CONTENTS

1988 Elections for Board of Directors .......................................................... 3
Dallas USENIX Conference ............................................................................. 4
    Tutorials ................................................................................................. 4
Supporting Members ......................................................................................... 5
Fifth Workshop on Real-Time Software and Operating Systems ..................... 6
Call for Papers: Summer 1988 USENIX Conference ........................................ 7
Future Workshops ............................................................................................. 8
Call for Papers: UNIX Security Workshop ....................................................... 8
    Jaap Akkerhuis
Book Review: A Software Tools Sampler ....................................................... 13
    Ozan Yigit
French UNIX User’s Group Conference ......................................................... 16
EUUG Spring 1988 Conference ................................................................. 17
Call for Papers – EUUG Autumn Conference ................................................. 17
UNUNET Progress Report ............................................................................. 18
2.10BSD Software Release Available .......................................................... 18
Future Meetings ............................................................................................... 19
Publications Available .................................................................................... 19
4.3BSD Manuals .............................................................................................. 20
    4.3BSD Manual Reproduction Authorization and Order Form .................. 21
Local User Groups ........................................................................................... 22

The closing date for submissions for the next issue of ;login: is February 26, 1988
NOTICE

::login:: is the official newsletter of the USENIX Association, and is sent free of charge to all members of the Association.

The USENIX Association is an organization of AT&T licensees, sub-licensees, and other persons formed for the purpose of exchanging information and ideas about UNIX\(^1\) and similar operating systems and the C programming language. It is a non-profit corporation incorporated under the laws of the State of Delaware. The officers of the Association are:

President
Alan G. Nemeth
\texttt{usenixlagn}

Vice-President
Deborah K. Scherrer
\texttt{usenixlscherrer}

Secretary
Waldo M. Wedel
\texttt{usenixlwedel}

Treasurer
Stephen C. Johnson
\texttt{usenixlscj}

Directors
Rob Kolstad
\texttt{usenixlkolstad}
Marshall Kirk McKusick
\texttt{usenixlmkusick}
John S. Quarterman
\texttt{usenixljsq}
David A. Yost
\texttt{usenixlday}

Executive Director
Peter H. Salus

The editorial staff of ::login:: is:

Editor & Publisher
Peter H. Salus
\texttt{usenixlpeter}

Technical Editor
Kevin Baranski-Walker
\texttt{usenixlkevin}

Copy Editor
Michelle Dominijanni
\texttt{(masscomp,usenix)!mmp}

Production Editor
Tom Strong
\texttt{usenixlstrong}

Other staff members are:

Betty J. Madden
Office Manager

Emma Reed
Membership Secretary

USENIX Association Office
P.O. Box 2299
Berkeley, CA 94710
(415) 528-8649
\texttt{(uunet,ucbvax)!usenix!office}

---

\(^1\)UNIX is a registered trademark of AT&T.

Contributions Solicited

Members of the UNIX community are encouraged to contribute articles to ::login::. Contributions may be sent to the editors electronically at the addresses above or through the U.S. mail to the Association office. The USENIX Association reserves the right to edit submitted material.

::login:: is produced on UNIX systems using \texttt{troff} and a variation of the \texttt{-me} macros. We appreciate receiving your contributions in \texttt{n/troff} input format, using any macro package. If you contribute hardcopy articles please send \texttt{originals} and leave left and right margins of 1" and a top margin of 1$\frac{1}{4}$" and a bottom margin of 1$\frac{1}{4}$".

Acknowledgments

The Association uses a SUN\(^2\) 3/180S running SUN OS for support of office and membership functions, preparation of ::login::, and other Association activities.

Connected to the SUN is a QMS Lasergrafix\(^3\) 800 Printer System donated by Quality Micro Systems of Mobile, Alabama. It is used for general printing and draft production of ::login:: with \texttt{ditroff} software provided by mt Xinu.

This newsletter is for the use of the membership of the USENIX Association. Any reproduction of this newsletter in its entirety or in part requires written permission of the Association and the author(s).

\(^2\)SUN is a trademark of Sun Microsystems, Inc.

\(^3\)Lasergrafix is a trademark of Quality Micro Systems.
1988 Elections for Board of Directors

The biennial elections of the Association will be held this spring.

After the Dallas meeting nominations from the membership will remain open until February 26. The procedure for nominations by the membership is a written statement of nomination signed by at least five (5) members in good standing (or five separate nominations), to be submitted to the Executive Director at the Association office and received by noon, PST, February 26.

Ballots will be sent to all paid-up members as of March 1, 1988, on or about March 14. Members will have until April 8 to return their ballots, in the provided envelopes, to the Association office. The results of the election will be announced at the San Francisco meeting and in the May/June issue.

The Board is comprised of eight directors, four of whom are “at-large.” The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential, with those candidates with the largest number of votes being elected. Newly elected directors will take office immediately following the San Francisco conference in June.

Report of the Nominating Committee

The USENIX Association Nominating Committee announces that it has nominated the following to run in the 1988 election:

Alan G. Nemeth      Deborah K. Scherrer
Deborah K. Scherrer  Stephen C. Johnson
Rob Kolstad          M. Kirk McKusick
M. Kirk McKusick     Bob Morris
Bob Morris           Sharon Murrel
Sharon Murrel        Michael D. O'Dell
Michael D. O'Dell    John S. Quarterman
John S. Quarterman   Charles H. Sauer
                      Prime Computer Corp., Framingham, MA
                      mt Xinu, Berkeley, CA
                      Ardent Computer, Inc., Sunnyvale, CA
                      Convex Computer Corp., Richardson, TX
                      Univ. of California, Berkeley, CA
                      Univ. of Massachusetts & Interleaf, Inc., Boston, MA
                      AT&T Bell Laboratories, Murray Hill, NJ
                      Maxim Technologies, Vienna, VA
                      Texas Internet Consulting, Austin, TX
                      IBM, Austin, TX
                     
The Nominating Committee felt that the USENIX Board must reflect the diverse needs and interests of the membership. The Board needs a range of talents, and it needs a good mix of experience and new ideas. Candidates must be committed to USENIX, and must in addition have the skills needed to direct a rapidly growing organization. The Nominating Committee attempted to balance these requirements in its deliberations.

The committee did not feel the need to nominate a large slate of candidates because the USENIX election procedures make it easy for the membership to nominate additional candidates.

As has been the practice of past committees, the committee has nominated one of its own members. We wish to note that the slate has the unanimous support of the committee.

Finally, the members of the committee wish to thank Lew Law for his efforts. Lew was originally chair of the committee until he resigned for personal reasons. His efforts were important to the success of the committee's work.

Michael Tilson, Chair
Bruce Borden         Peter Langston
Tom Ferrin           Mike O'Dell
Dallas USENIX Conference

Rob Kolstad, Program Chair

Tuesday through Friday, February 9-12, 1988, mark the Winter 1988 USENIX Technical Conference in Dallas, Texas. The Grand Kempinski Hotel (formerly The Registry Hotel) will host the conference's two days of tutorials and two days of technical sessions.

There will be 21 tutorials, including several new ones. The two days of technical presentations will include two special half-day sessions on large-scale networks of workstations (as implemented in the Andrew project at Carnegie-Mellon University and MIT's Project Athena) and sessions devoted to new systems management techniques, the popular work-in-progress sessions late in the day, and other talks on state-of-the-art technical advances in UNIX and its applications.

The program committee has reviewed over 80 abstracts that were submitted for the conference. I am confident that the program will be an exceptionally strong and interesting one.

Conference information may be obtained from:
Judy DesHamais (213) 592-1381 or 592-3243
USENIX Conference Office (uunet,ucbvax)!usenix!judy
P.O. Box 385
Sunset Beach, CA 90742

Tuesday Tutorials

**Introduction to 4.3BSD Internals**
INSTRUCTOR Thomas W. Doepner, Jr., Brown Univ.

**Introduction to UNIX System V Internals**
INSTRUCTORS Steve Buroff and Curt Schimmel, AT&T

**Software Development using C and UNIX**
INSTRUCTOR Rob Kolstad, Convex Computer Corp.

**UNIX System V Remote File Sharing (RFS)**
INSTRUCTORS Michael Padovano and Michael Forbes, AT&T

**Managing a Local Area Network (LAN)**
INSTRUCTORS Evi Nemeth, Brent Browning and Bob Coggeshall, Univ. of Colorado, Boulder

**X Window Systems® Programming**
INSTRUCTOR Michael Gancarz, DEC

**UNIX System V Streams Module and Driver Design**
INSTRUCTORS Sathis Menon, Hari Pulijal and Stephen Rago, AT&T

**An Introduction to C++**
INSTRUCTOR Robert Murray, AT&T Bell Laboratories

**Sendmail, News and uucp, or How to Set Up your Mail and Bulletin Board**
INSTRUCTOR Smoot Carl-Mitchell

**Managing a Network of NFS™ Systems**
INSTRUCTOR Ed Gould, mt Xinu

X Window Systems is a trademark of MIT
Wednesday Tutorials

Advanced 4.3BSD Internals: Data Structures and Algorithms
INSTRUCTORS Mike Karels and Marshall Kirk McKusick, Univ. of California, Berkeley

Advanced UNIX System V Internals
INSTRUCTORS Steve Buroff and Curt Schimmel, AT&T

Language Construction Tools on the UNIX System
INSTRUCTOR Steve Johnson, Ardent Computer, Inc.

UNIX Device Driver Design (4.2/4.3BSD)
INSTRUCTOR Daniel Klein, Software Engineering Institute, CMU

Special Topics in C
INSTRUCTOR Carol Meier, Consultant

Open Network Computing (ONC™) and NFS™
INSTRUCTORS Mark Stein and Sally Ahnger, Sun Microsystems, Inc.

Network Extensible Window System (NeWS™)
INSTRUCTORS David A. LaVallee and Owen Densmore, Sun Microsystems, Inc.

An Introduction to 3D Computer Graphics
INSTRUCTOR Bruce Borden, Ardent Computer, Inc.

POSIX Application Implementation
INSTRUCTORS Shane P. McCarron, Minnesota Educational Computing Corp., and John S. Quarterman, Texas Internet Consulting

MACH
INSTRUCTOR Avadis Tevanian, Jr., Carnegie-Mellon Univ.

UNIX 4.xBSD System Administration
INSTRUCTORS Rob Kolstad, Convex Computer Corp., and Eva Nemeth, Univ. of Colorado, Boulder

ONC, NFS and NeWS are trademarks of Sun Microsystems, Inc.

Get your face saved by the FaceSaver in Dallas

Supporting Members

USENIX has a special class of membership for those organizations which have made special contributions to the functioning of the Association.

AT&T, Quality Micro Systems, DEC, and mt Xinu have been Supporting Members of the USENIX Association for years.

This year, the Association has several new supporting Members:

Convex Computer Corporation, for its cooperation in production of the 1987 Software tapes;

Sun Microsystems for its donations to upgrade the Association's hardware; and

Sybase, for its donation of database software - by the time this issue is printed both the membership and the mailing list will be kept using Sybase.

The Association is grateful for its corporate support, without which we would certainly serve our members less well.

- PHS
This year's workshop broadens the scope to include general real-time systems. This workshop will bring together researchers, designers, and implementers of real-time operating systems and software. There will be a substantial emphasis on practical experience, so workers from industrial organizations are encouraged to attend. Topics of specific interest include:

- Primary requirements of real-time systems
- Distributed real-time operating systems
- Application-specific operating systems
- Practical experiences and implications
- Exotic applications: medicine, music, etc.
- Architectural support for real-time

- Language, programming support, and reusability
- Types of real-time constraints
- Scheduling and resource management
- Predictability, adaptability, and maintainability
- Reliability and fault tolerance
- Instrumentation and performance measurement
- Case studies

The format of the workshop will be geared to encourage intense technical interactions and focused discussions.

Attendance will be limited to between 75 and 100 active workers in the field. To participate in the workshop, please submit four copies of an extended abstract or position paper of up to 5 pages describing your current efforts to Lui Sha by March 1, 1988. The abstract should focus on insights and lessons gained from recent research and practical experience. Complete details regarding the workshop will be sent to all participants along with the acceptance letters by March 21, 1988. A digest of accepted abstracts will be made available to participants at the workshop.

General Chair
Professor John Stankovic
Department of Computer and Information Science
Graduate Research Center
University of Massachusetts
Amherst, MA 01003
(413) 545-0720
stankovic@cs.umass.edu

Program Co-chair
Dr. Marc Donner
IBM Research
P.O. Box 218
Yorktown Heights, NY 10598
(914) 945-2032
donner@ibm.com

Program Co-chair
Dr. Lui Sha
Computer Science Department
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213
(412) 268-7668
sha@k.gp.cs.cmu.edu
Call for Papers
Summer 1988 USENIX Conference
San Francisco
June 20-24, 1988

Papers in all areas of UNIX-related research and development are solicited for formal review for the technical program of the 1988 Summer USENIX Conference. Accepted papers will be presented during the three days of technical sessions at the conference and published in the conference proceedings. The technical program is considered the leading forum for the presentation of new developments in work related to or based on the UNIX operating system.

Appropriate topics for technical presentations include, but are not limited to:

- Kernel enhancements
- UNIX on new hardware
- User interfaces
- UNIX system management
- The internationalization of UNIX
- Performance analysis and tuning
- Standardization efforts
- UNIX in new application environments
- Security
- Software management

All submissions should contain new and interesting work. Unlike previous technical programs for USENIX conferences, the San Francisco conference is requiring the submission of full papers rather than extended abstracts. Further, a tight review and production cycle will not allow time for rewrite and re-review. (Time is, however, scheduled for authors of accepted papers to perform minor revisions.) Acceptance or rejection of a paper will be based solely on the work as submitted.

To be considered for the conference, a paper should include an abstract of 100 to 300 words, a discussion of how the reported results relate to other work, illustrative figures, and citations to relevant literature. The paper should present sufficient detail of the work plus appropriate background or references to enable the reviewers to perform a fair comparison with other work submitted for the conference. Full papers should be 8-12 single spaced typset pages, which corresponds to roughly 20 double spaced, unformatted, typed pages. Format requirements will be described separately from this call. All final papers must be submitted in a format suitable for camera-ready copy. For authors who do not have access to a suitable output device, facilities will be provided.

Four copies of each submitted paper should be received by February 19, 1988; this is an absolute deadline. Papers not received by this date will not be reviewed. Papers which clearly do not meet USENIX's standards for applicability, originality, completeness, or page length may be rejected without review. Acceptance notification will be by April 4, 1988, and final camera-ready papers will be due by April 25, 1988.

Send technical program submissions to:

Sam Leffler (415) 258-8195
SF-USENIX Technical Program ucbvxlsfusenix
PIXAR
3240 Kerner Blvd.
San Rafael, CA 94901

FULL PAPERS ARE DUE FEBRUARY 19, 1988
Future Workshops

In addition to the Technical Conferences in Dallas and San Francisco, the USENIX Association will sponsor four workshops during 1988. These will be

- UNIX Security, in Portland, OR, August 29-30
- UNIX and Supercomputing in Pittsburgh, PA, September 26-27
- Large Installation System Administration II, Monterey, CA, November 17-18

Melinda Shore of the Pittsburgh Supercomputer Center and Lori Grob of New York University are the co-Program Chairs for the Supercomputing workshop; Alix Vasiltos of MIT's Project Athena is the Program Chair of the Systems Administrator's workshop; Matt Bishop of Dartmouth College will be the Program Chair for the UNIX Security workshop.

As a result of the success of the C++ workshop in Santa Fe, NM, in November, there may be a "mini-conference" concerning C++ in October. The tentative dates are 17-20 October in Denver.

Call for Papers
UNIX Security Workshop

Matt Bishop is the chair for the UNIX Security Workshop to be held in Portland, Oregon, on Monday and Tuesday, August 29th and 30th, 1988. This workshop will bring together researchers in computer security dealing with UNIX and system administrators trying to use UNIX in environments where protection and security are of vital importance.

Some topics to be considered include: password security (password file integrity, enforcing choice of a safe password, spotting and handling crackers), network security (problems arising from logins over an unprotected Ethernet, containing a break-in to one machine in a networked environment), file system security (auditing packages, security in an NFS environment), new designs to obtain C-level (or better) certification, making existing UNIX systems more secure, and locating and fixing UNIX security problems.

Format

Each participant will submit electronically to {ihnp4,decvax}@dartvax%bishop a one or two page summary describing a solution to some problem. The summary should contain a description of the problem and a description of the solution detailed enough that fellow researchers and administrators can implement or use it. Also, include with your submission five (or so) topics that you’d like to hear about.

The workshop chair will collate the papers to schedule sessions for appropriate audiences. It is anticipated that some sessions will include all participants; some will be for smaller groups. Send your submissions to the chair by noon, EST July 1, 1988.

For further details on the workshop:
Matt Bishop
Dept. of Mathematics & Computer Science
Bradley Hall
Dartmouth College
Hanover, NH 03755
(603) 646-3267
{ihnp4,decvax}@dartvax%bear%bishop
bishop%bear.dartmouth.edu@relay.cs.net

For details about registration, contact the USENIX Conference Office.
The Set Space

Probably one of the most neglected troff requests is the .ss request. To revive your memory, the form of this request is .ss N, which sets the space-character size to N/36 ems. The default value is (nearly always) 12/36th of an em. This request works in the current environment, and if no argument is specified, the request is ignored. It is always ignored in nroff. The space-character size is the minimum space between words in adjusted text. It is also the word spacing you get when the text isn’t adjusted. This is known in the typographical world as the set space. It also sets the width of the unpaddable space character ‘\’. The widths of the 1/6 em narrow ‘\|’, the 1/12 em half-narrow ‘\”, and, of course, the digit width space character ‘\0’, are not affected.

The set space is dependent on the font size, and therefore also on the “em.” The size of the set space might vary between the different typefaces; some typefaces require a different value, depending on the design. Obviously, a constant width font wants to have a set space the width of normal characters, so that when the text isn’t justified (adjusted in troff terminology) it would appear to have been produced on a typewriter. Also, you may want to change it depending on the style of the lay-out. There are some still other reasons why you might decide to play with the .ss request.

Widows and Orphans

It happens all the time, you have written your article or book, keyed it into your favorite machine, formatted it with troff, and are dissatisfied with the output. After some advise from a neighbour, or reading a book about typography, the mess looks somewhat better, but there are still problems with orphans, etc. The word at the top of page 24, just before the opening of section 3, actually belongs on the bottom of page 23. Of course, one way to deal with this is to rewrite page 20 to 23, so it will fit, but then the opening of section 3 suddenly appears on the bottom of page 23, yuck! You can of course force this to page 24 by applying a .bp, or even better a .ne request, but then you have an enormous white space left on page 23. Another way to fix this problem is to use the .ss request. About half way through the text that is going to be on page 23, you can make the set space somewhat smaller than the default value; this will be just enough to pull the offending text off of page 24 and put it onto page 23, and the difference in spacing will hardly be noticeable. Of course, don’t forget to reset it to its previous value so you can play the same trick elsewhere.

The Late Paper Problem

Papers will always be finished too late. Nobody disputes that. The problem is, that after it finally is finished and about to be sent off, there is yet another mistake found. The errand boy is in your office, ready and waiting to bring your masterpiece to the printer. A double word is is found on the last page. Pushing the complete 200 pages through troff again on an overloaded machine will just be too time consuming. An easy fix – when the offending word doesn’t take a lot of space – is to retypeset the offending line and force it to be spread over the same line length using the \p escape. On the other hand, if there is a word missing, the set space is first reduced, before forcing the justification. Then the output can be glued on top of the offending line. If the errors are in rather big words, you may need to typeset a couple of lines with the changed set space.
Of course, you must still be prepared to do everything again. With your luck you will find that some coffee will accidentally be spilled over the galleys.

**A Large Number**

Your nephew pops up and wants to print the largest known Mersenne prime number in his school paper. Since this number is $2^{216091} - 1$, he noticed that his calculator didn’t grok it. By now he has discovered that typing in 65050 numerals on his IBM composer would probably not go without any error, so he comes to you in a state of desperation.

Calculating the number is not a big deal and of course you can keep the answer somewhere in a file. However feeding this to troff just produces an error message, "Word overflow." After grumbling "why are there always these stupid arbitrary limits in UNIX programs," you will realise that troff won’t hyphenate the number and that you wouldn’t want it to anyway; it wouldn’t look right.

Apparently you have to break the number in smaller units and glue them together. Breaking the number can be done with your favorite editor which can handle arbitrarily large files or just a little C program, replacing each numeral with itself and a space, and inserting some newlines as well. Then you set the setspace to the smallest possible value, 1/36th of an em.

Now you notice that the last line of the block of numerals has less space than all the lines before. This is fixed by making the line length exactly to the length of the numerals and spaces you want it to have, so the spaces are not stretched. As an alternative, you can of course set the lot not left justified. So this will get you something like:

```
746093103064661343687395794005114895402
28754089773288051130497779366272527096
... lots of numerals omitted
411805217750684693278676451411877691336
204103815528447
```

If you really get carried away by this problem, you can edit the ditroff output as well to get rid of that little space. Of course, that will make your line length somewhat shorter, but who cares, it’s just for your nephew after all.

**Constant spacing**

The constant spacing request, \texttt{.cs F N M}, will set the width of every character in the font \texttt{F} to \texttt{N/36th} em. If the optional \texttt{M} is given, the em is \texttt{M} points. The characters are centered in this space, even if this character is wider than the available space.

This suggests that you can use this request to generate a constant width font.

These lines are typeset with \texttt{.cs R 16}.

There is even spacing for the characters, but as you can see, the result is an insult to the eye.

The result depends a bit, of course, on the variety of widths in the font. But it is much better to use a typewriter style font if you want to simulate a typewriter. Even then you might want to use the \texttt{.cs} request, with the constant width set to the width of the typewriter font, in case you use a character which isn’t in that font. It will then get it from the special font (S) when mounted. With some luck the result won’t be too bad.\footnote{Alternatively, when using a constant width font and you need to print a character from the special font you can use the \texttt{\textbackslash o'ab'} construct to overprint the special font character with a blank, as long as the special font character is not wider than a blank. E.g., \texttt{\textbackslash o'\textbackslash o'ab'}. –TS}

**Watch that man**

Although it appears that the constant width request is quite useless, it can come in handy from time to time. For instance, maybe after an EUUG conference somebody sends you data like this:

```
0x0000,0x1FA0,0x0000,
0x0000,0x7FFC,0x0000,
0x0000,OxFEFF,0x0000,
... etc
```

The data is actually describing a 48 by 48 pixel bitmap. Apparently a picture has been taken of you, but showing off this hexadecimal stuff really won’t do much to impress the rest of the family. So how do you turn these numbers back into your usual self?
To print it on a line printer, you can either write a C program or use the following `sed` script.

```bash
# icontox
sed 's/0X//g
s/,//g
s/1/#/g
s/2/#/g
s/3/##/g
s/4/#/g
s/5/##/g
s/6/###/g
s/7/#/#/g
s/8/# #/g
s/9/# #/g
s/[Aa]/# #/g
s/[Bb]/# ##/g
s/[Cc]/## #/g
s/[Dd]/### /g
s/[Ee]/####/g
$* | sed 's/ *$//''
```

Applying this script you get lines which look a bit more printable.

```
-HHHH
```

To turn this into something nice, you need to work somewhat harder. This is what I used for this paper.

```
.nf \" nofill mode wanted
.nn y \n(.lu/2 \" half line length
.ps 9 \" pointsize 9
.tr #. \" translate # to .
.cs R 3 \" constant spacing
.vs \w' \" vertical spacing
.A" equal to horizontal
.l" center the face
.in \nyu-w'###############'u
(output of icontox comes here)
.l" now restore everything to normal
.cs R
.tr ##
.ps
.vs
.in
.fi
```

After doing all this you will get something like the next picture.

As you can see, this result can be shown to the rest of your family. To get the best possible result, you might want to play with the constant spacing factor or pointsize a bit, since these depend on the size of the full stop for your output device. Also, if you want to enlarge the picture, you can just bump up the pointsize and the constant spacing factor.

**Bitmap fonts**

This technique suggests that it is easy to include screen dumps from your favourite bitmap screen in your paper. It turns out not to be that easy. A locally written screendump program family for “blit terminals” (or DMD5620) has been converted to give it’s output in a format which can be processed by `troff`. To prevent the “Line overflow” message, `troff` had to be adapted. Apart from that, it takes considerable time for `troff` to process a bitmap created like this, about 820 CPU seconds for a blit screen.

The usual way to include screen dumps is to pass them through `troff` using the `!` escape or the `.ef` request (for modern versions of `troff`) and to do major surgery on the `troff` output. The problem with that is that `troff` loses control of the output, since it doesn’t know what magic might be done afterwards. A more elegant way of handling this would be to have a font available on the output device which describes pixels.

Consider a character set, where each character describes 8 pixels. So the font will contain 255 different characters of the same width. For displaying the bitmap of a 1K×1K screen you only need 1024 lines of 256 characters. The input to `troff` would be something like:

```
.nf \" nofill mode
.ft Bm \" select bitmap font
.A\b(es)(*A\- -
.fi \" restore fill mode
.ft \" and font
```

The moment that you want to have a dump of a colour screen, you are of course lost, but as long as there is not a way to directly print colours without colour separating, it is not a
real problem. Get your camera out and take that picture. The print shop will do the colour separation for you as usual.

If you really want to be fancy, you can have the screendump program do the colour separation.

Acknowledgements

The idea of abusing the set space request for patching up papers at the last moment comes from Bob Garufy. Rob Pike delivered the bitmaps and the icontox program. Paul Vitányi inspired the “large number problem.” Jim McKie made the local screendump program family. Paul Klint suggested the idea for the need of a bitmap character font. Peter J. Weinberger permitted the use of his picture. Carol Orange assisted in turning this article into quasi-American English.

POSIX Status

There was a meeting of IEEE 1003.1, 1003.2, 1003.3, and 1003.4 7-11 December 1987 in San Diego. 1003.1 is effectively out of business except for balloting, though the forthcoming and future reports will have quite a bit to say about the other subcommittees.

75% of the 1003.1 balloting quorum had responded by last week, and balloting was then closed. Balloting resolution letters from the technical reviewers to the quorum are expected in about the third week of January.

The USENIX POSIX Workshop led to USENIX, among others, submitting a balloting objection because the material in the draft standard and rationale on process groups and job control was incomprehensible. It also led to extensive discussions on those subjects by Berkeley and AT&T implementors and others. They expect to produce new, more understandable, more useful, and generally better wording for related areas in the standard in time for ballot resolution. The workshop also led to improvements in other areas of the standard.

In addition to USENIX, X/OPEN also balloted no. /usr/group balloted yes with four specific objections.

The 10-14 October meeting will be in Hawaii. There will be an ISO SC22 WG15 meeting in Japan the following week (and the preceding week is EUUG in Lisbon). The next meeting is 14-18 March at the Ritz-Carlton Hotel in Washington, D.C.

There was also an initial meeting of the U.S. TAG (Technical Advisory Group) to ISO SC22 WG15, the ISO version of 1003.1. USENIX’ representative is John Quarterman with Shane McCarron as alternate. /usr/group is also represented.

John Quarterman

Professional Activities

For the information of those who care, Peter H. Salus, the USENIX Executive Director, has been elected to a two-year term as President of the South Asian Literary Association (1988-89). Shantih.
Book Review

A Software Tools Sampler

by Webb Miller

Reviewed by Ozan Yigit
Department of Computing Services
York University
{(uunet,utzoo)!yunexus!oz

Introduction

Webb Miller’s new book A Software Tools Sampler should satisfy those appetites already whetted by earlier tools-oriented books by authors such as Kernighan and Plauger. Even more exciting, is the fact that, unlike those books, the chosen programming language for this book is C. Not surprisingly, Miller’s book is based on teaching an upper-level tools course given at the University of Arizona for which Kernighan and Plauger’s Software Tools¹ was the primary text.

A brief look at the table of contents suggests that this is an extension of the Software Tools books, with the fleshing out of several important topics that had received inadequate treatment elsewhere.

Detailed tour

The first of five chapters, “Getting Started,” quickly presents a small library of utility procedures used later in the book. For completeness, it also includes a rudimentary program to change the modification time of a file (tweak), another to locate all occurrences of a string (find), a simple but effective macro processor (macro), and a utility for generating the most effective compile command for a set of source files, based on modification times and filename suffixes (compile).

¹ The University of Arizona has had its own software tools distribution ever since the release of the software described in Kernighan and Plauger’s book.

The second chapter deals with a dependency-based file updating tool called update which is very similar to the UNIX make. Anyone who has read the UNIX documentation for make has an idea of how it works and how it may be implemented. Unfortunately, the actual implementation details are not discussed in the available UNIX literature, and some important issues such as off-line file updating (creating a command file for later execution) versus real-time file updating (immediate execution of update commands) are not even considered.

This chapter fills the gap with a thorough discussion of dependency graphs (sometimes called AOV-Networks) and various incrementally refined algorithms based on these graphs for the file update problem. The resulting program, although syntactically different, contains the basic functionality of make, and addresses the off-line update problem. Of course, the simple macro processor described in chapter one provides the shorthand facility for the program. Coupled with a paper published in Software-Practice and Experience (see additional readings at the end of this review), this chapter offers possibly the best available description of the internal workings of make.

The third chapter is devoted to file comparison programs that determine instructions for converting one file into another. The file comparison techniques have a variety of diverse uses. One of the earliest algorithms was invented by biologists to determine how many amino acids need to be deleted and inserted to convert one protein to another. In this chapter, two such algorithms are
discussed, where both algorithms generate the optimal sequence of instructions (edit script) to convert one file into another.

The first algorithm is originally by Walter F. Tichy, the author of a very well known revision control system (another application for file comparison techniques) for Berkeley UNIX, called rcs. This algorithm uses only append and copy instructions for conversion, and is presented in this chapter as a program called bdiff. The second algorithm, developed by Miller, uses only the insert and delete instructions. It was originally published in the November, 1985 issue of Software-Practice and Experience. A program that embodies the algorithm, called fcomp, is tested against the classic UNIX diff based on a different and well-known algorithm by J. W. Hunt and T. G. Szymanski. In normal circumstances, where the differences between the two files are small, it runs about four times faster than diff.2 Miller demonstrates that the difference in performance is strictly due to the algorithm, and not due to a tight implementation. There is also a discussion of worst case performance for both fcomp and diff.

Chapter three concludes with a very detailed description of a programming project based on the file comparison algorithms, a Version Control System, similar to AT&T's SCCS (Source Code Control System) and Tichy's rcs (Revision Control System). Completing this project should end the suffering for those who lack such a revision control facility in their programming environment.

The fourth chapter is concerned with pattern matching, with special emphasis on regular expressions. The first part of the chapter deals with the find program introduced in chapter one, and develops a much faster (about eight times) version of the program, called fastfind. Improved performance is obtained by the use of an better pattern matching algorithm similar to Knuth-Morris-Pratt algorithm and a clever circular buffer for file input.

The second part of the chapter is devoted to a proper implementation of the regular expression pattern matching algorithm, based on non-deterministic finite automata.

There is an overwhelming amount of literature on regular expressions. Practically every book on compiler design dedicates a chapter to regular expressions under the discussion of lexical analysis. It is also true, however, that only a fraction of this literature goes beyond the theoretical aspects to discuss actual implementation details. The implementations included in the Software Tools books cover only the ad hoc implementation of partial regular expressions, and omit the more difficult issue of the alternation [or] construct. For his part, Miller apparently decided to cut no corners, hence, this chapter implements full regular expressions that follow the traditional algorithms.

Following the details for implementing a parser to generate expression trees for regular expressions, a variation is introduced to generate a non-deterministic, finite-state machine instead of the expression trees. The rest of the chapter is a thorough discussion of how to use the finite-state machine to match text lines. The efficiency of the resulting pattern matcher is further examined, and it evolves into a final program called match, which internally uses a lazy evaluation technique for improved performance.

The material presented in this chapter is sufficient to develop sophisticated applications based on full regular expressions, such as lexical analyzer generators. (See additional readings.)

The last and longest chapter of A Software Tools Sampler implements one of the most popular software tools, a screen-oriented interactive text editor.3 Miller chose to model his editor, s, after the popular Berkeley screen editor, vi.

The sections in chapter five follow the logical organization of the screen editor: they cover the command processor, the buffer module and the screen module of the editor. While the sections covering the command processor and the buffer module are interesting, the most important section in this chapter

2 This claim has been verified by the reviewer.

3 In their books, Kernighan and Plauger show the implementation of a line editor.
deals with the development of the screen module. In most high-quality screen editors, the screen manager portion of the editor attempts to minimize the amount of data sent to the screen to update its visual contents. The screen update problem consists of both the algorithmic issues, and the low-level terminal-specific problems. Miller's important contributions in this particular problem area make this section particularly noteworthy (see additional readings). Here Miller develops an ad-hoc algorithm to generate minimal terminal control sequences to update the screen contents.4

The section contains a detailed design discussion of the update algorithm used in s. It also exposes the failure, in certain conditions, to generate optimal update sequences, as indicated by visually unpleasant changes to the screen. The resulting screen module is a reasonably autonomous, and may be adapted for other uses. Chapter five includes about 3000 lines of C code for the entire editor, built in a modular fashion. With the material presented here, another editor with a somewhat different interface (such as an Emacs-like editor) can be built quickly.

In general

This book is a must for programmers who take their craft seriously. Although most of the tools covered are found in UNIX and other systems, such a clean exposition of the algorithms and issues behind these tools were not so readily available before the publication of A Software Tools Sampler. Now, programmers are no longer restricted to an obscure or otherwise algorithmically unreasoned implementation (if they are so lucky as to obtain the source code) of these tools. Further, they should be able to utilize portions of these tools for other projects, such as a source code control system, or a lexical analyzer generator.

Miller's book does have its shortcomings, mostly in the area of typesetting. The source code listings are poorly set, without appropriate attention to language keywords (if, while, for etc.), or comments. These could have been highlighted to improve readability, but it appears that some publishers still think program source is not as important as the actual text; a big mistake in this type of a book.

The text itself could be better organized. Important references and additional readings are buried within the text, or scattered across programming assignments. An additional "Readings" section at the end of each chapter (as in the Software Tools books) would have been much more convenient for the reader. These minor issues do not undermine the fact that A Software Tools Sampler will make a valuable addition to to your library, close to the Software Tools books of Kernighan and Plauger, The UNIX Programming Environment of Kernighan and Pike, and The C Programming Language of Kernighan and Richie.

Some additional readings

 Dependency-Based File Updating


 File Comparison Algorithms and Applications


---

4 In a paper by E. W. Myers and Webb Miller, this algorithm is shown to be near optimal for certain update problems.


**Pattern Matching and Regular Expressions**


Thompson, K., "Regular expression search algorithm," CACM, 11(6), 1968.

**Screen Update Algorithms**


Myers, Eugene W. and Webb Miller, "Row Replacement algorithms for screen editors," TR 86-19, Department of Computer Science, The University of Arizona, Tucson, Arizona.


---

**French UNIX User's Group Conference**

**Paris**

**March 7-10, 1988**

The French Association of UNIX Users (AFUU) in cooperation with the Bureau International de Relations Publiques is organizing a conference in Paris, 7-10 March 1988. There will be tutorials on the first day and technical meetings and an exhibition running concurrently the next three days.

The chair of the Program Committee is Christophe Binot. Information is available from:

AFUU c/o SUPELEC  
Attn.: Anne Garnery, Convention UNIX 88  
Plateau de Moulon  
91190 Gif-sur-Yvette  
FRANCE

mcvax!inria!afuu!anne

16 January/February 1988 Volume 13, Number 1
EUUG Spring 1988 Conference
London
April 11-15, 1988

The UKUUG will host the Spring '88 European UNIX systems User Group Technical Conference at the Queen Elizabeth II Conference Center in London. Technical tutorials will be held on April 11 & 12, followed by the three day conference.

For further information, contact the EUUG Secretariat at the address below.

Call for Papers – EUUG Autumn Conference
Portugal
October 3-7, 1988

The Autumn '88 European UNIX systems User Group Technical Conference will be held in southern Portugal. Technical tutorials will be held on October 3 & 4, followed by the three day conference.

The theme of the conference is “New Directions for UNIX.” The EUUG invites abstracts from those wishing to present their work. Submissions from students are particularly encouraged under the EUUG Student Encouragement Scheme, details of which are available from the EUUG Secretariat. All submitted papers will be refereed. Abstracts must be submitted by post to the EUUG Secretariat.

The Programme Chair will be pleased to provide advice to potential speakers.

Deadlines are:
- Receipt of abstract: 30 April
- Acceptance notification: 15 May
- Final paper received: 1 August

Those interested in offering a tutorial should contact the EUUG Tutorial Officer as soon as possible.

For further information about this and future EUUG events, contact the Secretariat.

Secretariat
EUUG
Owles Hall
Owles Lane
Buntingford, Herts. SG9 9PL
United Kingdom
Phone: (+44) 763 73039
Fax: (+44) 763 73255 (G2)
Email: euug@inset.uucp

Tutorial Officer
Neil Todd
IST
60 Albert Court
Prince Consort Road
London SW7 2BH
United Kingdom
Phone: (+44) 1 581 8155
Fax: (+44) 1 581 5147 (G3)
Telex: 928476 ISTECH G
Email: neil@ist.co.uk

Programme Chair
Peter Collinson
Computing Laboratory
University of Kent
Canterbury, Kent CT2 7NF
United Kingdom
Phone: (+44) 227 764000, x7619
Email: pc@ukc.ac.uk
UUNET Progress Report

There have been a number of inquiries as to the status and future of the UUNET experiment. I think it is appropriate to stop referring to UUNET as a project or experiment and to just call it UUNET.

UUNET currently has nearly 250 subscribers. The Sequent B21K has been purchased and a long-term contract has been entered into with Tymnet. UUNET also has four 800 numbers.

UUNET information is available from
{uunet,ucbvax,decvax}@usenix!uunet-request
or from Madeleine McCall at 415-528-8649.

– PHS

2.10BSD Software Release Available

The “Second Berkeley Software Distribution” (2.10BSD), produced by the Computer Systems Research Group (CSRG) of the University of California, Berkeley, is being distributed by the USENIX Association. It is available to all V7, System III, System V, and 2.9BSD licensees for a price of $200. The release consists of two 2400 foot, 1600 BPI tapes (approximately 80Mb) and approximately 100 pages of documentation. If you require 800 BPI tapes, please contact USENIX for more information.

Sites wishing to run 2.10BSD should be aware that the networking is only lightly tested, and that certain hardware has yet to be ported. Contact Keith Bostic at the address below for current information as to the status of the networking. As of August 6, 1987, the complete 4.3BSD networking is in place and running, albeit with minor problems. The holdup is that only the Interlan Ethernet driver has been ported, as well as some major space constraints. Note, if we decide to go to a supervisor space networking, 2.10 networking will only run on:

11/44/53/70/73/83/84
11/45/50/55 with 18 bit addressing

If you have questions about the distribution of the release, please contact USENIX at:

2.10BSD
USENIX Association
P.O. Box 2299
Berkeley, CA 94710
+1 415 528-8649
{uunet,ucbvax}@usenix!office

If you have technical questions about the release, please contact Keith Bostic at:

{ucbvax,seismo}@keith
keith@okeeffe.berkeley.edu
+1 415 642-4948

Keith Bostic
Casey Leedom
Future Meetings

USENIX 1988 Winter Conference and UniForum – Dallas

The USENIX 1988 Winter Conference will be held on February 9-12, 1988, at the Grand Kempinski Hotel in Dallas, Texas. It will be concurrent with UniForum 1988, which will also be in Dallas.

AFUU UNIX Convention 88
Paris, March 7-10, 1988

EUUG Spring Conference
London, April 11-15, 1988

AUUG Spring Conference
Melbourne, September 13-15, 1988

EUUG Autumn Conference
Portugal, October 3-7, 1988

USENIX 1988 Summer Conference and Exhibition – San Francisco

The USENIX 1988 Summer Conference and Exhibition will be held on June 20-24, 1988, at the Hilton Hotel in San Francisco, California. There will be a conference, tutorials, and vendor exhibits.

Long-term USENIX Conference Schedule

Feb 9-12 '88 Grand Kempinski, Dallas
Jun 20-24 '88 Hilton Hotel, San Francisco
Jan 31-Feb 3 '89 Town & Country Inn, San Diego
Jun 12-16 '89 Hyatt Regency, Baltimore
Jan 22-26 '90 Washington, DC
Jun 11-15 '90 Marriott Hotel, Anaheim
Jan 22-25 '91 Dallas
Jun 10-14 '91 Opryland, Nashville

Publications Available

The following publications are available from the Association Office. Prices and overseas postage charges are per copy. California residents please add applicable sales tax. Payments must be enclosed with the order and must be in US dollars payable on a US bank.

Conference and Workshop Proceedings

<table>
<thead>
<tr>
<th>Meeting</th>
<th>Location</th>
<th>Date</th>
<th>Price</th>
<th>Overseas Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Air Surface</td>
</tr>
<tr>
<td>C++ Workshop</td>
<td>Santa Fe</td>
<td>November '87</td>
<td>$15</td>
<td>$25 $5</td>
</tr>
<tr>
<td>Graphics Workshop IV</td>
<td>Cambridge</td>
<td>October '87</td>
<td>$10</td>
<td>$15 $5</td>
</tr>
<tr>
<td>USENIX</td>
<td>Phoenix</td>
<td>Summer '87</td>
<td>$20</td>
<td>$25 $5</td>
</tr>
<tr>
<td>USENIX</td>
<td>Wash. DC</td>
<td>Winter '87</td>
<td>$20</td>
<td>$25 $5</td>
</tr>
<tr>
<td>Graphics Workshop III</td>
<td>Monterey</td>
<td>December '86</td>
<td>$10</td>
<td>$15 $5</td>
</tr>
<tr>
<td>USENIX</td>
<td>Atlanta</td>
<td>Summer '86</td>
<td>$10</td>
<td>$25 $5</td>
</tr>
<tr>
<td>Graphics Workshop I</td>
<td>Monterey</td>
<td>December '84</td>
<td>$3</td>
<td>$7 $5</td>
</tr>
</tbody>
</table>
4.3BSD Manuals

The USENIX Association now offers all members of the Association the opportunity to purchase 4.3BSD manuals.1

The 4.3BSD manual sets are significantly different from the 4.2BSD edition. Changes include many additional documents, better quality of reproductions, as well as a new and extensive index. All manuals are printed in a photo-reduced 6"x9" format with individually colored and labeled plastic “GBC” bindings. All documents and manual pages have been freshly typeset and all manuals have “bleed tabs” and page headers and numbers to aid in the location of individual documents and manual sections.

A new Master Index has been created. It contains cross-references to all documents and manual pages contained within the other six volumes. The index was prepared with the aid of an “intelligent” automated indexing program from Thinking Machines Corp. along with considerable human intervention from Mark Seiden. Key words, phrases and concepts are referenced by abbreviated document name and page number.

While two of the manual sets contain three separate volumes, you may only order complete sets.

The costs shown below do not include applicable taxes or handling and shipping from the publisher in New Jersey, which will depend on the quantity ordered and the distance shipped. Those charges will be billed by the publisher (Howard Press).

Manuals are available now. To order, return a completed “4.3BSD Manual Reproduction Authorization and Order Form” to the USENIX office along with a check or purchase order for the cost of the manuals. You must be a USENIX Association member. Checks and purchase orders should be made out to “Howard Press.” The manuals will be shipped to you directly by the publisher.

<table>
<thead>
<tr>
<th>Manual</th>
<th>Cost*</th>
</tr>
</thead>
<tbody>
<tr>
<td>User’s Manual Set (3 volumes)</td>
<td>$25.00/set</td>
</tr>
<tr>
<td>User’s Reference Manual</td>
<td></td>
</tr>
<tr>
<td>User’s Supplementary Documents</td>
<td></td>
</tr>
<tr>
<td>Master Index</td>
<td></td>
</tr>
<tr>
<td>Programmer’s Manual Set (3 volumes)</td>
<td>$25.00/set</td>
</tr>
<tr>
<td>Programmer’s Supplementary Documents, Volume 1</td>
<td></td>
</tr>
<tr>
<td>Programmer’s Supplementary Documents, Volume 2</td>
<td></td>
</tr>
<tr>
<td>System Manager’s Manual (1 volume)</td>
<td>$10.00</td>
</tr>
</tbody>
</table>

* Not including postage and handling or applicable taxes.

4.2BSD Manuals are No Longer Available

1 Tom Ferrin of the University of California at San Francisco, a former member of the Board of Directors of the USENIX Association, has overseen the production of the 4.2 and 4.3BSD manuals.
4.3BSD Manual Reproduction Authorization and Order Form

This page may be duplicated for use as an order form

USENIX Member No.: __________________
Purchase Order No.: __________
Date: __________________

As a USENIX Association Member in good standing, and pursuant to the copyright notice as found on the rear of the cover page of the UNIX®/32V Programmer's Manual stating that “Holders of a UNIX®/32V software license are permitted to copy this document, or any portion of it, as necessary for licensed use of the software, provided this copyright notice and statement of permission are included,”

I hereby appoint the USENIX Association as my agent, to act on my behalf to duplicate and provide me with such copies of the Berkeley 4.3BSD Manuals as I may request.

Signed: ____________________________
Institution (if Institutional Member): ____________________________________________

Ship to: 
Name: _____________________________
___________________________________
___________________________________
___________________________________
___________________________________
Phone: _____________________________

Billing address, if different:
Name: _____________________________
___________________________________
___________________________________
___________________________________
___________________________________
Phone: _____________________________

The prices below do not include shipping and handling charges or state or local taxes. All payments must be in US dollars drawn on a US bank.

4.3BSD User's Manual Set (3 vols.) _______ at $25.00 each = $________
4.3BSD Programmer's Manual Set (3 vols.) _______ at $25.00 each = $________
4.3BSD System Manager's Manual (1 vol.) _______ at $10.00 each = $________
Total _______ = $________

[ ] Purchase order enclosed; invoice required.
(Purchase orders must be enclosed with this order form.)

[ ] Check enclosed for the manuals: $________
(Howard Press will send an invoice for the shipping and handling charges and applicable taxes.)

Make your check or purchase order out to “Howard Press” and mail it with this order form to:
Howard Press
c/o USENIX Association
P.O. Box 2299
Berkeley, CA 94710

for office use: m.v.: ________ check no.: ________ amt. rec’d: ________
Local User Groups

The USENIX Association will support local user groups by doing an initial mailing to assist the formation of a new group and publishing information on local groups in ;login:. At least one member of the group must be a current member of the Association.

In the Atlanta area there is a group for people with interest in UNIX or UNIX-like systems, which meets on the first Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241
Marc Merlin (404) 442-4772
Mark Landry (404) 365-8108

In the Boulder, Colorado area a group meets monthly at different sites for informal discussions.

Front Range UNIX Users Group
USENIX Association Exhibit Office
5398 Manhattan Circle
Boulder, CO 80303
John L. Donnelly (303) 499-2600
{boulder,usenix}!johnd

A UNIX users group has formed in the Coral Springs, Florida, area.

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

Dallas/Fort Worth UNIX User's Group
Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205
Jim Hummel (214) 522-2324

Michigan!usr/group in the Detroit/Ann Arbor area has dinner meetings with feature presentations and informal discussions the first Wednesday of each month.

Linda Mason (313) 855-4220
michigan!usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

In Fresno, California, the Central California UNIX User's Group consists of an electronic mailing list to which members may post questions or information. Communication is by uucp; for connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 294-4373
Department of Computer Science
California State University
Fresno, CA 93740-0109
CSNET: brent@CSUFresno.edu
uucp: csufres!brent

Commercial institutions or individuals:
Gordon Crumal (209) 435-6062
Harry J. Wilson & Company
Insurance Correspondents, Inc.
2350 W. Shaw Ave, Suite 110
Fresno, CA 93711
uucp: csufres!tower!gordon

The Los Angeles UNIX Group meets on the third Thursday of each month in Redondo Beach, California.

Drew Bullard (213) 535-1980
{ucbvax,ihnp4}!trwrb!bullard

Marc Ries (213) 535-1980
{decvax,scdrdf}!trwrb!ries

In the Melbourne, Florida area the Space Coast UNIX User's Group meets at 8pm on the third Wednesday of each month at the Florida Institute of Technology.

Alex Stover (305) 724-3962
codas!lola!als
Bill Davis (305) 242-4449
bill@ccd.harris.com
In Minnesota a group meets on the first Wednesday of each month.
UNIX User's of Minnesota
Mark Colburn  (612) 935-2688
4732 Spring Circle
Minnetonka, MN 55343
mark@ems.mn.org
ihnp4!meet5!ems!mark

In the northern New England area is a group that meets monthly at different sites.
Emily Bryant  (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755
David Marston  (603) 883-3556
Daniel Webster College
University Drive
Nashua, NH 03063
dcexv!dartvax!hneuug-contact

In the New York City area there is a non-profit organization for users and vendors of products and services for UNIX systems.

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116
Ed Taylor  (212) 513-7777
{attunix,philabs)!pencom!taylor

The New Zealand group provides an annual Workshop and Exhibition and a regular newsletter to its members.
New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

In the Orlando, Florida area the Central Florida UNIX User's Group meets at 7pm on the third Thursday of each month. Programs include informal presentations, demonstrations and discussions.
Mike Geldner  (305) 862-0949
codas!sunfla@mike
Ben Goldfarb  (305) 275-2790
goldfarb@hcx9.ucf.edu

Mikel Manitius  (305) 869-2462
{codas,attmail)!mikel

A group has started in the St. Louis area:
St. Louis UNIX User's Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105
Eric Kiebler  (314) 725-9492
ihnp4!plus5!sluug

The San Antonio UNIX User's (SATUU) maintains communications largely by a local uucp network. They also meet the third Wednesday of each month at 7pm.
William T. Blessum, M.D.  (512) 692-0977
7950 Floyd Curl Dr. #102
San Antonio, TX 78229-3955
gatech,ihnp4)!petrolbles!wtb

The local uucp network “postmaster” is:
Bruce Andreen  (512) 656-3053
gatech,ihnp4)!petrolbruce

The Seattle area there is a group with over 150 members, a monthly newsletter, a software exchange system, and monthly meetings.
Bill Campbell  (206) 232-4164
Seattle UNIX Group Membership Information
6641 East Mercer Way
Mercer Island, WA 98040
uw-beaver!tikall!camco!bill

A UNIX/C language users group has been formed in Tulsa.
Pete Rourke
$USR
7340 East 25th Place
Tulsa, OK 74129

The Washington, D.C., area a group meets the first Tuesday of each month.
Washington Area UNIX User's Group
2070 Chain Bridge Road, Suite 333
Vienna, VA 22180
Samuel Samalin  (703) 448-1908
Conferences and Workshops

Space Out on Troff

Book Review

Nominating Committee Report

---

*Change of Address Form*

Please fill out and send the following form through the U.S. mail to the Association Office at the address above.

Name: _______________________________ _______________________________ Member #: __________

OLD: ________________________________

NEW: ______________________________

Phone: ______________________________

* uucp: {uunet,ucbvax}! ______________________