object, referencing it through a variable, that is bound at runtime to an actual or physical object instance. This gives the developer much more flexibility in the development process and structure of the application. The developer is not forced to know who or what they will be talking to, and calling out a specific function based on the appropriate object. This is one major difference between a true object-oriented environment, and utilizing a function library to access pseudo-objects.

HyperCard applications are not truly standalone applications. They are actually programs that run within the actual HyperCard application, and are typically interpreted. This contributes to Hypercard programs being extremely slow. It is not very realistic to do a sizeable project in Hypercard. It works well for small programs like an address book

“. . . you can easily get nailed by knowledgeable Macophiles”

etc., but anything more complicated and the performance (speed) becomes a major issue.

The NeXT environment is much more robust with its development environment. The developer has Interface Builder to interactively create the application. They have access to the Application Kit for standard objects, the Sound Kit for accessing the sound capabilities of the NeXT environment, and the Music Kit. Because of the object-oriented environment, and Interface Builder’s dynamic palette capability, they will have access to other custom object kits or palettes in the future, such as the Database Kit. The environment is fully customizable by the developer, and objects can continue to be reused across applications. The Interface Builder facility also includes a project management facility that is lacking in the MacIntosh environment.

In general, NeXT provides a complete development environment, that is well integrated with the system, as opposed to a simple application such as HyperCard.

The Macintosh employs a primitive development environment for complex programming when compared to the NeXT. Apple produces a “Macintosh Programmer’s Workshop” for the Mac development community. Unfortunately, it is an altogether new environment deviating from the GUI that is so familiar to Mac users. MPW does support C++, an object-oriented version of the C language, but since the OS code and ROM functions were originally written in assembly and Pascal language, much software “glue” is needed for interface between the developer’s code and the underlying OS. To combat this inconsistency, Apple has advocated a new environment know as MacApp. MacApp is described as both a stand-alone application shell and a set of reusable libraries upon which to base your program.

MacApp is written in a superset of Pascal called ObjectPascal. MacApp does help to rectify some of the difficulties found in developing for the Mac, but it still suffers from the foundational problems that the Mac OS was not developed using the object-oriented paradigm and that ObjectPascal is not a popular language with programmers in general.

NeXT v Apple Rebuttal

As a reformed Mac programmer, I found a number of inaccuracies in the summary. Just to name a few briefly:

1) Under MultiFinder, “switched out” applications can still run in the background, despite what the summary says. The main program can release some cycles to be used for these background apps.

2) Virtual memory has been supported by a third party for at least a year now, and Apple plans to make it a standard feature in the (still unreleased) System 7.0 (and you thought WE had delays with Motorola! see what rewriting the kernel in C++ can do for your “productivity”?! [cheap c++ flame, sorry])

3) MPW is a sophisticated Unix-like development environment, and third parties produce products like Think C which come bundled with nice graphic debuggers and nice class libraries (these sell for about $90 at Stanford). AUX gives the Mac many Unix benefits....

Without going deeper into this (this discussion could last days....) I merely want to point out that if you use the summary sheet as ammo for a “Mac-bashing” campaign, you can easily get nailed by knowledgeable Macophiles. The summary is a broad, impressionistic canvas that is nice to look at from far away, but certainly doesn’t have much support in the gory details, which is where I have found most of my CC/client discussions occurring. Caveat emptor.

Tom (I’m STILL working on the Icon summary) Affinito taff@spica.ucsc.edu :)

Date: Wed, 23 Jan 91 14:32:42 -0800
From: dennis@cs.washington.edu (Dennis Gentry)
To: taff@spica.ucsc.edu Cc: campus_consultants@cpac.washington.edu
In-Reply-To: Thomas Affinito’s message of Wed, 23 Jan 91 04:53:08 PST
< 9101231253.AA03157@spica.ucsc.edu>

Tom is definitely right that Macophiles will call you on some of the points in Conrad’s summary, but you don’t have to roll over and play dead when they do.

No one should take any of the following counter-arguments personally, I’m merely

providing them to help you to make sales when these objections come up. Be gentle if possible, since people *do* tend to take this stuff personally:

Date: Wed, 23 Jan 91 04:53:08 PST
From: Thomas Affinito <taff@spica.ucsc.edu>
To: campus_consultants@cpac.washington.edu

"not all virtual memory is created equal"

1) Under MultiFinder, "switched out" applications can still run in the background, despite what the summary says. The main program can release some cycles to be used for these background apps.

Even though multi-tasking usually works under MultiFinder, it's not certain since the currently running program might not release cycles because of the lack of *pre-emptive* scheduling. Pre-emptive scheduling makes programming much easier (no need to call "systemEvent()" thirty times a second), and greatly lowers the risk of the OS being crashed by a buggy app. As long as you never run buggy apps, MultiFinder works passably well. However, bugs in a single app can crash the whole machine, causing you to lose work from any other apps that are currently open. Mach will more robustly handle buggy apps, so only the offending app crashes (or just runs poorly!), rather than losing everything that's open.

2) Virtual memory has been supported by a third party for at least a year now, and Apple plans to make it a standard feature in the (still unreleased) System 7.0 (and you thought WE had delays with Motorola! see what rewriting the kernel in C++ can do for your "productivity"?! [cheap c++ flame, sorry])

The problem with virtual memory support on the Mac is that many of the existing Macs (even currently selling models) can't support it properly because they don't have the necessary hardware (with the 020 machines, you need the external MMU chip, with the 68000 machines, you're stuck). Also, not all virtual memory is created equal. The third party virtual memory that I know of is limited to 16Mb. Mach provides for as much virtual memory as you have free disk space on the swapfile's partition (subject to a 4Gb address space limitation, of course). From Apple, it's still vaporware.

3) MPW is a sophisticated Unix-like development environment, and third parties produce products like Think C which come bundled with nice graphic debuggers and nice class libraries (these sell for about $90 at Stanford). AUX gives the Mac many Unix benefits....

But it's not bundled with the machine, and it's only Unix-like. Everyone who buys a NeXT is licensed to use the software development tools under NeXT's state-of-the-art object-oriented system. It's true that A/UX brings Unix to the Mac, but it's a bag hanging off the side of the machine compared to Mach and the NeXT, where a real OS was built right in from the start.

Happy Selling, Dennis

Job Listing

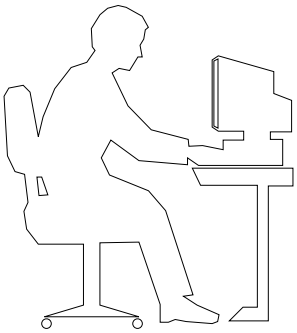
Dave Hieb--

I am looking for a system administrator/programmer to work on a net of NeXT machines (9 at present, more coming, including NeXT Dimension) and also on various remote Crays. the ad follows. If anyone in rmNUG is interested, they should respond to the ad, and/or they can e-mail me. The closing date to apply is ***February 15, 1991****. My phone number is (303)491-8474.Thanks.*

Computer Programmer

Colorado State University Atmospheric Science Department is seeking research coordinator/associate applications from qualified individuals to perform global modelling programming. Duties will include optimization and maintenance of very large atmospheric models running on remote supercomputers, installation and debugging of software including graphics packages on a network of NeXT workstations and system administration for this network. This work will be performed under faculty direction and as part of a research team including students and postdocs. Experience with Unix, Fortran and C programming is required. Experience with nfs and knowledge of the Postscript language desired. A B.S. degree in computer science or related field is also required. Application

deadline - February 15, 1991; position available February 18, 1991. Salary negotiable commensurate with experience. Submit resume to Dr. Stephen Cox, Professor and Head, Department of Atmospheric Science, Colorado State University, Fort Collins, CO 80523. Colorado State University is an equal opportunity/affirmative action employer and complies with all Federal and Colorado Laws, regulations, and executive orders regarding affirmative actions. EEO/AA offices located in Room 314, Student Services Building.



NeXTWorld Addresses:

NeXTWorld
501 Second Street
San Francisco, CA 94107
internet: nextwrld!ddinucci@uunet.uu.net
MCI Mail: NEXTWORLD
Compuserve: 76207,1326

Subscription Services
PO BOX 56429
Boulder, CO 80322-6429
1-800-775-6398

Back Issues ($8.00 each)
501 Second Street
San Francisco, CA 94107

NeXT Technical Alert 91-1 - FrameMaker

Summary

The version of FrameMaker shipped with Release 2.0 doesn't display some characters properly, and has a problem with FrameMath. Also, the Frame Update package shipped with Release 2.0 doesn't work properly.

Description

Users of FrameMaker under Release 2.0 may notice some of these problems:

1. Some characters may not display or print properly.

2. If a FrameMaker document is printed to a NeXT machine from a non-NeXT machine, some characters may not print properly.

3. The Pop-up lists on the panels used to manipulate mathematical equations do not work properly.

4. FrameUpdate makes FrameMaker versions 2.0 and 2.0a unusable. FrameUpdate has less severe effects on other versions of FrameMaker, but its use is not recommended.

5. Printing on 68040 machines is very slow.

Most of these problems are fixed in FrameMaker 2.0d. However, printing FrameMaker documents to a NeXT machine from a non-NeXT machine will still not work. This problem will be fixed in future FrameMaker releases of FrameMaker for each non-NeXT platform.

Workaround

Obtain the new release: FrameMaker 2.0d. Higher Education customers should obtain a copy from their NeXT Campus Consultants or Campus Support Center. All other customers can receive FrameMaker 2.0d by return mail from NeXT. Customers can get the optical version by sending a blank optical disk to NeXT, or the floppy version by sending their address with the request.

For More Information

All Higher Education customers should contact a Campus Support Center to obtain a free copy of FrameMaker 2.0d. For all other customers who need to take advantage of the optical disk swap program, mail an optical disk to:

FrameMaker 2.0d Upgrade
NeXT Telebusiness
900 Chesapeake Drive
Redwood City, 94063

Please be sure to include your return address to insure that your FrameMaker 2.0d is routed correctly. Customers that have floppy disk drives can mail their return address to the above address to receive FrameMaker 2.0d on floppies.

For further information call NeXT Telebusiness at 1-800-848-6398.

Member Profiles

Al Steele

Al attended the University of Colorado at Boulder where he obtained both a B. S. and Masters degree in Mathematics which he finished in 1973. At this point Al embarked on a

“it was love at first sight”

long stint in Graduate School which ended in 1987. Currently Al is working as an independent systems and engineering consultant for math intensive projects.

Professionally Al is interested in Numerical Methods, and Algebraic Number Theory. He uses his NeXT for Lisp programming and symbolic processing. Al was able to get his Cube from Businessland during their recent fire sale and he is very happy with the machine. He has also recently been working on a project to create a UUCP bridge between the Berkeley and System V UNIX operating systems.

Outside of his work Al enjoys mountaineering, music and music composition. He also dabbles in Classical Guitar performance. The members of rmNUG are pleased to have Al as a member. And to those of you who have not yet met Al, please take the time to introduce yourself. We all look forward to Al’s input in the group.

Demetrius Johnson

Demetrius Johnson, 17 years old and a Junior in High School has been programming since he was 9. Taught by his father on an Atari 400 in 1982 Demetrius didn’t take to games as much as to exploring what he could make the computer do. When his father got an Apple II Demetrius stood by his father’s side asking questions about code and from there he was off creating his own programs.

On the Apple he wrote a road map program that would connect home and destinations points. He also hooked up a robot toy he got at Christmas to the Apple II so you could see them robot movements on the screen. If he wasn’t watching Star Trek or Nova he was being a computer programer, computer cannibal (by 15 he had taken apart an Atari, an old IBM, Commodore and countless walkman radios and built electronic circuits from the Radio Shack series)

A few years later Demetrius heard about the Lisa and after seeing a demonstration he wanted to buy it. A few years later he talked his father into buying a Mac and began programming on the Mac in BASIC, Pascal, and ended up in love with C. Demetrius took on writing a program for a therapist who needn’t a special kind of record keeping and learned how to use a modem which linked him with like minded souls and more sources of information on AI (Artificial Intelligence)

Then it happened! He had only heard about the NeXT and when he happened on a demonstration of it, it was love at first sight. He would ride over to the Buffalo Chip to use this superior machine. It was, in fact, one of these visits that brought him to the CU Engineering School. Professor Rodgers, mistaking Demetrius for a University Student, asked him if he could write a program for the materials lab. At first hesitant but drawn to the lab, research and access to larger computers, Demetrius took on the ambitious task. This data acquisition program would retrieve data from Nicole 460 scope through a GPIB board. This project would make it possible to let researchers see data from their newly developed sensors. This program was written under a NASA funded grant.

A second connection at CU brought him into almost an apprenticeship under the computer science dept. that finally gave him access to a NeXT. For the following months Demetrius was known to disappear with no concern for food or human contact if not for anxious parents the Rip Van Winkle story would have a modern version.

These days find Demetrius trying to finish High School while at the same time taking Classes in Computer Science at CU. If not for his piano compositions, biking, and camping interests he would get lost in the hum of the hard drive. As you can see Demetrius is a fascinating and valuable participant in NeXT computing and in rmNUG as well. Please take the time to get to know Demetrius better.

Terry Tautz

January Meeting

The January meeting of rmNUG was held on the 16th at the Colorado School of Mines.

Business - Dave Hieb

The motion was made to not require the 5 primary board members to pay dues. That motion was carried by a hand vote of the attending general membership. Jacob Gore presented the current status of the efforts to

“thanks to everyone that participated”

spend the Golden Nugget award money on an electronic hub node for the group. An announcement was made about annual membership dues ($24) being due. Dave introduced our new NeXT Systems Engineer, Tim Carlin.

Election - Barb Dyker

The floor was opened for last minute nominations of which there were none. The candidates introduced themselves. Voting was done by write in ballots and were collected and counted by Aaron Gordon. People were requested to vote only if they had paid dues or intended to pay dues. The results are as follows:
President: Dave Hieb
Vice President: Karl Sierka
Treasurer: Jacob Gore
Administrator: Barb Dyker
Newsletter Editor: Dave Bowdish
At-large: Brad Green
At-large: Robert Gregory

The contested positions of Vice President and At-large were very close. Thanks to everyone that participated.

Presentation-Word Perfect

Representatives from WP gave a wonderful demonstration of their recent NeXT product which is already available via NeXT Connection. They also handed out M&M candies and WP mouse pads. Word Perfect has already proven their product on the PC, Macintosh and UNIX (X windows/sunview) platforms. Their NeXT product completely embraces the user interface NeXT users are accustomed to. Their features for reformatting page layouts were amazingly simple and quick.

From the Board

The executive committee met Feb 4th, 7pm, at Stratton Hall, School of Mines. All were in attendance (Dave Hieb, Karl Sierka, Jacob Gore, Dave Bowdish, Barb Dyker, Brad Green, Robert Gregory).

Newsletter: The membership list will no longer be included but rather dealt with as a separate item and distributed as needed. NeXT will pay for 20 copies of the newsletter each month to be used for enticing new members. We will be asking members with ftp or email access if they wish to waive printed copies in the interest of money and the environment. Dave Bowdish has received inquiries about non-local subscriptions to our newsletter, and he will investigate the feasibility of that as a break-even operation. Barb will post announcements monthly to netnews (co.general, comp.sys.next) about the availability of the newsletter via ftp. Dave Bowdish will draft newsletter submission guidelines and include them in the newsletter. Terry Tautz (NeXT) was volunteered to help coordinate upcoming articles.

Meeting Announcements: Meeting announcements will continue to be sent via email to all interested people even if they are not paid members because it is free. Regular mail announcements will always be sent to paid members. In the interest of fostering new members, regular mail announcements will be sent to non-paid interested people for a fixed period of time and discontinued if they do not join (period is to be determined). Meeting announcements will continue to be published in local papers: Rocky Mountain News, Colorado Daily, Boulder Business Report. Announcements will be posted to netnews co.general and comp.sys.next limited to Colorado distribution, because it’s free.

Monthly Executive Committee Meeting: The executive committee will meet regularly on the last Tuesday of each month at 7pm alternating between the CU campus in Boulder and the DU campus in Denver. The next executive meeting will be Feb 26, 7pm, DU campus. Meetings are open to all.

rmNUG Hub Node: Jacob reported on the status of negotiations with Colorado School of Mines for a joint project. The equipment needed barely fits our budget. There will be only one 9600 baud modem and one 2400 baud modem to trim costs. Dave Hieb, Barb and Jacob will meet with contacts at Mines before next general meeting to clarify agreement.

Miscellaneous: There are two email aliases setup: rmnug@cs.colorado.edu that goes to all paid members with email addresses, rmnug-exec@cs.colorado.edu that goes to the executive committee sans Bob. Dave Hieb will continue to coordinate the general meet-

“dedicated to fostering innovation”

ings while Karl will coordinate the software contribution program. Brad agreed to make reusable rmNUG signs to direct people through the mazes to the general meetings each month. Bob and Karl were delegated the task of completing the paperwork for incorporation. Long distance calling cards were ordered for the committee members in order to track any necessary calls

Barb Dyker

February rmNUG meeting.

The February rmNUG meeting will feature our own avid rmNUG member and Vice-President, Karl Sierka. Karl will be presenting GNU emacs and the GNU debugger for the NeXT. Karl has been a system programmer at NCAR for the last 5 years and between his research at work and home he has developed an intimate relationship with these GNU tools.

Emacs makes use of interprocess communication facilities available on the NeXT that allows it to create, manage and interact with sub-processes. In this presentation, Karl will be exploring the emacs key bindings and functions that are used to debug simple applications under NeXTStep using GNU emacs and GDB. If you have ever wanted to write a C program and step through it a line at a time using the debugger, this talk should get you started.

Karl explicitly told me that he will personally try to answer all of your questions about this topic, so try to bring some problems that interest you. Also, if he can’t answer them while at the meeting, he will be more than glad to try to answer them via email or over the telephone.

For reference purposes, here is Karl’s information on how to reach him:

Karl Sierka email - karl@labyrinth.com phone - (303) 443-5275 Labyrinth Computer Services, 646 Pearl Street, Boulder Colorado 80302 (-: Replies can have NeXT attachments in them.:-)

The February rmNUG meeting will be held Wednesday, February 13 at 7:00pm. It will be located in room 116 of the Business Administration building at the Denver University Campus.

Here are the directions to the meeting: Take I-25 to the University South exit and head straight south, 2 blocks down turn west on Asbury Avenue, 3 blocks to Race St. South. The building is a large red brick 4 story structure. Parking and doors are at the corner of Race and Asbury across from the tennis courts.

Dave Hieb

March Meeting

Doug Simons of Thoughtful Software will be giving a preview of HyperCube. This project, currently under development, is a NeXT application that is designed in the tradition of Hypercard for the Mac.

Dave Hieb

USENIX/Uniforum ‘91.

I recently attended the USENIX/UniForum ‘91 conference held in Dallas, Texas and thought a summary might be of interest.

USENIX, a not-for-profit UNIX and Advanced Computing Systems Professional Association, is dedicated to fostering innovation and communicating research and technological developments; sharing ideas and experience relevant to UNIX and UNIX-related systems and the C programming language; and providing a forum for the exercise of critical thought and airing of technical issues.

Founded in 1975, USENIX sponsors two annual conferences, a once-a-year technical exhibition and frequent workshops addressing special interest topics. The Association publishes proceedings of its meetings, a bimonthly newsletter; login:, Computing Systems, a refereed technical quarterly, and is expanding its publishing role with a book series on advanced computing systems. The Association also actively participates in and reports on the activities of various ANSI, IEEE and ISO standards efforts.

USENIX meets twice a year for a summer and winter conference in varied locations like; Summer 90 in Anaheim, Winter 91 in

"What better way to show off the Interface Builder can you think of?"

Dallas, Summer 91 in Nashville and Winter 92 in San Francisco. The conference is divided into a Tutorial section (Monday and Tuesday) and a Technical session (Wednesday thru Friday). The USENIX member fee for the Technical session is $225.00 ($275.00 for non-members) and each full day tutorial is $225.00. Half day Tutorials are $125.00 and two full day Tutorials cost $395.00.

The USENIX tutorials are taught by leading UNIX experts and directed to an audience of software professionals and technical managers. They provide and in-depth coverage of essential areas of UNIX technology. Each day (Monday and Tuesday) is composed of 12 different full day tutorials. Here is a listing of some of the tutorials offered this time:

Monday: Intro to TCP/IP Protocol Suite, Intro to C++, Programming the X Window System (Version 11), Intro to 4.3/4.4BSD Internals, Mach Overview, Intro to the GNU C Compiler (GCC), Intro to Object-Oriented Programming, Network Security.

Tuesday: UNIX Network Programming, Using C++ Effectively, Intro to the X Toolkit Intrinsics, Mach Virtual Memory Internals, Advanced Topics in Systems Administration, Programming in Perl, Parallel Programming and Scalable Software.

As you can see, these tutorials are addressing relevant issues that interest a wide body of people. Plus, where else can you get the chance to listen to original authors discuss their projects like Avie Tevanian from NeXT Computers Inc. as he presents an overview of Mach, or Richard M. Stallman from GNU who presents the Internals of GNU C. The tutorials can be quite rewarding considering the close interaction your afforded with these noted authorities. The tutorial materials are very comprehensive and serve as a wonderful tool for further study.

I attended the "Programming the X Window System" tutorial on Monday and the "Programming with the X Toolkit Intrinsics" tutorial on Tuesday. The Monday tutorial was intended for the first time X Window programmer and based on Xlib, the low-level C primitives that implement the X protocol. Our instructor was Oliver Jones, Software Developer and X Window expert for Saber Software Inc. The Tuesday tutorial was taught by Paul Kimball of DEC and was the next logical instructional step after the Xlib programming class on Monday. The X Toolkit Intrinsics are a set of high level user interface abstractions called widgets that a X Window programmer can use instead of having to hand-craft "objects" in Xlib. Some classic examples of widgets are scroll bars, button boxes, menus, display areas, etc. The class also spent a lot of time covering other basics such as Resource Management, Event/Translation Management and Window Management.

Together, these two days of tutorials served as a great introduction to programming in the X Window System. In my opinion, I think NeXT should check into offering "Developer Camp" tutorials at Usenix. What better way to show off the Interface builder can you think of?

The USENIX technical programs consist of research papers presented in groups of 3 related 30 minute talks. Each talk ends in an open microphone question/answer period where members of the audience are encouraged to ask questions. Some of the more interesting topics were: File System Performance, Kernels, Insecurity and Distributed Processing. Interspersed with the technical programs are a series of presentations called "Invited Talks". These special programs are usually presented in a panel type discussion which also have open microphones. Some of the interesting topics presented at the "Invited Talks" were: UNIX Security Today and Tomorrow, Systems Administration, Debugging X and X Toolkits. These are actually some of the most interesting sessions due to the extremely high level of interaction between the audience (and the panel).

In case you can't find enough to do in the evening, USENIX plans ahead for you by scheduling the Birds Of a Feather Sessions (better known as the BOF's). These sessions are a REALLY informal time to sit around and let a project leader or company representative tell you about the current status of their project and also gives you a wonderful opportunity to ask questions. Here are some of the BOF's that I found interesting: The GNU

“I strongly recommend you try some out”

Project, Project Athena, UUNET and UNIX Standards. The GNU BOF was presented by Len Tower of the GNU project. Len is a senior programmer and long time member of the GNU team. The GNU project is composed of Richard M. Stallman, founder and president of GNU, Len and a host of other paid/volunteer workers and is actually parented by the Free Software Foundation (FSF). The FSF is dedicated to eliminating restrictions on copying, redistribution, understanding, and modification of computer programs. FSF does this by promoting the development and use of free software in all areas of computer use. Specifically, FSF is putting together a complete integrated software system named “GNU” (GNU’s Not Unix) that will be upwardly compatible with Unix. Some large parts of this system are already working, and FSF is distributing them now.

The FSF is also parent to the League for Programming Freedom, a grass-roots organization of professors, students, businessmen, programmers and users dedicated to bring back the freedom to write programs. The LPF works to abolish the new monopolies by publishing articles, talking with public officials, boycotting egregious offenders, and possibly in the future by intervening in court cases. If your interested in reading more about the LPF, see the “User Interface Copyright Controversy” article in the November issue of the Rocky Mountain NeXT Users’ Group Journal available via anonymous ftp to alumni.-Colorado.EDU:~ftp/pub/rmNUG.

Here is some of the GNU software that is available from GNU: emacs, the editor; gdb, the debugger; bison, a Yacc replacement; GCC, a c compiler; GAS, an assembler; make, a BSD System V make replacement; BASH, an extensible shell; gawk, an awk replacement; RCS and CVS, source code control systems; ghostscript, a postscript replacement; groff, a troff and related programs replacement; and many more valuable pieces of software. If you have not used GNU software before, I strongly recommend you try some out. The code is well written, usually highly optimized which allows it to run faster/better than the code it replaces and since you get the source, you can see what is being done on the inside.

For more information on GNU, FSF or LPF, you can send email to ‘gnu@prep.ai.mit.edu’ in discretion or pick up one of the pink GNU bulletins that will be at the rmNUG meetings.

****

Copyright 1991 Free Software Foundation, Inc. Permission is granted to anyone to make or distribute verbatim copies of the GNU/FSF portion of the document as received, in any medium, provided that the copyright notice and permission notice are preserved, and that the distributor grants the recipient permission for further redistribution as permitted by this notice.

****

One of the really nice things about Usenix is how the vendors like Sun, IBM and others provide hospitality suites for members to attend in the evenings. They rent a large hotel room, cater good food, and bring in lots of their hardware to play with. IBM did a real bang-up job this year and provided at least 2 of my meals during that week (thanks IBM).

USENIX, in association with many hardware vendors, provides a terminal room where you can login to you home systems from one of 50 some graphics X terminals that are connected via a T1 link to the Internet. This also gives you a good time to try out some of the new X terminals from NCD and GraphOn, among others. The terminal room also provides facilities to copy GNU software to tapes.

The next USENIX is the summer 91 conference held at the Opryland Hotel in Nashville, Tennessee. The focus of the conference will be “Multimedia - for now and the future”. Here are some of the Multimedia presentations that will be featured at the conference: flight and other simulations, interactive video instructional systems, multimedia documents and desktop metaphors, hypermedia authoring systems, multimedia conferencing systems, integrated music systems, systems employing gestural interfaces.

The UniForum conference is completely separate and is typically held at the same time (and in a very close proximity) to the USENIX conference. This year UniForum ‘91 was held in the InfoMart in downtown Dallas and had 275 national and international UNIX System vendors presenting their newest prod-

"Here are some of the products I found interesting and worth mentioning"

ucts and services. This type of conference provides the attendee with a wonderful chance to actually talk with the vendors and get their hands on the new products, as well as pick up on all the freebies and literature you can carry.

I was very happy to see that NeXT finally committed themselves to having a main floor booth this year vs. only having a hospitality suite at the summer '90 Usenix. Their booth was well manned, had lots of literature and even had developer representatives demoing products from companies like: Lotus, Adamation, Knowledge Transfer, etc. As I was walking around looking at the booths of some of the large NeXT developers, I was very disappointed to see that Lotus was the only vendor that actually had a NeXT in their own booth. I asked both Ashton Tate and WordPerfect why they didn't have NeXT's and both camps seemed confused by the question and muttered something about this being a "UNIX" conference. If you can figure that one out, please clue me in.

Here are some of the products I found interesting and worth mentioning:

Pencom Software was displaying their port of the X Windows program to the NeXT. See my review of that product in the "From the President" section of this issue of the rmNUG Newsletter.

I talked with the systems people from Saber and they relayed to me their interest in porting their popular Saber-C programming environment to the NeXT. Saber-C is an integrated set of tools for programming, debugging, testing and maintaining code. Saber-C features a C interpreter, static and runtime error detection, incremental linker/loader, graphical browser, source-level debugger and interactive workspace. This might be a good systems program for the NeXT.

Visix, the company that makes the "looking glass" graphical user interface for UNIX systems, eluded to the fact that they would be introducing an office/desktop product for the NeXT.

Morning Star Technologies was sporting their VersaLink SCSI Communications Interface on a NeXT system. VersaLink supports Morning Star's X.25 and PPP protocols in addition to being able to operate in HDLC, SDLC and bisynchronous modes. This single device operates as an asynchronous serial port expansion device or high speed (T1) synchronous communications link. The product appears to the system as a SCSI disk and communicates with the operating system through the standard disk driver. Each communication line can be configured on an individual basis. VersaLink comes in both an internal and external packaging. I have since been in contact with the systems people at Morning Star and they have indicated that they would let us "try out" a VersaLink. This product would be perfect to provide our Internet access project with more dialin modem lines (if we could afford the bill for the phone lines).

Telebit was displaying their latest networking product; the NetBlazer. The NetBlazer is basically a smart ethernet to serial and serial to ethernet routing device which implements wide area networking over the public switched telephone network. When IP packets destined for a system on a remote network arrive at the NetBlazer, the NetBlazer establishes a dial-up connection with the remote network and then forwards authenticated packets. When the traffic dies down on the link, the NetBlazer breaks the connection to save telephone charges. The NetBlazer also works the other way such that users can call in over the serial line and establish TCP/IP connections (SLIP or PPP).

For more information on USENIX or upcoming events, call or write:

USENIX Conference Office
22672 Lambert St., Suite 613
El Toro, CA 92630
phone: (714) 588-8649
email: judy@usenix.org

Dave Hieb

Denver Rocky Mountain Chapter of the ACM February Meeting Announcement

Dr. Don Gotterbarn
Computer Ethics and Technology

Now at East Tennessee State University, Dr. Gotterbarn was educated at the University of Rochester and has taught at University of

“project is finally beginning to materialize”

Southern California, Dickenson College and Wichita State University. Among the software project s he has been responsible for were a nationwide videotex system and several database systems for the US Navy and the Saudi Arabian Navy. In addition he is the developer of the ethics module for the Software Engineering Institute (SEI).

Don’s topic will be Computer Ethics and Technology. He will focus on computer ethics as it relates to the day-to-day activities of practicing computer professionals. The emphasis is on real world moral and legal issues for the software engineer and how to resolve them.

Thursday, February 14, 1991
Armadillo Restaurant
15th and Market, Denver
5:30 Social Hour (cash bar)
6:00 Dinner
7:00 Meeting and Speaker

ACM meetings are always open and free to the public.Dinner, on the other ha nd, is another matter - its not free! To make reservations for dinner or for mo re information about ACM or the meeting, please call 696-6863 and leave your name, number of people attending (for dinner), and a telephone number we can cal l for confirmation. Please make reservations no later than Tuesday, Feb. 12, 1991

rmNUG Connectivity Project

Our electronic connectivity project is finally beginning to materialize. The Computing Center at the Colorado School of Mines will share one of their 68040 NeXTcubes with us, and that machine will serve as our connectivity host. We have started purchasing the equipment for it already, and expect to have all the pieces (hardware and software) together by the end of February. Hopefully, by the March general meeting, we will be able to offer specific instructions on how to connect through that machine.

Jacob Gore <jacob@gore.com>

rmNUG’s Software Contribution Program.

Here is the list of the software that rmNUG has received as of this date:

TextArt, Stone Design.
Compilation Disk, Lighthouse Design.
TopDraw, Media Logic.
DisplayTalk 1.0, Adobe.

Here is the list of the software that rmNUG has been promised:
Diagram!, Lighthouse Design.
Adobe Plus Pack, Adobe

Here is the current wish list for the rmNUG software contribution program:

BugByte 1.0 Absoft
Fortran 77
FrameMaker 2.0
Wingz 1.1
PaperSight
Contact! 1.0
Communicae
Click Art
Create!
DataPhile

FTP Access For The rmNUG Newsletter.

Issues of the rmNUG Newsletter are now available via anonymous ftp from the following sites:

cs.ubc.ca:~ftp/next/rmNUG.
alumni.colorado.edu:~ftp/pub/rmNUG.
sonata.cc.purdue.edu:~ftp/pub/next/Newsletters/rmNUG.

From the President.

During my recent trip to Dallas Texas for the Usenix/Uniforum conference, I had the privilege of attending the January Dallas Area NeXT Users’ Group (DaNG) meeting. Their special feature that night was presented by the NeXT developers at Pencom. Pencom’s product is called co-Xist and is the X11 Release 4 (X11R4) windowing system from MIT that coexists with the NeXT window server. co-Xist contains the MIT X11R4 server, clients and the Motif window manager.

“hopefully the speed will get better”

Kari Karhi, the Director of Engineering for Pencom and the main developer of co-Xist, gave the presentation and answered questions from the group. His development cycle was very short and considering he was new to both X-window programming and the NeXT, has come a long way. co-Xist is still in the alpha test stage and should be ready for shipping a complete package early this spring (March?).

I was able to use co-Xist in the NeXT booth at Uniforum on a 16 Mbyte 68040 NeXTstation and was unfortunately very disappointed by the lack of speed. Either I expected too much from the alpha product or am too spoiled from the speed of the 80Mbyte, 3 Sparc CPU machine that I run X windows on at work. Kari told us that hopefully the speed will get better as he optimizes the software and cited that most of the speed problems were due to the fact that a co-Xist X-window client has to image in a 2 phase process. Normally a X-window client asks the server for a window and basically draws directly into the window. Due to the fundamental windowing system differences in X-windows and NeXTstep, co-Xist X-window clients first have to draw into a (memory) pixmap and then have the server convert that pixmap into a NeXTstep data structure and finally display the pseudo window.

I had a long visit with Kari after the meeting and stressed my personal convictions about X-windows and how important I felt a good and fast port of X-windows is to the NeXT. We discussed NeXT’s initial lack of interest in the Pencom X-windows port and yet how the lack of X-windows compatibility has hindered too many potential sales.

DaNG has a regular meeting place in the Dallas “Tech Center” which is provided to them by one of the DaNG members that works in the building. Their meeting place is rather plush and is equipped with microphones, projection equipment and even an Extron video board. I arrived at the meeting a little before 7:00pm and immediately partook in the food and drinks that were available. DaNG is using their Golden Nugget Award for food and drinks and other fancies at the meetings.

Dirk Hardy is DaNG’s president and is also the president of Knowledge Transfer Intl., a NeXT development house that has produced the SimuLEARN package (page 202 of the Fall Software and Peripherals guide). Chris Bradley, the program director, is an energetic NeXT developer that led out in the meeting. Michael Green is DaNG’s NeXT Campus Consultant and is a computer science student of the University of North Texas.

DaNG has a mailing list of ~300 names and has regular attendance of ~50 NeXT enthusiasts. Interestingly enough, a very large percentage (much higher than that of rmNUG) of the group were programmers and NeXT developers. DaNG has just started to produce a Newsletter and was happy to get a copy of the rmNUG Newsletter to see what we are doing in the area. They even allowed me to address the group, thank them for the opportunity of being with them and invite them to meet with rmNUG if ever in our area.

davehieb...

rmNUG Membership

The monthly meetings of the Rocky Mountain NeXT Users’ Group are free and open to the public. However, we ask people to join the group by paying a nominal membership fee if they plan to frequent the meetings and take advantage of the newsletter and other group benefits. rmNUG is a non-profit organization. The membership fee is designed to defray the costs of providing services to its members in promoting the NeXT within the community. rmNUG currently has 35 members (including elected members) and over 130 NeXT enthusiasts on their mailing list.

Membership in rmNUG is $24 per calendar year. New members pay on a pro-rata basis for the rest of the current year. To become a member, see me to fill out a short information form and pay dues. Dues may also be sent to: rmNUG, 4521 Wellington Rd., Boulder, CO 80301. Be sure to include your name, address, phone numbers, and email address if you have one already. Benefits of membership include:

- Monthly meetings with interesting and motivating special features - Conversation and feedback from other enthusiasts like yourself - Evaluation of the software/hardware in our

"Thanks for voicing your opinions"

contribution program - A monthly newsletter that helps keep you up to date at the local and national level - Access to the rmNUG electronic node to take advantage of electronic mail, USENET news, and network software and information archives

Barb Dyker

NeXT Directory

The Boston Computer Society NeXT User's Group proposes to create a nationwide (or wider?) directory of personal, technical contacts for the NeXT computer. The idea is that if you have a question a person local to you could answer, the directory would help you find assistance quickly.

Quite often these kinds of lists, once produced, have been free, if electronic, or at very low cost, if printed. For a long time, members of the BCS have donated some of their time to answering questions.

Before we begin this project, what do you feel? Would such a directory be useful to you? What kind of information should be included? How can we protect volunteers from getting too many requests? Do you have concerns about privacy, propriety, ethics, cost, completeness or timeliness?

Opinions RSVP to geoff@camex.com or to comp.sys.next.

Geoffrey S. Knauth

UseNet Highlights!

Lighthouse Bargain

You are absolutely right. In fact, the old way of doing things (as you note above) defeated the best of our intentions: giving higher education easy access to our software. In conversations with students, bookstores, and academic computing facilities, we quickly learned that our policy wasn't as grand as we'd envisioned, and that it should change. So we changed it.

As of the second issue of NeXTWORLD (~March 15, 1991), students or faculty/staff may call NeXTConnection (800-800-NeXT) and purchase Diagram! for $25 (twenty-five dollars) and $75 (seventy-five dollars) respectively. What's the hook? Two hooks.

Hook 1: no packaging. Academic buyers get the full commercial software distribution (list $399), but you won't get the paper manual, or our nifty box and wrapper. Of course, all documentation is on-line (indexed for digital librarian), and the disk comes in a snazzy mailer.

Hook 2: keep in touch. If you like the software, we'd like you to tell us why. And if you don't like the software, we'd like to know why, too, so we can fix it. Diagram! has a built-in feedback feature (a "Make Suggestion..." menu item), so it should be easy.

We'd also encourage you to tell a friend. Lighthouse Design started developing for the NeXT when the higher ed market was the *only* market around. They hold a special place in our corporate vision (despite their bookstores :-), and if we're to continue offering discounts, we need your help in finding customers.

Thanks for voicing your opinions. It makes a huge difference.

Feel free to post this to the net. Feel encouraged, even.

Jonathan Schwartz
Vice President, Sales and Marketing
Lighthouse Design, Ltd.

Understanding Fax

There are two different sorts of standards in the fax world (relating to these issues): how the computer communicates with the modem, and how fax machines (including modems) communicate with each other.

"Class 2" describes a pending standard for how a computer communicates with a fax modem. It is to fax modems as the Hayes AT command set pseudo- standard is to data modems. Note that this standard is still in the balloting-updating-balloting stage.

"Group III" describes how one fax machine communicates with another, over the phone lines. It is to fax modes as V.32 (such as it is) is to data modems.

The software NeXT ships in 2.0 includes support for Class 2 fax modems. If a modem does not support Class 2 a custom modem driver would have to be used.

"I packaged each font into a standard format"

Note that there's a small bug in 2.0: a symbolic link is missing for the file Class2_Fax_Modem_Driver in /usr/lib/NextPrinter. The symptom of this: a known-compliant Class 2 modem, installed and configured correctly in PrintManager as a Class 2 Fax Modem, does not work right. The simple workaround: indicate in PrintManager that the modem is an InterFax Fax Modem. The simple fix: create the link; it should reference Interfax_Fax_Modem_Driver, also in the /usr/lib/NextPrinter directory.

Regarding old InterFax fax modems (for a Mac, e.g.): according to Abaton, the difference between the 24/96 and the 24/96 NX is more than just a PROM change. They will not be providing an upgrade from the 24/96 to a 24/96NX. They will, though, provide a 30% discount for a 24/96 NX to existing 24/96 Customers.

For more information, contact your Abaton dealer or Abaton.

Alan M. Marcum
NeXT Technical Support
amm@NeXT.COM

Fonts

Over the past two weeks, I've looked at 111 fonts from um-mts.cc.umich.edu and 5 from sumex-aim.stanford.edu. Of these 116 fonts, I rejected all but 52. (Most that I rejected did not have AFM files.)

I packaged each font into a standard format and divided them into two groups: those usable at no cost (32 total) and those that are shareware (20 total).

The included Makefile should work for installing the fonts under NeXT Software Release 1.0 or 2.0. (I am running 2.0, hence I have been unable to validate performance under 1.0.)

There may be more usable fonts at Michigan; I only checked those that the index seemed to indicate were laser or PostScript fonts. (If you'd like a list of what fonts I checked send me a note.)

Available from:
nova.cc.purdue.edu
sonata.cc.purdue.edu

Directory:
pub/next/misc

Files:
Fonts-1.0-free.tar.Z
Fonts-2.0-sw.tar.Z

Oregon (cs.orst.edu) is currently have problems with disks filling up. As soon as space is freed up, I should be able to get the fonts over there too.

Each file unpacks into it's own directory. Within each directory is a ReadMe.rtf and a Makefile. See the ReadMe.rtf for more font descriptions and installation instructions. (You may also find comments in the Makefile of interest.)

Good luck and I hope you find these useful. -dcb

Doug Brenner
<dbrenner@umaxc.weeg.uiowa.edu>

SIMMS And Hard Drive

A List of SIMM sellers with prices for 1 Mb and 4 Mb SIMMS with both toll free numbers and the usual kind. These are all U.S. sources. PLUS a list of hard drive sources with prices. Note these prices are as of January 31, 1991. Call for prices with a ?

Sources and prices for 1 Mb simms and 4Mb simms (80ns all low profile)

PSI
2005 Hamilton Ave., #220
San Jose, CA 95125

Ph. 408-559-8544
Ph. 800-622-1722

They take PO's

Price:
1 Mb $?
4 Mb $?

Lifetime Guarantee

Third Wave Computing, Inc.
1826-B Kramer Lane
Austin, TX 78758

They take PO's

Ph. 800-284-0486
Ph. 512-832-8282

Price: 1 Mb $? 4 Mb $?

Memory International

Ph. 714-588-0488

Lifetime warranty.

Price:
1 Mb $39.95
4 Mb $185.00
16 Mb $999.00

They take P.O.'s

Turbo Technologies, Inc.

Ph. 800-542-7466
Ph. 516-877-7777

Lifetime warranty

They take PO's

Price:
1 Mb $?
4 Mb ?

South Coast Electronics
10920 Wilshire Blvd.Suite 110
LA,CA 90024

Ph. 800-289-8801
Ph. 213-208-3260

They take PO's

Lifetime warranty.

Price:
1 Mb $?
4 Mb $?

Newer Technology
1117 S. Rock Rd. Suite 4
Wichita, KS 67207

Ph. 800-678-3726
Ph. 316-685-4904

Price:
1 Mb $?
4 Mb $?

Micro Electronic Technologies, Inc.
35 South St.
Hopkinton, MA 01748

They take POs

Ph. 800-766-SIMM
Ph. 508-435-9057

Price:
1 Mb $41.00
4 Mb $179.00

Peripheral Outlet
314 S. Broadway
Ada, OK 74820

Lifetime guarantee

they take PO's

Ph. 800-332-6581
Ph. 405-332-6581

Price:
1 Mb $39.00
4 Mb $192.00

ETC
5426 Beaumont Center Blvd. Suite 340
Tampa, FL 33634

They take PO's

Ph. 800-882-2863
Ph. 813-884-2863

Price:
1 Mb $?
4 Mb $?

Shecom Computers
22755 Savi Ranch Pkwy G
Yorba Linda, CA 92686

They take PO's

Ph. 800-366-4433
Ph. 714-637-4800

3 year warranty

Price:
1 Mb $43.00
4 Mb $250.00

Chip Merchant
9285 Chesapeake Dr. Suite L
San Diego, CA 92123

5 year guarantee

Ph. 619-268-4774

no credit cards

Price:
1 Mb $39.00
4 Mb $185.00

Memory Plus
P.O. BOX 367
Milford, Mass 01757

They take PO's

Ph. 800-388-PLUS
Ph. 508-366-2240

Price:
1 Mb $40.00
4 Mb $195.00

Technology Works
4030 Braker Lane West
Austin, TX 78759

They take PO's

Ph. 800-688-7466
Ph. 512-794-8533

Price:
1Mb $?
4 Mb $?

Stratum Technologies Inc.
12191 Technology Blvd.
Austin, TX 78727

Lifetime warranty

30 day trial period

Ph. 512-258-3570
Ph. 800-533-1744

They take PO's

Price:
1Mb $?
4 Mb $?

Impediment

Ph.617-837-8877

warranty 6 years

Price: ?

I think they take PO's

ClearPoint Computers

Ph. 800-253-2778

Price:

Associated Technologies Marketing

Ph. 214-248-0700

Mactel Technology Corp.
3007 North Lamar
Austin, TX 78705, USA

Ph. 1-800-950-8411
512-451-2600
U.K. 0800-98-1428

Price:
1 MB $43.00 (1x8)
4 MB $209.00 (4x8)

Some companies will let you trade in 1 MB SIMMS for a lower price on 4 MB SIMMS

Sources for disk drives

Prices are for INTERNAL drives, add $100 to $200 for EXTERNAL drives generally. Sizes in MB are unformatted sizes unless otherwise stated. Formatting can use about 15% of the drive space.

The CubeRoute

Ph. 1-800-CUBE-RTE

Guaranteed

660 MB WREN VI $1,995 (formatted)

1.2 GB WREN VII $3,295 (formatted)

Alliance Peripheral Systems

Ph. 1-800-233-7550
Ph. 816-478-8300

WREN VI $2,199
WREN VII $2,999
WREN (Half-height 502 MB) $1,899
WREN V (702 MB) $1,849
WREN (350 MB) $1299
QUANTUM 40 MB $309
QUANTUM 105 MB $519
QUANTUM 210 MB $959
MAXTOR drives available at similar prices

MacTel Technology Corp.

Ph. 1-800-950-8411
Ph. 512-451-2600

WREN VII $2,999
WREN VI $1,819
WREN Runner $2,900
(II)[650MB,10.5ms]

CMS Peripherals

Ph. 1-800-626-MEGS
Ph. 407-482-3821

WREN VII $2,995
WREN VI $2,195
WREN (300 MB) $1,489 (formatted)

ClubMac

Ph. 1-800-clubmac
Ph. 714-768-1490

QUANTUM (210 MB) $885
WREN VII $2,799
WREN VI $2,039
MAXTOR (345 MB) $1,359
MAXTOR (645 MB) $1,979

Aurora Industries, Inc.

1-800-426-1591
203-624-9018

650 MB drive $1,799
[these are Wrens I think]

400 MB drive $1,499

1 yr. warranty

Manufacturer's warranty on Maxtor drives is 2 years Seagate (WREN) is one year. This usually takes the form of a "disk swap". You will have to construct your own disktab for these disks, unless you are running 2.0. I make no warranty concerning these companies, their products or services. Other stan-

dard disclaimers apply. Don't take my word for it. Call for the current price.

Enjoy!

Prof. William V. Smith
smithw@hamblin.math.byu.edu
uunet!hamblin.math.byu.edu!smithw

"Next month we will be publishing under a new name"

From The Editor

This past month has been one filled with debate, discussion, and promising signs. The best news of the month comes from the February 1st issue of the *Wall Street Journal*. In that article Steve Jobs confirms that NeXT's monthly sales projects to an annualized gross sales for the year of over $100 million! An analyst with International Data Corp. states in the same article that they project NeXT to sell 36,000 to 44,000 units *this year*.

Also, now that NeXT and Businessland has mutually agreed to end their exclusive sales agreement, NeXT has aggressively been adding new resellers.

There has been much debate over the new keyboards. In order to make the keyboards fit ISO standards for sales to Europe and Asia, they made the keyboard less convenient for programmers. NeXT is currently re-examining it's new keyboard policy.

NeXTWORLD was finally released, giving NeXT national exposure at the newsstands. The premiere issue received mixed reviews from NeXT enthusiasts, but most are waiting to see what the second issue has in store for it's readers.

A special congratulations goes to Conrad Geiger at NeXT whose wife recently gave birth to a boy - Colin Davis Geiger. Much of what you have read in the rmNUG Newsletters was sent to us by Conrad.

Finally, as you may have noticed, the rmNUG newsletter has outgrown it's name. At 34 pages and growing, next month we will be published under a new name. If you have any name suggestions, you can email them to me.

David Bowdish

NeXT Benchmarks

These benchmarks were performed at ISI labs.

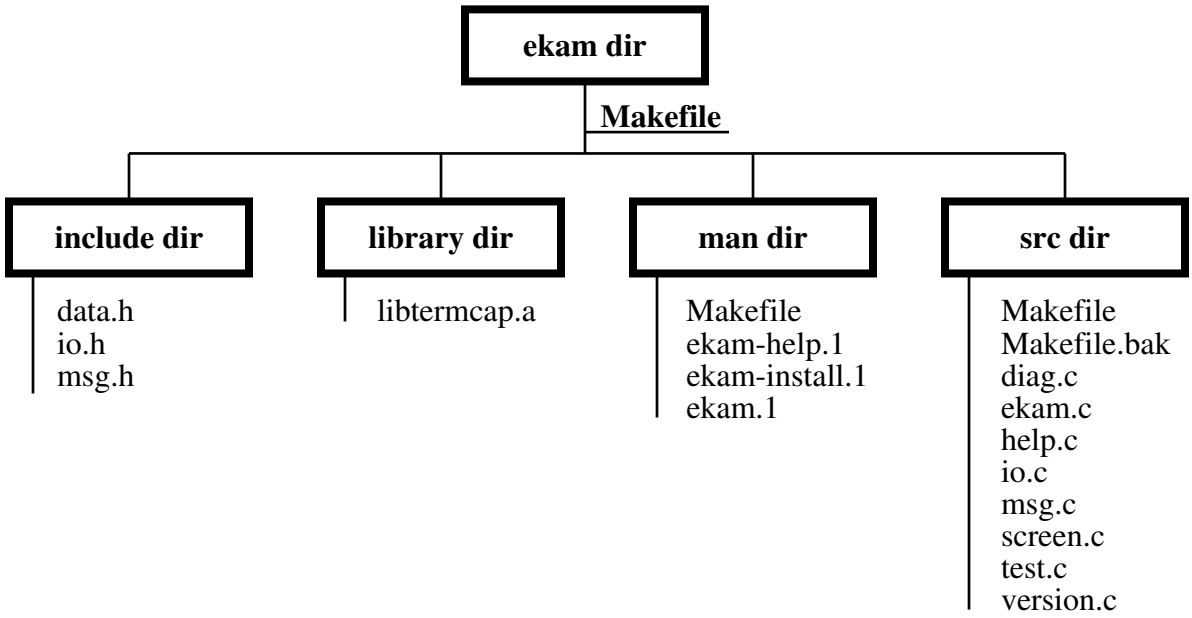
Dhrystone 1.0 (Optimized, Registers)

Sparc 2	42075/sec	(11 sec)	SUN C, -O4
NeXT 68040	28142/sec	(17 sec)	GNU C from rel 2.0
NeXT 68040	27223/sec	(18 sec)	GNU C from rel 1.0a
Sparc 1	21382/sec	(23 sec)	SUN C, -O4
Sparc 1	10467/sec	(47 sec)	SUN C, no specified opt
Vax 8650	9436/sec	(52 sec)	
NeXT 68030	7884/sec	(63 sec)	GNU C, Shell
SUN 3/260	7028/sec	(71 sec)	SUN C
NeXT 68030	6866/sec	(72 sec)	SUN C, Shell
NeXT 68030	6785/sec	(73 sec)	GNU C, Terminal
NeXT 68030	5920/sec	(84 sec)	SUN C, Terminal
SUN 3/60	4964/sec	(100 sec)	SUN C
SUN 3/80	4856/sec	(102 sec)	SUN C
SUN 3/140	3794/sec	(131 sec)	SUN C
Vax 780	1712/sec	(292 sec)	
SUN 2/120	1390/sec	(359 sec)	SUN C

Whetstone (Floating Point) Benchmarks

	Whetstones/sec	Seconds
Sparc 2	125.0000	8
NeXT 68040	38.4615	26
Sparc 1	30.3030	33
NeXT 68030 w/ 68881	14.9254	67
VAX 8650	7.0423	142
Sun 3/140 w/ 68881	6.7114	149

The Ekam Directory Structure



“this article was inspired out of my personal desire . . .”

To “Make” Or Not To “Make”

The writing of this article was inspired out of my personal desire to better understand the UNIX project-management program called ‘make’. The UNIX environment is full of all kinds of utilities that are designed to aid the UNIX user/programmer in everyday usage. Make is no exception and in my mind is a very ingenious and useful program that will no doubt increase your productivity and reduce your frustration. These Makefiles were written on a NeXT computer with Version 1.0 of the NeXTOS and were processed with the native make program on the NeXT (/bin/make).

This article is written for first time or semi-experienced make users and requires a rudimentary understanding of UNIX and Bourne shell syntax. It is based on a hypothetical project called ‘ekam’ of dubious real life practicality, but none the less provides a wonderful example of the functionality of make. The ekam project is composed of the directory structure detailed in Figure 1. As with many hierarchical projects, there is one top level directory with many subdirectories that contain the different categories of files. The ekam project contains a Makefile in the top level directory along with the src, include, library and man subdirectories. As there are actually three Makefiles in the ekam project, I will henceforth refer to the Makefile in the top level ekam directory (lines 1-15) as the ekam Makefile, the Makefile in the src directory (lines 16-84) as the src Makefile and the Makefile in the man directory (lines 85-124) as the man Makefile.

It might be instructive to first consider the high level objective of utilizing make in the ekam project, or any project for that matter. Once the ekam project is uncompressed and untarred, we wish to modify the three Makefiles to our specific site preferences and then build, install and eventually clean up the entire project from the ekam directory. The ekam Makefile therefore controls the src and man Makefile’s and orchestrates the entire project.

Make operates on the simple structure of dependency and command lines normally contained in a description file referred to as a Makefile. A dependency line consists of one or more targets, one or two colon(s) followed by zero or more component(s).

A command line is a TAB indented Bourne shell command or set of commands. The make program then repetitively asks the question; “Is the current target out of date (older) with respect to any of the components or does this target even exist?”. If the target is out of date or non-existent, then the command line(s) is executed to “build” a new target. These targets can be specified on the command line as in ‘make target-name’.

If a target appears on more that one single-colon-separated line, then only one of those dependency lines may have a command line associated with it. But, a target may appear on multiple dependency lines, and have multiple command lines if a double colon is used to separate the target from the component(s).

Upon invocation make first looks for a description file called “makefile” and if not found then looks for a file called “Makefile”. Alternative names can be used for description files and referenced with the ‘make -f non-standard-makefile’ syntax. Make parses the entire description file and concatenates all dependency lines and commands into one static set of rules.

I) The ekam Makefile.

```
 1  #
 2  # ...ekam/Makefile
 3  # davehieb::Sun Jan 13 11:07:41 GMT-0700 1991
 4  #
 5  SUBDIR = src man
 6
 7  .DEFAULT: all
 8
 9  all install clean:
10      @for i in ${SUBDIR}; do \
11          (cd $$i && \
12          echo ""; \
13          echo "'making $@' in $$i"; \
14          make ${MFLAGS} $@); \
15      done
```

As is normal with most UNIX flat files, anything after a sharp (#) character up to the newline is considered a comment (lines 1-4). Line 5 defines a macro called SUBDIR that upon evaluation contains the string ‘src man’. Macro names of length greater than 1 character are expanded with the ${MACRO_NAME} or $(MACRO_NAME) syntax as is shown in lines 10 and 14. Macro names of length 1 character can just be dereferenced with the ‘$A’ syntax, where ‘A’ is the name of the macro. Macro definitions may also be defined on the command line and passed into the description file with the following syntax:

% make all MACRO1=/root MACRO2=”1 2 3” ...

The first formal dependency line (9) instructs that if one or more of the three targets (all, install or clean) are specified, the command line (10-15) will be executed. In this dependency line there are no components present which simply implies that the command line will be executed. This command line is actually the “for name in list ...” Bourne shell construct that has been newline escaped by the backslash (\) characters. In general, a command line can be any valid shell command or construct as long as the command line either fits on a single line or is newline escaped.

The ‘-n’ command line option to make shows which command line(s) will be executed (if any) in order to bring a target up to date. Here is the output from a ‘make -n all’ command:

ekam % make -n all

```
for i in src man;
    do
        (cd $i &&  echo "";  echo "'making all' in $i";  make -n all);
    done
```

Therefore it is seen that the "for i in ..." command line construct produces the following steps for a 'make all' command:

1) for subdirectory "src".
   1.1) change directory into the directory and if that "cd" is successful.
      1.1.1) echo a newline.
      1.1.2) echo the string "'making all' in src".
      1.1.3) issue a 'make ${MFLAGS} all' command in the src directory.

2) ditto for subdirectory 'man' with each occurrence of 'all' being replaced with 'man'.

It is necessary to expand all shell variables that are not part of the environment (including those variables like ‘i’ in line 11) with a double dollar sign ‘$$i’ syntax. ‘$@’ is an internal macro that evaluates to the current target name. This is one of several internal macros that will be described in the other Makefile’s. The MFLAGS macro is an implicitly defined macro that contains the command line flags that the instantiating make was invoked with. A complete list of the implicitly defined macros (and much, much more) can be obtained with the following Bourne and Csh command:

Bourne-sh% make -fp - < /dev/null 2> /dev/null > make-defaults

Csh-sh% (make -fp - </dev/null > make-defaults) >& /dev/null

The ampersand ‘@’ preceding the ‘for’ command on line 10 suppresses the current command line from being echoed. Normally all command lines are echoed to the standard output unless the ‘-s’ command line option is used or the ‘.SILENT’ pseudo-target is defined in the Makefile. A pseudo-target is basically a Makefile option that is defined internal to the Makefile. Another pseudo-target has been defined on line 7 which serves as a dependency/command line set in the absence of an explicit target. This pseudo-target is used throughout these Makefile examples to make the following two commands in essence equivalent: ‘make all’ and ‘make’.

II) The src Makefile.

```
16  #
17  # ...ekam/src/Makefile
18  # davehieb::Sun Jan 13 11:07:41 GMT-0700 1991
19  #
20              ROOT =
21            BINARY = ekam
22             MKDIR = /bin/mkdir
23           INSTALL = /usr/bin/install
24           INCLDIR = ../include
25            LIBDIR = ../library
26     INSTALLBINDIR = ${ROOT}/usr/local/bin
27      INSTALLFLAGS = -c -o root -g wheel -m 0755
28              OBJS = diag.o ekam.o help.o io.o msg.o screen.o test.o
29              SRCS = diag.c ekam.c help.c io.c msg.c screen.c test.c
30            HFILES = ${INCLDIR}/io.h ${INCLDIR}/data.h ${INCLDIR}/msg.h
31
32              LIBS = ${LIBDIR}/libtermcap.a
33
34             RMCMD = ${RM} *.o core ekam
35
36  .DEFAULT: all
37
38  SUFFIXES: .o .c
39  .c.o    :
40      ${CC} -c -I ${INCLDIR} $<
41
42  all     : ${BINARY}
43
44  ${BINARY}: ${OBJS} ${LIBS}
45      ${CC} -o $@ ${OBJS} -L ${LIBS}
46
```

```
47  install: ${BINARY}
48      @if [ `whoami` != root ]; then \
49          echo "you must be root to install these files" ;\
50          exit 1;\
51      elif [ ! -d ${INSTALLBINDIR} ]; then \
52              echo "making: ${INSTALLBINDIR}" ;\
53              ${MKDIR} ${INSTALLBINDIR}; fi
54      ${INSTALL} ${INSTALLFLAGS} $? ${INSTALLBINDIR}/$?
55
56  depend  : ${SRCS}
57      -rm Makefile.bak
58      (sed '/^# DO NOT DELETE THIS LINE/q' Makefile && \
59      cc -M ${CFLAGS} -I ${INCLDIR} ${SRCS} | \
60      sed '/\.c/d' ) > Makefile.new
61      cp Makefile Makefile.bak
62      mv Makefile.new Makefile
63
64  clean   :
65      -${RMCMD}
66
67  # DO NOT DELETE THIS LINE, 'make depend' uses it #
68  diag.o: ../include/data.h
69  ekam.o: ../include/io.h
70  ekam.o: ../include/msg.h
71  ekam.o: ../include/data.h
72  ekam.o: /usr/include/stdio.h
73  ekam.o: /usr/include/stddef.h
74  ekam.o: /usr/include/stdarg.h
75  help.o: ../include/msg.h
76  help.o: ../include/data.h
77  io.o: ../include/io.h
78  io.o: ../include/data.h
79  msg.o: ../include/io.h
80  msg.o: ../include/msg.h
81  msg.o: ../include/data.h
82  screen.o: ../include/io.h
83  screen.o: ../include/data.h
84  test.o: ../include/data.h
```

The src Makefile begins with several macro definitions that define things like UNIX binaries, command options, make commands and the ekam system files. Note that this is the section of a Makefile that you would tailor to your site-specific needs. For example, locally compiled programs (like ekam) are often installed in the /usr/local/bin directory. If this does not conform to your local conventions, then you would modify the INSTALLBINDIR macro to reflect your local binary directory.

As you might have noticed, the RM macro (line 34) is not defined in the src Makefile and is actually an implicitly defined make macro. Also, it is perfectly legal (and recommended) to use cascading macro definitions like those in lines 26, 30, 32 and 34.

The first dependency line (42) instructs that the target ‘all’ depends on the component ‘ekam’ which in turn depends on the object and library files (line 44). Therefore when a ‘make all’ or ‘make ekam’ command is issued, command line (45) will try to compile the object and library files into an executable named after the target; ekam. The only problem is that at this point there are not any object files available for compilation. This is where the powerful concept of suffix rules comes into play.

Lines (38-40) establish a suffix rule which in essence defines a dependency/command line set for a class of targets, not a specific target. Line (38) states that the suffixes .o and .c are to be considered significant to make. Line (39) instructs that given a .c file, a .o file with the same base name can be produced with command line (40). The internal macro ‘$<‘ is similar in

meaning to '$?' in that it evaluates to the current component that is "newer" that the current target. The '$<' macro can only be used in suffix rules and likewise the '$?' macro can only be used in normal description file entries (not suffix rules).

Therefore, when command line (45) tries to check the object file components against the target, the suffix rule in line (39) is automatically triggered and command line (40) is executed. Line (40) in response individually compiles the .c source files into .o object file format so that line (45) will now have object files to compile (actually link) into an executable.

As is evidenced, suffix rules can do a lot of work by relying on the established nomenclature of the UNIX operating system. Dozens of implicitly defined suffix rules are available by default and a general understanding of them should be obtained (refer to the '-fp' command line options of make). By doing so, the differences in this suffix rule, and hence it's necessity, will be understood. A '.SUFFIXES:' line by itself in a Makefile resets the default suffix rules while a '.SUFFIXES:' line followed by a space separated list of suffixes (that are to be considered significant) adds to the default suffix rules.

Line (47) instructs that when a 'make install' is issued, command lines (48-54) will be executed and if necessary, the ekam component itself will become a target and get "built". Lines (48-50) check to see if the user trying to install the ekam binary is root and if not, exits the Makefile. If the user is root and the installation directory exists, command line (54) installs the binary in the correct directory. If the user is root and the installation directory does not exist, lines (52-53) create the installation directory.

Dependency line (56) allows the user to actually introduce explicit dependency rules (lines 68-84) into the Makefile that are deduced from the ekam source and include files. For each .c source file, the C preprocessor produces a dependency rule whose target is the object file name for that source file and whose dependency components are all the files #included in it. Line (58) utilizes sed to output the contents of the Makefile up to and including line (67) to the standard output. If this sed operation was successful, the output is actually redirected into the standard input of the C preprocessor (line 59) which generates the aforementioned dependency rules and thereupon directs it's standard output through another instantiation of sed (line 60) which before passing it's standard output to the file Makefile.new, deletes all the .c source file dependency rules. The .c source file dependency rules are redundant information that is already implied in the .o object file dependency rules.

Lines (61-62) copy the current Makefile to a backup and move the newly generated Makefile.new to Makefile. A 'make depend' command should therefore be executed once the .c source files and .h include files have been established and any time thereafter when a #include line is added or removed from source files.

Line (64) allows the user to "clean" up the directory after a build (and install) by removing the .o object files, a possible core file and the ekam binary. The command on line (64) and also line (57) is preceded by a '-' so that any return codes will be ignored. Normally any command returning an error code will result in the termination of the building of the current target. It is seen that these commands are not critical and we don't care to have the "clean" build terminated if for instance a core file can't be removed.

III) The man Makefile.

```
85  #
86  # ...ekam/man/Makefile
87  # davehieb::Sun Jan 13 11:07:41 GMT-0700 1991
88  #
89                ROOT =
90               MKDIR = /bin/mkdir
91               PAGER = /usr/bin/less
92             INSTALL = /usr/bin/install
93        INSTALLFLAGS = -o root -g wheel -m 0444
94       INSTALLMANDIR = ${ROOT}/usr/man/manl
```

```
 95        RAWMANPAGES = ekam.l ekam-install.l ekam-help.l
 96  FORMATTEDMANPAGES = ekam.roff ekam-install.roff ekam-help.roff
 97
 98            MANROFF = nroff -man
 99              RMCMD = $(RM) ${FORMATTEDMANPAGES}
100
101  .SUFFIXES: .roff .l
102  .l.roff :
103      ${MANROFF} $< > $*.roff
104
105  .DEFAULT: all
106
107  all      :  ${FORMATTEDMANPAGES}
108
109  install : ${RAWMANPAGES}
110      @if [ `whoami` != root ]; then \
111          echo "you must be root to install these files" ;\
112          exit 1;\
113      fi
114      @for i in $?; do \
115          ${INSTALL} ${INSTALLFLAGS} $$i ${INSTALLMANDIR}/$$i; \
116      done
117
118  view     : ${FORMATTEDMANPAGES}
119      @for i in $?; do \
120          ${PAGER} $$i; \
121      done
122
123  clean    :
124      -${RMCMD}
```

A ‘make all’ in the man directory results in the suffix rules automatically formatting the unformatted roff man pages and saving them in files of the same base name with a .roff extension. This is accomplished by another internal macro: ‘$*’, on line 103 that evaluates to the base name of the current component. This internal macro, like the ‘$<‘ macro, can only be used in .SUFFIX rules.

A ‘make install’ starts off in the same manner as the src Makefile did by checking to see if the user is root. Lines (114-116) set up a “for name in ...” looping construct that loops through each of the unformatted man pages and installs them in the manpage directory. Here is the ‘make -n install’ output:

```
ekam/man % make -n install

if [ `whoami` != root ]; then
        echo "you must be root to install these files" ;
        exit 1;
fi
for i in ekam.l ekam-install.l ekam-help.l;
        do
                /usr/bin/install -o root -g wheel -m 0444 $i /usr/man/manl/$i;
        done
```

A ‘make view’ results in the unformatted man pages being formatted (if they haven’t already been formatted with a ‘make all’) and then having each formatted man page previewed with the user defined PAGER macro.

David R. Hieb
davehieb@eclipse.Colorado.EDU