CONTENTS

Results of the Election .............................................................. 3
Know Your Board and Staff ......................................................... 4
USENIX Online Library/Index ................................................. 5
Call for Participation: Software Development Environments Workshop 6
Call for Papers: Winter 1991 USENIX Conference ......................... 7
Call for Papers: Mach Workshop ............................................. 9
Call for Papers: Large Installation Systems Administration Conference 9
UKUUG Summer 1990 Technical Conference ................................ 10
Long-Term Calendar of UNIX Events ...................................... 12
Book Review
  Life with UNIX – A Guide For Everyone .................................. 13
    Douglas A. Gwyn
Summary of Board of Directors’ Meeting .................................... 14
Report on ISO/IEEE JTC1/SC22/WG15
Rapporteur Group on Internationalization Meeting ...................... 17
    Dominic Dunlop
International Standardization ................................................... 21
    Dominic Dunlop
An Update on UNIX and C Standards Activity ............................ 25
    Jeffrey S. Haemer
CSUUG Established .................................................................... 55
Proceedings Subscription Order Form ....................................... 56
Proceedings Order Form ............................................................ 57
Local User Groups .................................................................... 58

The closing date for submissions for the next issue of ;login: is June 29, 1990.
login: (ISSN 1044-6397) is published bi-monthly by the USENIX Association.

The USENIX Association is a not-for-profit organization of those interested in UNIX\(^1\) and UNIX-like systems. It is dedicated to fostering and communicating the development of research and technological information and ideas pertaining to advanced computing systems, to the monitoring and encouragement of continuing innovation in advanced computing environments, and to the provision of a forum where technical issues are aired and critical thought exercised so that its members can remain current and vital.

The officers of the Association are:

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>President</td>
<td>Deborah K. Scherrer</td>
<td><a href="mailto:scherrer@usenix.org">scherrer@usenix.org</a></td>
</tr>
<tr>
<td>Secretary</td>
<td>Rob Kolstad</td>
<td><a href="mailto:kolstad@usenix.org">kolstad@usenix.org</a></td>
</tr>
<tr>
<td>Treasurer</td>
<td>Stephen C. Johnson</td>
<td><a href="mailto:scj@usenix.org">scj@usenix.org</a></td>
</tr>
<tr>
<td>Directors</td>
<td>M. Kirk McKusick</td>
<td><a href="mailto:mckusick@usenix.org">mckusick@usenix.org</a></td>
</tr>
<tr>
<td></td>
<td>Sharon Murrel</td>
<td><a href="mailto:eowyn@usenix.org">eowyn@usenix.org</a></td>
</tr>
<tr>
<td></td>
<td>Alan G. Nemeth</td>
<td><a href="mailto:agn@usenix.org">agn@usenix.org</a></td>
</tr>
<tr>
<td></td>
<td>Michael D. O'Dell</td>
<td><a href="mailto:mo@usenix.org">mo@usenix.org</a></td>
</tr>
<tr>
<td></td>
<td>John S. Quarterman</td>
<td><a href="mailto:jsq@usenix.org">jsq@usenix.org</a></td>
</tr>
<tr>
<td>Executive Director</td>
<td>Ellie Young</td>
<td><a href="mailto:ellie@usenix.org">ellie@usenix.org</a></td>
</tr>
</tbody>
</table>

USENIX Association Office
2560 Ninth Street, Suite 215
Berkeley, CA 94710
(415) 528-8649
office@usenix.org

The editorial staff of login: is:

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editor</td>
<td>Ellie Young</td>
<td><a href="mailto:ellie@usenix.org">ellie@usenix.org</a></td>
</tr>
<tr>
<td>Managing Editor</td>
<td>Carolyn Carr</td>
<td></td>
</tr>
<tr>
<td>Copy Editor</td>
<td>Michelle Dominijanni</td>
<td><a href="mailto:dominijanni@usenix.org">dominijanni@usenix.org</a></td>
</tr>
<tr>
<td>Production Editor</td>
<td>Tom Strong</td>
<td></td>
</tr>
</tbody>
</table>

Contributions Solicited

Members of the UNIX community are encouraged to contribute articles to login:. Contributions may be sent electronically to login@usenix.org 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 troff and a variation of the -me macros.

UUNET Subscriptions

UUNET Communications
3110 Fairview Park Drive, Suite 570
Falls Church, VA 22042
(703) 876-5050
uunet-request@uunet.uu.net

Acknowledgments

The Association uses a Sun\(^3\) 3/180S running SunOS for support of office and membership functions, preparation of login:, and other Association activities. Connected to the Sun is a QMS Lasergrafix\(^*\) 800 Printer System donated by Quality Micro Systems of Mobile, Alabama. It is used for general printing and draft production of login: and Computing Systems with ditroff software provided by mtXinu. The membership and mailing lists are maintained using Sybase.\(^4\)

This newsletter is for the use of the members 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).
Results of the Election for the
USENIX Association Board of Directors

The results of the elections for Board of Directors of the USENIX Association for the 1990-92 term are as follows:

Elected:

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>President</td>
<td>Marshall Kirk McKusick</td>
<td>472 *</td>
</tr>
<tr>
<td>Vice-President</td>
<td>Michael O'Dell</td>
<td>730 + 112 abstentions</td>
</tr>
<tr>
<td>Secretary</td>
<td>Rob Kolstad</td>
<td>716 + 126 abstentions</td>
</tr>
<tr>
<td>Treasurer</td>
<td>Sharon Murrel</td>
<td>725 + 117 abstentions</td>
</tr>
</tbody>
</table>

Directors:
- Ed Gould: 468
- Rick Adams: 426
- Evi Nemeth: 380
- Barry Shein: 360

Not elected:

For President: Stephen C. Johnson: 347 *

For Director:
- Sonya Neufer: 317
- Daniel Geer: 291
- Daniel Klein: 280
- Peter Collinson: 261
- Max Vasilatos: 242
- Dave Taylor: 232

Total number of ballots cast: 838 (6 invalid ones)

* 13 abstentions (people abstaining for both candidates for President)
There were 6 invalid ballots.

Ellie Young
Executive Director

May/June 1990
KNOW YOUR BOARD and STAFF

M. Kirk McKusick
President

Michael O’Dell
Vice President

Sharon Murrell
Treasurer

Rob Kolstad
Secretary

Rick Adams
Ed Gould
Evi Nemeth
Barry Shein

DIRECTORS

STAFF

Ellie Young
Executive Director

Carolyn Carr
Publications Manager

Judy DesHarnais
Conference Coordinator

John Donnelly
Tutorial/Exhibit Manager

Toni Veglia
Receptionist

Andrea Galleni
Office Manager

Marilynn Allemann
Conference Associate

Michael McLaughlin
Exhibit Assistant
USENIX Online Library/Index

What Is It:

The USENIX online index is an electronically available list of papers published by the USENIX Association and related groups. It contains title, author, and related information about papers published in USENIX and UNIX-related conference and workshop proceedings, newsletters, journals, and the like.

The index is freely available, and is kept as a simple ASCII file, in refer/bib format, sorted by author. In some cases, electronically readable versions of full papers or abstracts are also available. If a paper is available online, this is indicated in its index entry.

How to Get the Index:

The index is available online from UUNET, either via a mail server or anonymous ftp. The index is about 200K, and available only in entirety. To get it via electronic mail:

$ echo send bibliography | mail uunet!library

A (non-human) server will automatically break the index up into mailable chunks (if necessary), and return it to the sender of the mail.

Or, the index can be retrieved via anonymous ftp to uunet.uu.net:

ftp> get library/bibliography my_local_file

To get a help file:

$ echo help | mail uunet!library

To pick up the date the index was last changed:

$ echo send date | mail uunet!library

For those unable to reach UUNET, the index is also available in hardcopy format from the Association office.

Online Papers and Abstracts:

We are actively soliciting the donation of papers and abstracts to include in the library. If you have had a paper published in any of the publications listed below, and you wish to donate the paper, you must provide us with an electronic version and give us permission to distribute it. You or your employer may retain the copyright if you wish.

If you wish to donate an abstract, we are prepared to type it in for you – all we need is your permission.

Publications Indexed:

Currently we have indexed all available issues of the following:

USENIX:
- Conference proceedings
- Workshop proceedings
- Computing Systems

European UNIX User Group:
- Conference proceedings
- Newsletters

Software Tools User Group:
- Conference proceedings

Australian UNIX User Group:
- Newsletters

UNIX Review periodical

We are in the process of incorporating Japanese UNIX Society publications to the index. Other sources (AFUU, GUUG, NZUSUGI, etc.) are being continually evaluated and will be included as deemed suitable.

More Information:

For additional information about the online index and library, and/or instructions for donating abstracts or papers, contact:

usenix@index (index@usenix.org)

Or contact the Association's executive office.

---

USENIX Supporting Members

Supporting Membership is open to any individual or institution that wants to support the Association to a greater degree. The following organizations are 1990 Supporting Members:

Aerospace Corporation
AT&T Information Systems
Convex Computer Corporation
Digital Equipment Corporation
mt Xinu
Open Software Foundation
Quality Micro Systems
Sun Microsystems, Inc.
Sybase, Inc.

May/June 1990
Call for Participation:
Software Development Environments Workshop

Hotel Grand Kempinski, Dallas, Texas, USA, January 16-18, 1991
Co-sponsored by: USENIX Association (USA) and SIGMA Project (Japan)

Many software development environments have been described, built, or used which are intended to operate atop the UNIX system. The goal of this workshop is to share information on what these systems look like, what problems were solved by using UNIX and what problems were caused by it. We expect strong representation from the Japanese SIGMA workstation project, which defines a national software engineering environment that uses UNIX as its base, as well as American and European academic and industrial organizations.

Participants will be selected by a program committee on the basis of submitted position papers. Attendance will be limited to 75 to encourage discussion. Meetings will include descriptions of important systems and presentations on particular technical points involving implementation and usage. Significant time will be set aside for panels and informal discussions of such topics.

Position papers will be evaluated by a program committee including researchers and practitioners from Europe, Japan, and the United States. Please send a 1-4 page position paper by August 1, 1990 to one of the co-chairs:

Noboru Akima
Sigma Project
5th Akihabara Sanwa Bank Building
3-16-8 Soto-Kanda, Chiyoda-ku
Tokyo, Japan 101
USA

Stuart Feldman
Bellcore
445 South Street
Morristown, NJ 07962-1910
USA

Electronic versions may be mailed to sdeconf@bellcore.com.

Relevant topics that might be addressed in the position paper include:

- description of a significant system (by a designer or builder)
- experience with using such a system
  - novel tools or facilities offered by such a system
  - evaluations of usage
  - positive and negative experiences
- experience with building such a system
  - architectural considerations for a UNIX-based Software Development Environment
  - advantages resulting from basing the system on UNIX
  - problems (and solutions) encountered in designing and implementing such a system
    (e.g., file and database systems, networking and cooperation, scheduling and resource usage)

Workshop Format
The structure of the workshop will be decided after participants are selected. A likely agenda is:

Wednesday: Descriptions of SIGMA project, descriptions of other large systems,
  discussion of technical difficulties, group reception, and Birds of a Feather sessions

Thursday: Panels on topics arising on Wednesday, subgroups, Birds of a Feather sessions

Friday: Subgroup reports, panels, debate, wrapup
Call for Papers: Winter 1991 USENIX Conference

Dallas, Texas, January 21-25, 1991

USENIX seeks original papers which describe new and interesting work for the Winter 1991 Technical Conference. Papers which are accepted for this conference will be published in the conference proceedings and will be presented during the three days of technical sessions.

The previous conference had a theme which was retrospective in nature, so for this conference we look to the future. We would like to include papers that emphasize changes to operating systems and environments as we know them today. Thus, the theme is:

What's next: by the year 2010, evolution or revolution? UNIX derivative or something else?

Appropriate topics include, but are not limited to:

Operating systems of the future:
- Distributed Systems
- Real-time systems
- Object-Oriented systems
- Fault Tolerant Systems
- Multiprocessor and Multicomputer Systems
- Workstation Systems
- Systems for Novel Architectures

Communications and Networking:
- Protocols
- Performance
- Administration
- Security

Applications:
- Databases
- Transaction Processing
- Arts and Social Applications
- Novel Application Areas

User Interfaces
- Human Factors
- Graphics and Window Systems
- Graphical User Interfaces

Programming Environments and Languages

Testing and Debugging

We will require at least an abstract and an outline in a form that gives the committee confidence in the final paper.

A submission should be 2-3 typewritten pages and include the following:

1. Author names, addresses, telephone numbers, and E-mail addresses.
2. Abstract: 100-300 words (half a page) to be included in the final paper.
3. Outline: 1.5-3 pages, giving the major headings of the paper, plus a few sentences per section that give the major points that will be covered in that section in the final paper.
4. References: List a few key references to other work on the topic, preferably to other people's work.

The following is a sample outline, which is not necessarily appropriate for all papers, but which illustrates the important topics.

1. Introduction
   Background
   Introduce the problem to be solved; why is it important?
   Reference previous work; make sure the committee knows the wheel is not being reinvented

2. How We Solved the Problem
   More details on the problem and its issues
   Design decisions and tradeoffs, and why they were made
   Implementation issues

3. Evaluation
   Data, on performance, effort required
   How well does it work?
   What would we do differently?
   If it failed, why? and what can we learn from it?

4. Conclusion
   Summarize the paper, emphasizing why it is important, and what was learned
5. References

Please submit abstracts with outlines as soon as possible, and mail one hard copy and one electronic copy to the addresses below. The final deadline for receipt of submissions is **August 13, 1990**. Abstracts received after this deadline will not be considered. Notification of acceptance or rejection will be made by October 3, 1990. Final camera-ready papers are due by November 14, 1990.

The final paper should retain the 100-300 word abstract, add illustrations (where needed), and citations to relevant literature. Only previously unpublished submissions will be considered. Final papers should contain 8-12 pages of single spaced typeset materials. All final papers must be submitted in a camera-ready format or electronic format (`troff` `-ms` if possible). Typewritten or dot-matrix output is not acceptable. For authors without access to a laser printer or typesetter, appropriate facilities will be provided by the program chair.

Please send the hard copy of your submission to:

Lori S. Grob  
Dallas Conference  
USENIX Association  
2560 9th Street, Suite 215  
Berkeley, CA 94710

To request additional information, please contact:

Lori S. Grob  
Dallas USENIX Technical Program  
Chorus systèmes  
6, avenue Gustave Eiffel  
F78182 Saint-Quentin-en-Yvelines CEDEX  
France

Internet: dallas-conf@usenix.org  
UUCP: uunet!usenix!dallas-conf  
Telephone: +33 (1) 30 57 00 22  
FAX: +33 (1) 30 57 00 66

Please include your physical and electronic mail address in all correspondence.

Program Committee:

Lori S. Grob, Chair, Chorus systèmes  
Steve Bourne, Sun Microsystems  
Marc Donner, IBM Research  
Tom Duff, AT&T Bell Laboratories  
Jan Edler, New York University  
Michel Gien, Chorus systèmes  
Barry Gleeson, Unisys Corp.  
Trent R. Hein, University of Colorado, Boulder  
Andrew Hume, AT&T Bell Laboratories  
Michael J. Karels, University of California, Berkeley  
Deborah K. Scherrer, mt Xinu  
Melinda Shore, mt Xinu  
Max Meredith Vasilatos, Open Software Foundation
Call for Papers: Mach Workshop
Radisson Hotel, Burlington, VT, October 4-5, 1990

The use of Mach in what has traditionally been the UNIX community is growing as DARPA and OSF increase their Mach-related activities and more vendors are supporting Mach on a variety of platforms. Because Mach itself is changing rapidly and there hasn't been any convenient mechanism for communication among developers, the USENIX Association is pleased to sponsor its first Mach workshop, in which researchers, vendors, and users can share results of Mach-related development work and status reports on work-in-progress.

The workshop will be oriented towards those who have actually worked with Mach or have done Mach-based applications development, and will not be tutorial in nature. The program will consist largely of refereed papers and panels. Abstracts of 350-700 words should be sent to the program chair at the address below (those submitting hardcopy abstracts should send five copies). The deadline for submissions is June 22, 1990. All submissions will be acknowledged. Authors will be notified by July 20, 1990, and full papers will be required by Aug. 27, 1990.

For further information about the workshop, contact the program chair:
Melinda Shore (415) 644-0146
mt Xinu shore@mtxinu.com
2560 Ninth St., Suite 312
Berkeley, CA 94710

Program Committee:
Alan Langerman, Encore Computer Corp.
Douglas Orr, Carnegie-Mellon University
Homayoon Tajalli, Trusted Information Sys.
Avadis Tevanian, NeXT, Inc.

Call for Papers: Large Installation Systems Administration Conference
Colorado Springs, CO, October 17-19, 1990

The Fourth USENIX Large Installation Systems Administration Conference will be held in Colorado Springs, Colorado on October 18-19, 1990. A tutorial program will be offered in conjunction with the conference on October 17.

The program committee will be reviewing papers submitted on subjects including but not limited to:

Automation of tasks
Network management
Distributed services
System backup
File and data archiving
Electronic mail
Security
Account/user management
Accounting
USENET News/Notes
Performance monitoring and tuning
Configuration management
Vendor issues
Distributed administration

We are especially interested in papers which provide freely available or fully described solutions to existing problems, or which in some way advance the state of the art. Administration of installations which are "unique" in any fashion (size, hardware, number of users, security level, etc.) is also of special interest.

Papers should be from 5 to 15 pages in length, including diagrams, figures, etc. Papers should include a brief description of the site, an outline of the problem and issues, and a description of the solution. We prefer, but do not require electronic form, e.g., nroff/troff, TeX, Postscript, etc. The deadline for submission of papers is July 25, 1990.

Workshop proceedings will be distributed to all the attendees and are also available after the Conference from the USENIX Association.

For further information about the conference, contact the program chair:
Steve Simmons 313-769-4086
Industrial Technology Institute scs@iti.org
2901 Hubbard Road
Ann Arbor, MI 48109
UKUUG Summer 1990 Technical Conference


The UKUUG Summer 1990 Technical Conference will take place at the Royal Lancaster Hotel in London on July 11-13, preceded by two days of advanced tutorials beginning on Monday July 9. The technical conference program is listed below. A pre-conference registration booklet containing detailed information is available from:

UKUUG
Owles Hall
Buntingford
Herts SG9 9PL UK

Tel: +44 (763) 73039
Fax: +44 (763) 73255
Email: ukuug-conf@ukc.ac.uk

Wednesday July 11, 1990

Opening Address: Sunil K. Das, UKUUG Chairman and Programme Director
City University London Computer Science Department, UK

Keynote Speech: Rob Pike; AT&T Bell Laboratories, New Jersey, USA
Plan 9 from Bell Labs

David L. Presotto; AT&T Bell Laboratories, New Jersey, USA
Multi-Processor Streams for Plan 9 and UNIX

Tom Duff; AT&T Bell Laboratories, New Jersey, USA
c - A Shell for Plan 9 and UNIX

Dennis M. Ritchie; AT&T Bell Laboratories, New Jersey, USA
Variable-Size Arrays in C

Ken Thompson; AT&T Bell Laboratories, New Jersey, USA
A New C Compiler

Tom A. Cargill; Independent Consultant, Colorado, USA
Does C++ Really Need Multiple Inheritance?

Thomas J. Bannon; University of Texas at Dallas, USA
GROUP: A Participants Manager for Multi-User Applications

Jim Reid; Strathclyde University, Glasgow, UK
NFS: The Protocol is the Problem

UKUUG Business Meeting, BoFS and WiPS

Thursday July 12, 1990

Ciaran O'Donnell; O'Donnell, Palaiseau, France
rcc – An Optimising C Compiler for the Motorola 88000

Greg Rose; Softway Pty Ltd, Sydney, Australia
Limits: A System for Resource Management under UNIX

Marshall Kirk McKusick, Michael J. Karels, Keith Bostic; University of California, Berkeley, USA
A Pageable Memory Based File System

Chris Torek; University of Maryland, USA
A New Framework for Device Support in Berkeley UNIX
Jon L. Bentley; AT&T Bell Laboratories, New Jersey, USA
Pictures of Programs

Michael Hawley; MIT Media Laboratory, Massachusetts, USA
Symphonic Emulation and the Ultravirtuoso

Daniel V. Klein; SEI – Carnegie Mellon University, Pittsburgh, USA
Foiling the Cracker: A Survey of and Improvements to Password Security

Greg Rose; Softway Pty Ltd, Sydney, Australia
UNIX and Supercomputers

C. H. Forsyth; University of York, UK
More Taste: Less Greed? – Sending UNIX to the Fat Farm

Robert Swartz; Mark Williams Company, Illinois, USA
The Case Against UNIX Standards

Peter H. Salus; OTF – Open to oFfers, Boston, USA
Standards, Specifications and Open Systems

A. N. Other; UNIX International
The Case For UNIX Standards

Standards Panel and BoFS

Friday July 13, 1990

Laszlo Biczok, Zoltan Dioszeghy and Kalman Szeker;
Central Research Institute, Budapest, Hungary
XEUS: An Intelligent Terminal System

Martin D. Beer; University of Liverpool, UK
Developing Document Management Systems Using the Andrew Toolkit

Andrei G. Yaneff, Trevor I. Fenner; University of London, UK
G3 – A Language for Typesetting Three Dimensional Graphics

Brian O’Donovan, Jane Grimson; Trinity College, Dublin, Ireland
Development of a Distributed Revision Control System

Stuart I. Feldman; Bell Communications Research, New Jersey, USA
Large Scale Software Development Under UNIX

Brian W. Kernighan; AT&T Bell Laboratories, New Jersey, USA
The UNIX System and Software Productivity

Ken Dove; Sequent Computer Systems, Oregon, USA
A High Capacity TCP/IP in Parallel Streams

Phil Winterbottom and Tim Wilkinson; City University London, UK
Meshix: A UNIX-like Operating System for Distributed Machines

A. W. Morris; University of Cambridge, UK
Project Grantia – The Cambridge University FDDI Network

Andrew S. Tanenbaum, Robbert van Renesse, Hans van Staveren, Gregory J. Sharp;
Vrije Universiteit, Amsterdam, The Netherlands
Beyond UNIX – A True Distributed System for the 1990s
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990 Jul 9-11</td>
<td>15th JUS Symposium</td>
<td>Tokyo, Japan</td>
</tr>
<tr>
<td></td>
<td>UKUUG</td>
<td>London, UK</td>
</tr>
<tr>
<td>1990 Jul 11-13</td>
<td>UNIX Sys Software Tech. Seminar</td>
<td>ACM, Dallas, TX</td>
</tr>
<tr>
<td>1990 Jul 12-13</td>
<td>IEEE 1003</td>
<td>Boston, MA</td>
</tr>
<tr>
<td>1990 Aug 6-10</td>
<td>SIGGRAPH</td>
<td>Portland, OR</td>
</tr>
<tr>
<td>1990 Aug 20-23</td>
<td>Interex</td>
<td>Cannes, France</td>
</tr>
<tr>
<td>1990 Aug 27-28</td>
<td><em>Security</em></td>
<td>Wiesbaden, Germany</td>
</tr>
<tr>
<td>1990 Sep 3-7</td>
<td>DECUS Europe Symposium</td>
<td>World Congress Centre, Melbourne, Aust.</td>
</tr>
<tr>
<td>1990 Sep 4-5</td>
<td>GUUG</td>
<td>Burlington, VT</td>
</tr>
<tr>
<td>1990 Sep 25-28</td>
<td>AUUG</td>
<td>San Jose, CA</td>
</tr>
<tr>
<td>1990 Oct 4-5</td>
<td><em>Mach</em></td>
<td>Seattle, WA</td>
</tr>
<tr>
<td>1990 Oct 8-12</td>
<td>InterOp 90 ACE</td>
<td>Colorado Springs, CO</td>
</tr>
<tr>
<td>1990 Oct 15-19</td>
<td>IEEE 1003</td>
<td>Nice, France</td>
</tr>
<tr>
<td>1990 Oct 22-26</td>
<td>EUUG</td>
<td>Tokyo, Japan</td>
</tr>
<tr>
<td>1990 Oct 31-Nov 1</td>
<td>UNIX Expo</td>
<td>Dec 17-19</td>
</tr>
<tr>
<td>1990 Nov 5-9</td>
<td>Computer Communication Conf.</td>
<td>San Jose Fairmont/SJ Conference Center</td>
</tr>
<tr>
<td>1990 Nov 8</td>
<td>Open Systems, NLUUG</td>
<td>Tokyo, Japan</td>
</tr>
<tr>
<td>1990 Nov 14-16</td>
<td>UNIX EXPO '90 UniForum</td>
<td>Grand Kempinski, Dallas, TX</td>
</tr>
<tr>
<td>1990 Nov 15</td>
<td>POSIX APP Workshop</td>
<td>Grand Kempinski, Dallas, TX</td>
</tr>
<tr>
<td>1990 Nov 15-16</td>
<td>16th JUS Symposium</td>
<td>Infomart, Dallas, TX</td>
</tr>
<tr>
<td>1990 Dec 2-6</td>
<td>Sun Users Group</td>
<td>Atlanta Hilton &amp; Towers</td>
</tr>
<tr>
<td>1990 Dec 4-5</td>
<td>JUS UNIX Fair '90</td>
<td>Liverpool, UK</td>
</tr>
<tr>
<td>1990 Dec 10-12</td>
<td>UNIX Asia '90</td>
<td>Budapest, Hungary</td>
</tr>
<tr>
<td>1990 Dec 10-14</td>
<td>DECUS</td>
<td>Edinburgh, UK</td>
</tr>
<tr>
<td>1990 Dec 17-19</td>
<td>UKUUG</td>
<td>San Jose Fairmont/SJ Conference Center</td>
</tr>
<tr>
<td>1991 Jan 7-11</td>
<td>IEEE 1003</td>
<td>Hilton Square, San Francisco, CA</td>
</tr>
<tr>
<td>1991 Jan 16-18</td>
<td>*Software Devel. Environments</td>
<td>Moscone Center, San Francisco, CA</td>
</tr>
<tr>
<td>1991 Jan 21-25</td>
<td>USENIX</td>
<td>Jersey, UK</td>
</tr>
<tr>
<td>1991 Jan 22-25</td>
<td>UniForum</td>
<td>Atlanta, GA</td>
</tr>
<tr>
<td>1991 Feb</td>
<td>UNIX in Government</td>
<td>Tromso, Norway</td>
</tr>
<tr>
<td>1991 Feb 18-22</td>
<td>DECUS</td>
<td>Opryland, Nashville, TN</td>
</tr>
<tr>
<td>1991 April</td>
<td>IEEE 1003</td>
<td>Atlanta Hilton &amp; Towers</td>
</tr>
<tr>
<td>1991 May</td>
<td>UNIX 8x/etc</td>
<td>Liverpool, UK</td>
</tr>
<tr>
<td>1991 May 6-10</td>
<td>DECUS</td>
<td>Budapest, Hungary</td>
</tr>
<tr>
<td>1991 May 20-24</td>
<td>EUUG</td>
<td>Edinburgh, UK</td>
</tr>
<tr>
<td>1991 Jun 10-14</td>
<td>USENIX</td>
<td>San Jose Fairmont/SJ Conference Center</td>
</tr>
<tr>
<td>1991 Jun 16-19</td>
<td>Sun Users Group</td>
<td>Hilton, CA</td>
</tr>
<tr>
<td>1991 Jun/Jul</td>
<td>UKUUG</td>
<td>Moscone Center, San Francisco, CA</td>
</tr>
<tr>
<td>1991 Sep 16-20</td>
<td>EUUG</td>
<td>Jersey, UK</td>
</tr>
<tr>
<td>1991 Dec</td>
<td>UKUUG</td>
<td>Atlanta, GA</td>
</tr>
<tr>
<td>1991 Dec 8-11</td>
<td>Sun Users Group</td>
<td>Marriott, San Antonio, TX</td>
</tr>
<tr>
<td>1991 Dec 9-13</td>
<td>DECUS</td>
<td>Washington (DC) Hilton &amp; Towers</td>
</tr>
<tr>
<td>1992 Jan 20-24</td>
<td>USENIX</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 Jan 21-24</td>
<td>UniForum</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 Spring</td>
<td>EUUG</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 May 4-8</td>
<td>DECUS</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 Jun 8-12</td>
<td>USENIX</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 Jun 21-24</td>
<td>Sun Users Group</td>
<td>Amsterdam, Netherlands</td>
</tr>
<tr>
<td>1992 Autumn</td>
<td>EUUG</td>
<td>Amsterdam, Netherlands</td>
</tr>
</tbody>
</table>

† Compiled with the assistance of Alain Williams of the EUUG, Susanne Smith of Windsound Consulting and John Quarterman of Texas Internet Consulting.

* USENIX Workshops
Book Review

Life with UNIX – A Guide For Everyone
Don Libes & Sandy Ressler

Reviewed by Douglas A. Gwyn
Gwyn@BRL.Mil

This book should answer the vast majority of questions that a thoughtful user of the UNIX operating system is bound to have concerning aspects of UNIX that are not addressed by reference manuals and “how to” tutorials. This is important information, since UNIX is not only an operating system but also a philosophy of computing, a set of traditions, an active subculture, and a major market force. Most of the material in Life with UNIX is not readily available from other sources, making this an essential reference volume for a well-rounded UNIX library.

Written in an informal style, Life with UNIX tells you everything you ever wanted to know about UNIX and even things you didn’t know you should ask about, among them: UNIX evolution and politics of its development; versions of UNIX and portability issues; UNIX licensing and the UNIX-based systems market; standards, changing technologies, and the future of UNIX; sources of printed information; tools and the use of the shell environment; C, system programming, and programming support tools; system administration and system (in)security; Usenet, public-domain software, and games; benchmarking, consulting, mailing lists, validation, and typesetting services; UNIX applications; and databases, emulators, internationalization, networks, parallel processing, real-time processing, and workstations. Also useful are the listings of conferences, workshops, courses, and user groups.

This guide includes a “Who’s Who” listing of significant names in the UNIX community, brief reviews of UNIX-related books and periodicals, addresses for numerous organizations, and an excellent comprehensive index that helps locate answers to such vexing questions as “What is the NUXI problem, anyway?” There are also many items that readers may find entertaining. These include quotations, anecdotes, and descriptions of some of the ways the UNIX community has fun, such as the P1003 WeirdNIX competition and the annual International Obfuscated C Code Contest.

Life with UNIX is quite an accomplishment. I noticed only two nontrivial flaws: It is riddled with minor inaccuracies, roughly one per page. While these do not detract from its use as a significant source of information about the UNIX phenomenon, they do render it unsuitable as a primary reference source and for settling “bar bets.” I also feel that its attempts to foretell the future, especially its pronouncements about “the way things should be,” do not measure up to the quality of the rest of the book. For instance, the authors speak approvingly of interfaces like the Macintosh Finder replacing the traditional UNIX shell, presumably as one step toward turning computers into appliances. Improved user interfaces are undoubtedly possible, but the Finder is not sufficiently “programmer friendly.” What made UNIX great was that it was designed by skilled programmers for use by them, e.g., enabling programs to support further programs, thereby obtaining tremendous leverage.

This book should be a prerequisite for posting questions to the Usenet newsgroup comp.unix.questions, because it answers nearly all the obvious questions about UNIX.
Summary of Board of Directors’ Meeting
Berkeley, CA, April 6, 1990

The regular quarterly meeting of the USENIX Board of Directors was convened at the Claremont Hotel in Berkeley, CA on April 6, 1990.


Action Items

DesHamais said that 34 attendees had requested child care information at D.C. and only one utilized the service.

D.C. '90 Conference

DesHamais reported that there were 1,485 attendees, and that perhaps being in Baltimore the previous June and having less room for tutorials at the Shoreham affected attendance. Everyone agreed that the overall evaluation of the conference was good.

C++ Conference

DesHamais reported that 492 people had registered to date, and the two half-day tutorials on Macs were under-enrolled. There was a discussion regarding tutorial speakers’ expenses and it was suggested that we do an analysis of all speakers’ expenses. Young was asked to find a chair to present a proposal to hold another conference next year.

1994 Winter Conference

Young informed the board the UniForum’s dates had changed to February 14-18, and it was decided to move the USENIX conference back to the week of January 17 in order to have enough time between the Winter and Summer meetings.

1994 Summer Conference

Proposals from San Francisco, Anaheim and Boston were reviewed, and it was decided to hold this conference in Boston. Even though the cost was higher there, we felt we needed to serve the east coast members by holding a conference in that area.

Complimentary Registrations at Conferences

Young asked for guidance from the Board as to whether the present policy reflects an accurate picture. She pointed out that a complimentary registration at Anaheim will have an incremental cost of $110 per. Everyone agreed that we are comfortable with the present number of comps. It was decided that the organizer of the alternate track be given the same perks as the technical program chair, and that speakers for that track be given comp registration.

Distributed & Multiprocessor Systems II Symposium

The proposal from Spafford and Leach was accepted with an amendment to strike the words “attendance limited to 250.” The board felt that limiting registration in this way would cause difficulty for members, and suggested that attendance be open.

Commemorative Book Proposal

It was decided that we would not fund Katz’s proposal to produce a commemorative publication that would include history, anecdotes, memorabilia, and FaceSaver pictures.

Professional Development Seminars

The Houston seminar netted $4,700 with 40-42 people attending each tutorial. Donnelly felt that it was more successful because of considerable support and assistance from the Houston UNIX Users Group (Hounix). It was agreed that we should go ahead with another one in the Fall.
Speakers Bureau

Donnelly passed out a draft of a flier that would be mailed to universities, student ACM chapters, and local user groups. Forty-six volunteers had signed up.

Budget

The projected FY 1990 budget contained adjustments made during the D.C. board meeting, as well as corrections for member dues income.

O’Dell felt that a committee for allocation of funds for emergencies in between board meetings is a prudent thing to do and it was agreed that we allocate $10,000 in FY '90 for creation of a miscellaneous fund.

The $3,000 in additional funds for a T-1 Link and phone installation charges for the terminal room was approved.

Student Scholarship

It was agreed to take the $5,000 for scholarship and apply it to expand the pool of potential applicants to include undergraduates.

Journal Report

Young and O'Dell reported that the journal was a full cycle behind in publication schedule, due to the problems in securing permissions for the use of music being used on the CD. It was suggested that other issues be substituted for this one until the matter is resolved.

Anaheim Technical Program

Mashey reported that the program was in reasonable shape and there were good papers. It was suggested that we have examples of the abstracts and subsequent final papers available as a guide to authors, and that examples of good and bad extracts could be published in :login:. Various people brought up scenarios for criteria on how to handle a best paper/presentation award. Mashey said he will figure it out.

Paper Review Process

Murrel stated that there were problems with feedback from the program committee regarding the review process. A large number of authors received slips with comments that did not contain enough practical information as to why their paper wasn’t accepted, or instructions to help guide them with their next paper. Klein and Murrel would work on a summary form.

Report on Transactions with Interested Directors

Appelman summarized his report on transactions, and these standards would apply to the Executive Director as well. Under Delaware and California law, transactions with interested directors are perfectly legal when the required procedures are followed:

1. The transaction must be approved either by a majority vote of the disinterested directors or by the voting members of the Association.

2. If neither of these is possible, the transaction must be demonstrably fair, both substantively and procedurally.

3. All material facts must be fully disclosed and considered by the Board prior to its approval.

4. The directors should make certain that their approval, and the discussions preceding that approval, are fully documented in the records of the meeting. If the approval is by committee, there should also be minutes of the committee meetings.

5. Each director must be pro-active with respect to the duties owed to the Association. These include the duty to proceed legally and within the authority given by the Board; the duty to exercise due care in making decisions and carrying out assignments; and the duty to act as a fiduciary in the best interests of the Association.

Appelman stressed that the fuller the disclosure the better. Board members’ responsibilities are primarily to the members, and these go past the legal dimension.

UUNET Report

Adams reported they had 1,400 sites, and there was the continuing problem of meeting the demand. They need to pay Sequent for upgrades soon, and would USENIX be willing to lend UUNET $40,000? Adams proceeded to
make a full disclosure of the details and the need to keep net cash available rather than putting it into the upgrade. It was agreed to extend a loan of $40,000 to UUNET, principal to be paid off over installments over the next 8 months at an interest rate and terms to be negotiated.

University of Capetown Request

Appelman said he had received an invitation from the University of Capetown (UCT) to write to the Department of Commerce for a clarification as to their status under the law as being an apartheid-enforcing agency. If we were to get a determination from the U.S. that they are not an apartheid-enforcing agency, we could make them members. Nemeth went over the past history of this issue. A discussion ensued regarding legal and ethical implications about restricting membership. It was suggested that we schedule a discussion for the June meeting on whether to proceed with a determination from the Dept. of Commerce, and to explore the various issues of the power and obligations of the Board of Directors to limit memberships.

Standards

Quarterman reported that the D.C. Standards BOF at the D.C. conference had gone well. He had also attended an IEEE standards policy seminar and an IEEE Computer Society Standards Activities Board meeting. These facilitated writing the USENIX ballot on IEEE/CS Technical Committee on Operating Systems (TCOS) procedures. However, he had not turned in USENIX ballots for 1003.1b or 1003.4 due to lack of time. There is a movement afoot, partly among the Institutional Representatives, to limit the rate of appearance of new IEEE standards committees by limiting the approval of new Project Authorization Requests (PARs) at the next IEEE TCOS meeting in April in Utah.

International Standards Committee Report

Pursuant to a proposal by Dunlop at the previous board meeting, lengthy discussions had taken place among representatives of EUUG, USENIX, AUUG, and JUS about the possibility of an international umbrella group of UNIX-related user groups for the purposes of: (1) tracking the growth of UNIX-related standards; (2) reporting on them; (3) obtaining some form of official representation to ISO/IEC JTC1 or SC22 (the parent groups of WG15), and (4) sharing costs. A letter of invitation had been drafted in hopes that the four groups mentioned above could send it to other user groups. Quarterman reported the recommendation of the USENIX subcommittee on this topic to have USENIX approve sending the letter of invitation with the understanding that any actual decisions on joining anything would be made by the whole board.

Next Board Meeting

It will be held at the Marriott Hotel in Anaheim, California, beginning June 10 and continuing on June 11.

– EY
Report on ISO/IEEE JTC1/SC22/WG15
Rapporteur Group on Internationalization Meeting
March 5-7, 1990, Copenhagen, Denmark

Dominic Dunlop, The Standard Answer Ltd., domo@tsa.co.uk

Denmark. A small country which has tax rates so high that its five million inhabitants complain that, when they buy themselves a car, they have to buy one and a half cars for the government. Some part of that tax goes to fund Dansk Standardiseringraad (DS), the national standards body, which works hard to ensure that the needs of Danes are not overlooked when larger nations get together to write standards. DS has got its teeth into international standards for computers, and with good reason: we’ve been doing things wrong all along. We'll have to mend our ways if we are to produce standards which really fill international needs, even if we don’t go as far as building in a framework which can easily accommodate Danish taxation.

Metropolitan Chicago today has a population larger than that of Denmark. Imagine that you’ve just rebuilt the downtown area after the fire of 1871, only to have Alexander Graham Bell come along with the telephone, Edison deciding to generate electricity, and railroad companies starting to promote interurban lines. All these innovations need new infrastructure – cables and conduits and tunnels which you just hadn’t known you’d need when you laid the roads, put up the buildings, and connected them to gas, water and drainage. As a result, competing telephone and electric companies string a tangle of wires from poles with little regard to safety and no regard for aesthetics or standardization, while elevated railways appear above existing roads, cutting off light at street level and filling upper floor rooms with smoke.1 Only after many years of disruption, digging up streets and making holes in the walls of existing buildings would telephones, electricity and public transportation be safely hidden beneath the ground,2 unseen, but playing an essential part in supporting the life of the city.

A descendant of Alexander Graham Bell’s telephone company now supports the UNIX operating system out of Chicago. UNIX is a lot like the Chicago of the last century. We’re at the stage of unifying the major variants in the POSIX standards and the commercial System V, release 4, only to find that there is an increasing clamor for whole new infrastructures to support international needs, to improve security, and to show that the system is performing as billed. Suddenly, we’ve got to add features to handle these requirements, and we’ve got to try to do it while observing the three conflicting maxims of standardization: do it once, do it right, and do it now. What’s more, we have to try to do it in a way which remains hidden: existing programs should not break, nor should they get noticeably bigger or slower.

POSIX is not alone: those responsible for computer language standards face the same problems, and have also been the subject of constructive Danish criticism.2,3 The Danes’ long-standing interest makes it particularly appropriate that the first meeting of the ISO POSIX working group’s special interest group on internationalization should be hosted by DS in Copenhagen. Internationalization is the process of removing cultural bias from a system, and then providing tools to allow system administrators to localize the system by adding a cultural bias of their own choosing. No wonder Dansk Standardiseringraad is interested in this technology: its employees

1. In 1887, the West Chicago Protective League complained "... the proposed elevated road would materially and irreparably depreciate the value of real estate upon said streets... and render the dwellinghouses thereon unfit for private residences..."[1] but amid the kind of political maneuverings for which the city is justly famous, the "El" got built anyway.

2. Well, in the case of Chicago, some of the public transportation. You can still ride the El.
court a syntax error every time they type its name at the UNIX shell. Internationalization will allow Danes to mold systems to their requirements, rather than having to rub along with implementation assumptions based on American practice.

The Japanese are interested too: their cultural differences make Denmark look close enough to the U.S.A. to be a fifty-first state! And the U.S.A. is interested because it has been charged by ISO with the production of ANSI standards base documents for the international POSIX standards, and wants them to reflect international needs. Denmark, Japan, and the U.S.A. sent representatives to the internationalization meeting. There were also observers from EUUG/USENIX (myself), the IEEE’s 1003.0 working group, and from an ISO study group which is grappling with the issues of character set use in computer languages.

The official title of the POSIX internationalization group is the ISO/IEEE JTC1/SC22/WG15 Rapporteur Group on Internationalization. Just to explore some of the jargon, a rapporteur is a technical expert nominated by a member body—a national standards organization such as ANSI or DS—to take an interest in a specialized aspect of a particular standards effort. WG15, the ISO POSIX working group, has rapporteur groups on security, conformance testing, and internationalization. The security group met in January, in conjunction with the New Orleans meeting of the IEEE 1003.4 working group; the conformance test group, which corresponds to the IEEE 1003.3 effort, met in Copenhagen along with the internationalization group (although this report does not cover its meeting).

Internationalization is peculiar in that, although the IEEE’s POSIX standards are drafted with international needs in mind, there is no internationalization working group within the POSIX project. There is a study group which, as part of the 1003.0 “POSIX Guide” work, is trying to decide how to bring internationalization into the official structure, so that it can be given officers, schedules, terms of reference, and all those other good things which make us standards people feel safer. It’s a big problem, because the issue really affects every aspect of POSIX—it just took a while to realize that it was an issue at all. Unlike realtime extensions, security extensions, or transparent remote file access for POSIX, internationalization doesn’t really make sense as an add-on to a basic operating system interface standard. Rather, the operating system and all its extensions need to be internationalized as a matter of course. Every other working group in the IEEE POSIX is charged with producing a distinct standard, but it is difficult to see how a new group dealing with internationalization could be given such a goal.

ISO has a similar problem, but it’s worse because the organization has so many balls to keep in the air. If it is to apply the “do it once” and “do it right” maxims to internationalization, it seems clear that the issue must be handled near the top of Joint Technical Committee 1, the information technology standards group. After all, as well as computer languages and operating systems, internationalization affects communications, document standards, database, and much more. ISO recently bit a similar bullet, establishing a new subcommittee (SC27) immediately below JTC1 to handle the security issues which are beginning to affect so much of its work. It may yet do the same with internationalization.

The “do it now” criterion, on the other hand, argues in favor of addressing internationalization at a lower level—doing the work in a new department, rather than going to the trouble of establishing a whole new division. SC22, which is responsible for language and operating system standards, is currently considering the formation of a new working group at the same level as WG15 (C language), WG15 (POSIX), and the rest. This proposal has run into opposition, both from those who say that the issue should be handled at a higher level, and from those who feel that there isn’t an

3. ISO 646, the earliest ISO standard for information technology, is the international derivative of ASCII. Its Danish variant replaces ASCII’s \ with aa. Around the world, # and * are replaced by other characters in standards derived from ISO 646. See for much more information.
issue: after all, aren't ISO's standards supposed to be international anyway?

Meanwhile, WG15 has established a subordinate group to handle internationalization at the lowest level possible. As somebody said at the meeting, "You can't get much lower than us." We spent our time discussing what we were supposed to be doing — and, equally important, what we could leave to others. In the end we came up with a little list:

Terms of Reference
The rapporteur group on internationalization (RIN) will study the aspects of internationalization related to POSIX and report its findings to SC22/WG15.
(Bland, imposing no needless restrictions on what we can do.)

Program of Work
1. Carry out survey to capture most of the requirements relevant to internationalization.
(A job and a half. We have to search out users around the world, and persuade them to tell us what features they really want, rather than what they can put up with, or program their way around.)

2. Identify and forward requirements with recommendations to WG15.
(So WG15 gets to carry the can for us...)

(Denmark and Japan have already done some work on "profiles" that customize POSIX to suit local needs. Their work suggests that current internationalization features are inadequate.)

4. Perform investigations as needed to advance the internationalization work of WG15.
(We can poke our noses into anything that takes our fancy...)

5. Review, from an internationalization perspective, documents submitted to WG15 for review and comment from an internationalization perspective.
(We definitely get to poke our noses into anything that comes past WG15...)

6. Review, and evaluate impact on work of WG15 of, other documents relevant to internationalization circulated in JTC1 or its subcommittees.
(And we'll try to get our hands on information from further afield.)

That's a lot of work. It defines the function of our particular mill, but that mill still needs grist. That feedstock has to come from outside our group, and, because of our lowly position, we have to ask WG15 to ask others to supply it. WG15, in turn, may have to refer some requests to higher authority: we want to be aware of anything which happens in SC22 which is relevant to POSIX internationalization — for example, what the C language people in WG14 are up to. That involves going up another level in JTC1's hierarchy. Getting in touch with other subcommittees, such as SC2, which looks after character sets, potentially involves going right to the top of the bureaucracy. (Luckily, in this particular case, SC22's study group on character sets can stand in for SC2.) Consequently, when WG15 next meets in Paris in June, it will have to deal with several resolutions concerned with turning on the taps and starting the information flow to the rapporteur group.

One of these taps is a little sticky: WG15 doesn't officially have a relationship with the IEEE's 1003.0 group, although it can, via ANSI, talk to 1003.1, 1003.2, and 1003.4 through 1003.9. The problem is that 1003.0 deals with profiles, baskets of standards which, when brought together, solve particular classes of problems — for example, those of transaction processing, realtime, or batch-oriented systems. Profiles are outside the scope of the ISO POSIX effort, so we can't officially talk to 1003.0, even though its study group is currently holding the baton on internationalization. Never mind. We'll do things unofficially until some official pathway is sorted out.

5. SC2's answer to life, the universe and everything is DP (draft proposal) 10646, which defines a 32-bit wide character set with 8- and 16-bit wide canonical versions for storage and transmission, and a 24-bit wide processing version for those who can get by with only eight million characters or so. As it's still at the DP level, it'll be a long time before it hits the streets, and, even when it does, there's the little matter of getting people to use it...
Apart from all this organizational stuff, we did review some existing documents. For example, DTR (draft technical report) 10176, a product of SCI4, discusses the treatment of characters appearing in language constructs, variable names, literals, and comments, and turns out to have implications for *sh*, *awk*, *yacc*, and the other "little languages" defined in DP 9945-2, the forthcoming international standard for the shell and tools. And a document from SC22’s study group on character sets suggests that source files should have some means of announcing the character set that they’re using. Could this mean typed files or resource forks for POSIX? How would we hide that?

The group next meets in Paris in June, just before the WG15 meeting. If you want to come along, you have to persuade your national standards body firstly that you’re a technical expert on POSIX, and then that they should appoint you as internationalization rapporteur. This may be surprisingly easy – considerably simpler, for example, than getting somebody to fund your trip. To quote from [8], “…standards committees would be hard-pressed to find people who participate on their voluntary committees with purely rational-economic expectations. Standards committees seem bent on justifying their existences by using hard data to prove that standards are good, yet they persist in using altruistic appeals to attract committee members.” If you feel like responding to the altruistic appeal of this article, contact me by electronic mail.

Alternatively, if you’re a European, you can remain seated in front of your terminal and participate in a news forum on ISO 646 and all that: Keld Simonsen of the Danish UNIX Users’ Group has volunteered to initiate a discussion of the European perspective on character sets for POSIX. Denmark may be small, but it’s certainly making its voice heard on this issue!

References

6. ANSI X3.159, 1989, Programming Language C.

---

6. UNIX’s elegant and flavorless files have already taken a beating from X3.159, the ANSI C standard[6], since other operating systems tend to support filing schemes which are merely tasteless.[7]
International Standardization

An Informal View of the Formal Structures as they Apply to POSIX Internationalization

Dominic Dunlop, domo@tsa.co.uk
January, 1990

This article provides an overview of the way in which the international standards community works, insofar as it affects POSIX and the incorporation into POSIX of internationalization features. I'm not going to describe the technology underlying internationalization other than to say that its aim is to make the operating system and applications software independent of the user's spoken language and its representation (character sets, collation, text direction, and so on). This done, localizations specific to each group of users can tailor programs to their requirements without the need for expensive and legally-problematic hacking of source code. (If you want to know more, let me know, and I'll either expand on the topic, or give a few pointers.)

Figure 1 shows the relationship of standards bodies as far as POSIX is concerned. (The picture may look very different for other standards efforts, such as Open Systems Interconnection, but that need not concern us here.)

All standards must originate somewhere, whether in industry, in a professional association, in a national standards body, or in an international standards body. In the case of the POSIX family of standards, the Institute (IEEE) has assumed responsibility for the initial production of the documents. The IEEE is a professional association which is open to qualified engineers, no matter what their nationality. It has been involved for many years in the production of consensus standards—that is, standards arrived at through a formal process which gives ample opportunity for any interested party to comment and vote on proposals.

According to the standards procedures of the IEEE, the main group of interested parties is its membership, although non-members are also allowed to participate. Unusually among standards bodies, voting on IEEE standards is nominally “one member, one vote.” (More typical standards bodies vote by corporation or by country.) The exception to the IEEE’s individual voting scheme is that institutions can also participate, provided that they represent a broad constituency, rather than a single narrow commercial interest. Currently represented on the POSIX effort are the Open Software Foundation, UniForum, UNIX International, USENIX, and X/Open. None of these is an official standards body, although all are involved in the production of materials on which future standards may be based. In some cases, the organizations produce documents which look and smell like standards but which, because they are not produced by an open (and slow, and legalistic) consensus process,
may well show some bias towards the interests of the originating organization. Known broadly as industry standards, these documents appear before consensus standards, and must subsequently be brought into line if a consensus standard is to succeed.

As Figure 1 shows, in the hierarchy of standards organizations, the IEEE is near the bottom. Above it is firstly the national level, then the international. As the IEEE is based in the U.S.A., it has gained accreditation from the U.S. national standards body, ANSI (the American National Standards Institute). This means that ANSI considers the IEEE competent to produce national standards on behalf of ANSI. Of course, accreditation by ANSI gives rise to an anomaly: the IEEE, through a democratic process potentially involving an international membership, is creating national standards for the U.S.A. I shall return to this issue later.

ANSI, in turn, is a “member body” of Joint Technical Committee 1 (JTC1), an international standards body formed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to handle the standardization of information technology. ANSI’s role in JTC1 is nominally to represent U.S. interests in the “one nation, one vote” process by which international standards are ratified. Other member bodies such as DIN (West Germany), JISC (Japan), and IRISI (Iran), play a similar part, making sure that no standard conflicts with their own national interests.

Member bodies may sponsor draft standards at the JTC1 level. In the case of POSIX, ANSI is the sponsor. It is the expectation of the international standards community that a draft standard sponsored by a national member body in this way is likely to show a bias towards the needs and culture of that member body, and so may require amendment and perhaps extension before it is suitable for adoption as an international standard. Certainly, both POSIX and the C language have come in for criticism at the international level for their lack of support for non-Roman alphabets.

In order to root out and correct any bias or omission in a draft standard sponsored by a particular member body, other member bodies are expected to pore over the proposal, and feed in changes which reflect their national needs. Obviously, this could take forever: approaching a hundred countries are represented on JTC1. Typically, the number of member bodies participating in a particular standards effort is limited, and of these few play a very active role. In the case of the POSIX effort, around a dozen member bodies are circulated with the working group’s paperwork, and of these, perhaps half are regularly represented at its meetings. Even so, by the time a national standard has progressed to the level of becoming a JTC1 draft, it is rather late to begin making changes — particularly if, as is the case for POSIX and C, there is a pressing need for an international standard.

As presented so far, the standards world is strictly hierarchical: a standard such as POSIX progresses from an accredited special interest group within a country, firstly to national level, and finally to international status. Officially, it is not until the final stage that interests outside the originating country get to comment on it. The process could be made more efficient if interest groups outside the originating country had a means of commenting at an earlier stage, but the hierarchy seems to preclude such comment.

Interestingly, there is a “side door” at the international level which can be used to short-circuit the normal time-consuming process. The top level of Figure 1 shows an organization in liaison with JTC1, the European Computer Manufacturers’ Association (ECMA), which has gained the privilege of being allowed to propose and comment on standards at the international level. The process of obtaining liaison status is both difficult and lengthy, and is open only to international organizations with a valid claim to representing a specific broad area of interest. (Besides ECMA, the World Health Organization and Mastercard International are among the sixty or so bodies in liaison with JTC1.) If the members of a liaison body can formulate a standard which is useful to them, liaison status allows that standard to be proposed for
adoption as a formal international standard. Since all bodies with such status are themselves international (or at least regional), such proposals are likely to satisfy international needs without much need for amendment. (ECMA has sponsored several standards for magnetic media in JTC1; banking interests have been active in the standardization of credit cards.) Indeed, JTC1 has developed a "fast track" approvals mechanism for use when member bodies agree that little review is necessary - although it has to be said that not every use of the fast track has resulted in a standard being approved.

The strict hierarchy imposed by ISO makes for easy and obvious management control, but is under some strain. Firstly, where emerging standards seek to accommodate international needs from their first drafting, the late review by national member bodies provided by ISO makes for unnecessary delay - delay which could be avoided if national bodies had an official means of providing input at an earlier stage. Secondly, regional standards organizations - most notably CEN, the European Standards Centre - are growing in importance, and do not fit well into a scheme which is set up according to strictly national guidelines.

These two problems combine to foster provincial attitudes on the part of standards makers - and politicians - involved with POSIX both inside and outside the U.S.A. Those inside reason that, since they are creating a U.S. national standard, international considerations are relatively unimportant, and can be left for later. Outside the U.S., standardizers reckon that it will be so long before they can mold a U.S.-produced standard to their own requirements that they might as well develop their own, probably incompatible, standards to fill their immediate needs. In Europe, a proposal to adopt issue 3 of X/Open's Portability Guide (XPG3) as a standard was strongly backed for a while, even though XPG3 is not wholly aligned with POSIX. (On the reasonable grounds that the 1003.1 standard had not been approved at the time of publication. XPG4 will be aligned with POSIX.) Interestingly, just as the IEEE is seen in Europe as representing U.S. interests, X/Open is seen by many U.S.-based observers as a European outfit, despite its many U.S. members.

Provincial attitudes among technical people and their managers outside the U.S.A. exacerbate the problems. Although the IEEE makes some effort to reach this constituency by holding one of the quarterly working group meetings outside U.S. every couple of years, the majority of attendees are always Americans. Europeans in particular seem, even if they have the inclination to attend, to find it difficult to justify the expense to their management. The interests of Arab countries and the Indian subcontinent are seldom represented at all. In contrast, delegates from Japan and other Pacific rim countries have been attending meetings in increasing numbers, even when lengthy and costly travel is involved.

Given the current structure of the international standardization community, is it possible to work within it and yet overcome the two problems which face the POSIX effort: that of obtaining useful international input at an early stage; and the parallel problem of preventing divergence between POSIX and emerging industry, national, and regional standards? Can the current structure accommodate formal mechanisms which provide for solutions, or will the problems remain unless the structure itself is changed?

Until now, practical international input to POSIX has come from two sources which are not a part of the formal hierarchy of international standardization: UniForum and X/Open. As I have already mentioned, X/Open is an international grouping seen by some as primarily European; its active membership has to date consisted of computer suppliers. UniForum, which was known as /usr/group until 1989, is a grouping of hardware suppliers, software authors, value-added resellers, and users. As with X/Open and other groupings, it is the suppliers which have played the largest part in the organization - users have seldom made their voice heard. UniForum is U.S.-based, but has affiliates around the world. These affiliates are largely autonomous, and, despite efforts to involve them, have played almost no part in UniForum's standards activities - even when
these are involved with internationalization. (While UniForum's Technical Subcommittee on Internationalization has active participation from outside the U.S.A., the people concerned became involved directly, rather than through their local UniForum affiliates.) USENIX, the other user grouping with institutional representation to the IEEE POSIX project, has a better claim to providing a forum for users, but is almost exclusively North American, and, unlike UniForum, has no internal structures concerned with standardization. The European UNIX systems User Group (EUUG) has a truly pan-European membership made up, like that of USENIX, primarily of computer programmers and technical users, but has not participated officially in any standards effort. Its involvement to date has been confined to the co-sponsorship with USENIX of a standards monitor service, which provides members with information about progress on POSIX and in related areas.

It is my view that, if international interests are to play a greater part in the drafting of POSIX standards, they must be represented formally within the IEEE. This is not to minimize the importance of the work done by UniForum, but rather to say that an official stamp of some sort is necessary in order that its importance receives a wider recognition both inside and outside the IEEE. Unlike other topics handled in the past by UniForum, real-time and transaction processing among them, internationalization has never officially been incorporated into the POSIX effort because it cannot stand alone. There cannot usefully be such a thing as a standard for internationalization; rather, internationalization should be a consideration in the drafting of any standard for computer software.

The 1003.0 (POSIX Guide) working group is currently wrestling with the problem of handling internationalization issues within POSIX. It may be possible to borrow a useful concept from ISO: that of the rapporteur group. Rapporteur groups cut across normal boundaries, bringing together those who are interested in some problem or activity which is common to a number of standards projects.

It is over-optimistic to hope that bringing internationalization officially into the POSIX fold will result in immediate participation by those who currently wait until documents reach the ISO level before commenting through their national member bodies. One way to reach this audience might be to convince it that the IEEE is indeed an international, rather than strictly North American, grouping. A radical way of achieving this would be for the IEEE to seek liaison status with JTC1, so obtaining a means of submitting base documents directly, instead of through ANSI. To do this would involve the IEEE in the considerable expense and logistic complexity of sponsoring standards – a task for which resources are not currently in place in an organization which seldom gives the appearance of being over-endowed with resources.

In any event, even if the IEEE were to apply for liaison status tomorrow, it would be a long time before it was granted. Unless or until this happens, it seems to me that it is the duty of user groups around the world to encourage their members to play a part in the process through the IEEE. So that's what I've been doing in this article!
An Update on UNIX and C Standards Activity

Jeffrey S. Haemer
Report Editor, USENIX Standards Watchdog Committee

What the reports are about

Reports are done quarterly, for the USENIX association, by volunteers from the individual standards committees. The volunteers are familiarly known as "snitches" and the reports as "snitch reports." The band of snitches and I make up the working committee of the USENIX Standards Watchdog Committee. The group also has both a financial committee: Alan G. Nemeth, Ellie Young, and Kirk McKusick (chair); and a policy committee: the financial committee plus John S. Quarterman (chair). Our job is to let you know about things going on in the standards arena that might affect your professional life — either now or down the road a ways.

An official statement from John:

The basic USENIX policy regarding standards is:

to attempt to prevent standards from prohibiting innovation.

To do that, we:

- Collect and publish contextual and technical information such as the snitch reports that otherwise would be lost in committee minutes or rationale appendices or would not be written down at all.
- Encourage appropriate people to get involved in the standards process.
- Hold forums such as Birds of a Feather (BOF) sessions at conferences. We sponsored one workshop on standards.
- Write and present proposals to standards bodies in specific areas.
- Occasionally sponsor White Papers in particularly problematical areas, such as IEEE 1003.7 (in 1989).
- Very occasionally lobby organizations that oversee standards bodies regarding new committee, documents, or balloting procedures.

- Starting in mid-1989, USENIX and EUUG (the European UNIX systems Users Group) began sponsoring a joint representative to the ISO/IEC JTC1 SC22 WG15 (ISO POSIX) standards committee.

There are some things we do not do:

- Form standards committees. It's the USENIX Standards Watchdog Committee, not the POSIX Watchdog Committee, not part of POSIX, and not limited to POSIX.
- Promote standards.
- Endorse standards.

Occasionally we may ask snitches to present proposals or argue positions on behalf of USENIX. They are not required to do so and cannot do so unless asked by the USENIX Standards Watchdog Policy Committee.

Snitches mostly report. We also encourage them to recommend actions for USENIX to take.

We don’t yet have active snitches for all the committees and sometimes have to beat the bushes for new snitches when old ones retire or can’t make a meeting, but the number of groups with active snitches continues to grow (as, unfortunately, does the number of groups). This quarter, you’ve seen reports from .0, .1, .2, .3, .4, .7, .8, .11, and .12, as well as reports from 1201 and from X3J11 (not really a New Orleans report, but useful none the less).

If you have comments or suggestions, or are interested in snitching for any group, please contact me (jsh@usenix.org) or John Quarterman (jsq@usenix.org). If you want to make suggestions in person, both of us attend the POSIX meetings.
Reports on the October 1989 Meeting in Brussels
(continued from ;login: vol. 15, no. 2)

Report on IEEE 1003.2: Shell and tools

Randall Howard <rand@mks.com> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Background on POSIX.2

The POSIX.2 standard deals with the shell programming language and utilities. Currently, it is divided into two pieces:

- POSIX.2, the base standard, deals with the basic shell programming language and a set of utilities required for application portability. Application portability essentially means portability of shell scripts and thus excludes most features that might be considered interactive. In an analogy to the ANSI C standard, the POSIX.2 shell command language is the counterpart of the C programming language, while the utilities play, roughly, the role of the C library. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., popen, regular expressions, etc.). [Editor's note — This document is also known as "Dot 2 Classic.]"

- POSIX.2a, the User Portability Extension or UPE, is a supplement to the base POSIX.2 standard; it will eventually be an optional chapter of a future draft of the base document. The UPE standardizes commands, such as screen editors, that might not appear in shell scripts but are important enough that users must learn them on any real system. It is essentially an interactive standard that attempts to reduce retraining costs incurred by system-to-system variation.

Some utilities have interactive as well as non-interactive features. In such cases, the UPE defines extensions from the base POSIX.2 command. An example is the shell, for which the UPE defines job control, history, and aliases.

Features used both interactively and in scripts tend to be defined in the base standard.

In my opinion, the biggest current problem with the UPE is that it lacks a coherent view: it's becoming a repository for features that didn't make it into the base standard. For example, compress is in the current UPE draft. It's hard to rationalize classifying file formats as an "interactive" or "user portability" issue, yet the one used by compress is specified in the UPE. It certainly doesn't fit in with a view of the UPE as a standard that merely adds utility syntax information (e.g., information that would allow users to type the same command line to compress a file on any system). This highlights the schizophrenic nature of the UPE: it addresses a range of different needs that, taken together, do not appear to define a whole. Dot 2 Classic, to my taste, appears to have far more unified scope and execution.

A second, related, problem with the UPE is that there appears to be less enthusiasm for it than for the base standard. A number of people, including me, understand the need for it, but it doesn't appear to have the strategic importance of the base. [Editor's note — The UPE is, frankly, controversial. Like 1201, the committee undertook the UPE out of a fear that if they didn't, NIST would do the job without them. Supporters note that although its utilities are probably not necessary for portability of most software, it would be unpleasant for programmers to do the porting work without them. Detractors counter that POSIX was never intended to cover software development and that the group is exceeding not only its charter, but that of the entire 1003 committee.]"

Status of POSIX.2 Balloting

POSIX.2 is in its second round of balloting. The first ballot, on Draft 8, produced many objections that are only partially resolved by Draft 9. Although there were only fifty-four pages of unresolved objections remaining after Draft 9 was produced, the current balloting round is not restricted to existing objections, and there will almost certainly be many new
ones. Remaining objections range from the perennial war between David Korn and the UNIX Support Group over what features should be required in the POSIX shell, through the resolution of the incompatible versions (Berkeley and USG) of `echo`, to the treatment of octal and symbolic modes in `umask`.

A digression to illustrate the kind of issues being addressed:

In March of 1989, a study group from 1003.2 met at AT&T to resolve major objections to the shell specified in Draft 8 by the two warring parties. This was a good place to hold the meeting, since both parties are from AT&T: one led by David Korn of Bell Labs, the author of the popular Korn Shell (KSH) the other, a group led by Rob Pike of Bell Labs Research and the UNIX Support Organization, advocating more traditional shells, like the System V Bourne Shell and the Version 9 Research Shell. Korn’s group contends that the shell should be augmented to make it possible to efficiently implement large scripts totally within the shell language. For example, while the more traditional camp views shell functions as little more than command-level macros and uses multiple scripts to modularize large shell applications, the Korn shell views functions as a tool for modularizing applications, and provides scoping rules to encourage this practice.

The two philosophies engender different opinions on issues such as the scoping of traps within functions and the use of local variables. Other contentious issues were the reservation of the brace (\{\}) characters as operators (rather than as the more tricky “reserved words”), the promotion of tilde expansion to a runtime expansion (like parameter expansion), and the issue of escape sequences within `echo`, `print`, and `printf`.

The meeting produced a false truce. I attended, and believe that both parties had different views of the agreement that came out of the meeting. As a result, Draft 9 produced balloting objections from both parties and the dispute continues unabated. Shades of POSIX.1 Tar Wars...

I suspect the next draft (Draft 10) will fail to achieve the consensus required for a full-use standard. This is a good thing. Useful features are still finding their way into the document. (Draft 9 introduces `hexdump`, `locale`, `localedef`, and more.) Also, the sheer size (almost 800 pages) of Draft 9 has prevented manyballoters from thoroughly reviewing the entire document. Still, there is a stable core of utilities that is unlikely to change much more than editorially; I predict the standard will become final around Draft 12.

A mock ballot on Draft 4 of the UPE will probably start after the New Orleans meeting in January, and the resulting Draft 5 will probably go to a real ballot somewhere in summer to early fall of 1990. Although many sections remain unwritten or unreviewed, the UPE is a much smaller standard than POSIX.2 and should achieve consensus more quickly.

**Status of the Brussels Meeting**

The Brussels meeting focused on the UPE, with only a summary report on the status of balloting for the base standard. For most of the meeting, small groups reviewed and composed UPE utility descriptions. The changes generated at the meeting will appear in Draft 3.

The groups reviewed many utilities. The chapter on modifications to the shell language (for interactive features) is now filled in, and such utilities as `lint89` (the recently renamed version of `lint`), `more`, etc. are approaching completion. Still, much work remains.

[Editor’s complaint — We think renaming common commands like `lint` (“`lint89`”) and `cc` ("`c89`”) is both cruel and unusual. We are not eager to re-write every makefile and shell script that refers to `cc` or `lint`, nor to retrain our fingers to find new keys each time the C compiler changes. The name seems to have been coined by either a hunt-and-peck typist, or someone who has longer and more accurate fingers than we do. (Was it, perhaps, the work of Stu Feldman, author of `f77`?) Moreover, replacing commands with newer versions is
commonplace and traditional in UNIX. Examples like make, troff, and awk spring to mind. If an older version is kept on for die-hards, it's renamed (e.g., otroff, oawk).

One Dot-Two member rebuffed our objections with the reply, "But, you see, this isn't UNIX: it's POSIX."

Because the meeting was in Europe, attendance at the working group meetings was lower than normal (20-25 rather than the normal 35-40 in POSIX.2). Nevertheless, the choice of location served a purpose. The meeting was held in Brussels to garner international support and participation, particularly from the European Economic Community. There were many EEC representatives at the background sessions on POSIX and two or three European working group members in the POSIX.2 meetings who wouldn't normally have attended. Though it remains to be seen what will come out of having met in Brussels, I am convinced that the extra effort will prove to have been justified.

Report on IEEE 1003.5:
Ada-language Binding

Ted Baker <tbaker@ajpo.sei.cmu.edu> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

The P1003.5 group is producing an Ada-language binding for 1003.1. The Brussels meeting had two objectives: to reach consensus on a draft document to be distributed for mock ballot, and to solicit input from the European community. We achieved the first but not the second; only one of the ten attendees was European (Olle Wikstrom, from Ericsson).

The technical editor (David Emery) and the chapter authors had worked very hard between meetings to produce version 3.2 of the document, and Dave brought copies to the meeting. The working group reviewed it to try to correct any serious errors or omissions before mock ballot.

There was a lengthy discussion about schedule and logistics for the mock ballot. The present plan is to send out copies of the next draft, in ISO format, to both the ISO and the entire 1003.5 mock ballot mailing list. [Editor's note: All committees are reformatting their documents in ISO format to smooth the way for ISO acceptance (see Dominic Dunlop's report on WG15 for more details), and an IEEE copy editor appeared on the scene in Brussels to give P1003.5 guidance and help in this.] Since there is no way that enough input can be received before the next POSIX meeting, in January, the group has scheduled a special meeting for mock ballot resolution, between the January and April POSIX meetings, to be held in Tallahassee. The objective will be to produce a proposed standard to be reviewed at the April meeting.

Most technical issues discussed were minor, compared with previous meetings. The most significant, and complicated, was the treatment of system configuration limits. Here are three problem areas:

1. Tri-state configuration parameters (true, false, undefined) in the POSIX C binding need to be treated differently in the Ada binding, because Ada prohibits references to undefined symbols (i.e., Ada lacks an #ifdef facility).

2. For the same reason, it isn't clear how an Ada binding can accommodate future POSIX extensions. Suppose, for example, a future extension adds a new configuration constant. How does one write an Ada program that takes advantage of the new feature on implementations where it's available without preventing the same program from compiling on older implementations, where it's not?

3. Because Ada compilers can do optimizations, such as dead code elimination, based on static expressions (the nearest analog to some C preprocessor capabilities), it is important to provide compile-time constants, where safe. At the same time, to support "bubble pack" software that is usable on different system configurations, programs should also be able to defer binding such values until run time.
The group did achieve consensus on a treatment of configuration limits for the mock ballot. It includes a combination of functions, to allow software to defer resolution of system limits and characteristics until runtime, and implementation-defined constants and numeric ranges, to allow optimizers to take advantage of information available at compile time. This does not fully solve all the problems mentioned above. Perhaps the mock ballot process will turn up some suggestions for improvements.

The treatment of process arguments and environment variables, which must be provided as parameters when starting a new process or calling Exec produced another controversy.

Unlike C, Ada does not allow pointers to stack or statically allocated objects. An Ada POSIX interface implemented over a C-language binding must bridge this gap somehow. For example, an implementation might use a C-compatible data structure and hide the non-Ada details, or use an Ada data structure and translate between the two forms. Everyone agreed that the interface should avoid constraining the implementation, but the first interface solutions appeared to rule out desirable implementations. The present solution permits an application to ensure that if the Ada POSIX interface machinery allocates any “heap” storage this storage is be recovered, while allowing an implementation to impose restrictions that would permit stack allocation. A price paid for this compromise is that writing portable applications takes more care: an application that works OK with one implementation may lose storage or exceed size limits with another.

At the previous two meetings, we had substantial interaction both with other groups working on language-independence and with P1003.4 (real-time). There was much less this time, partly because the group was concentrating so hard on getting ready for mock ballot, partly because meetings were spread over several buildings, and partly because P1003.4 mostly skipped Brussels.

On the administrative side, Steve Deller was promoted from Vice Chairman to Chairman (in charge of external affairs and running meetings) and Jim Lonjers was chosen as Vice Chairman (in charge of administering ballot resolution). This change was required because the ex-Chairman (Maj. Terry Fong) has been unable to participate regularly in the working group recently, owing to conflicts with his professional duties.

Another issue that came up was whether working group members are at liberty to publish papers or present talks on the 1003.5 work. The answer is, “Yes.” Until now, some members have been exercising self-censorship, based on an earlier agreement designed to discourage anyone (e.g., defense department personnel) from making commitments (e.g., requiring use of the POSIX Ada binding in contracts) based on erroneous (e.g., overly optimistic) progress reports. It did not take much discussion to agree that such censorship is now counterproductive, and may never have been wise. At this point, P1003.5 certainly wants public exposure of its draft document, and hopes that such exposure will generate more reviewers and active working group members.

Report on IEEE 1003.7: System Administration

Steven J. McDowall <sjm@mca.mn.org> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Background

Now, almost everyone agrees that 1003.7 should deal with networks, not just isolated systems. To wit, it would be nice if I could administer all the machines in a network from a single machine with simple commands. For example, to add a user to all machines in the domain mn.org, all I should need to do is issue a command like adduser -d mn.org -options -parameters username. The question is, without any de facto standard already in place to adopt, how can we achieve this?
The Approach

This is important, so pay attention. Because the major goal of 1003.7 is to create a standard way to manage a set of objects, the group has decided to take an object-oriented approach. Our idea is to begin by creating a list of objects to manage, then to follow that by defining the set of commands to manage each object. This approach is novel for both system administration and POSIX. It will probably require more work on the front end to define the objects, their attributes, and their relationships, than to define the actual command structure to support and manipulate them. Whether this approach will work remains to be seen.

The Meeting

The meeting was boring. To put it bluntly, the week was simply a work week. Objects (and sub-objects) were defined and discussed in detail, then put in the draft. Little got done on the first and last days, due to EEC formalities, which left us with three working days instead of the normal four and a half. Attendance was pretty dramatically reduced, too. About half the normal North Americans showed up, probably because of the location, and only one (yes one...) new European came even though we were meeting in Europe. Oh well, except for my having had my passport stolen, it was a good chance to see Belgium.

Concerns

1. The process is taking a long time to move ahead, both because of the difficulty involved and because we seem to attract less manpower than many other groups. Moreover, since we're taking a radical approach, it takes extra time to teach the ideas to anyone new that does come.

2. System administration doesn't have the glamour of some of the other areas being standardized. As the Rodney Dangerfield of POSIX, 1003.7 gets no respect.

3. The notation we're using to define our objects is ASN.1. "Why ASN.1?" you ask. Simply because it's a standardized meta-language to describe abstract data types. The feeling was that this would help make the whole package more suitable for interoperability. I bring this up because there's some movement throughout 1003 to redo all data structures in a new meta-language created by some of the people working on language-independence. Not only would this require that we go back and redo our definitions, but I also think ISO will only allow the use of standardized data-languages in their standards. Does anyone out there know if there is such an ISO restriction? If so, it's important for 1003 as a whole, not just for dot seven.

4. Currently, almost all working-committee members are from vendors. IBM, DEC, HP, AT&T, and others are well-represented. A few interested parties, like OSF and /sys/admin are there as well, but as far as I can tell, there isn't one real user. By "real user" I mean someone who does nothing but administer a system. All of us are connected somehow with creating an administrable system or getting paid to do so. Of course, I should make clear that we all have to administer systems of our own, so we're not simply an ivory tower group with no real experience, but representation is still grossly unbalanced.

5. Finally, there's been a loss of focus on interoperability directly attributable to the loss of our X/Open representative, Jim Oldroyd. Jim was well respected and made many valuable contributions, but can no longer attend our meetings. As the X/Open representative, he was very concerned with multi-vendor environments, and was a major force in helping us focus on and ensure interoperability. I am not saying that no one else on the committee cares about the issue, but it does seem to be being pushed aside in a spirit of, "I think we shouldn't have any interoperability problems if we do this, so let's do it and worry about it later on." Jim had helped provide a more positive, direct approach of determining up front what would be needed for true interoperability. If X/Open is still interested in System Administration, and in making sure
the 1003.7 standard includes provisions for interoperability, we could still use their help.

Report on IEEE 1003.8/2: Networking (IPC)

Steve Head <smh@hpda.hp.com> reports on the October 16-20, 1989 meeting in Brussels, Belgium:

Overview

P1003.8 is the IEEE POSIX networking standards committee, working on network standard interface definitions for POSIX. The committee is currently divided into six subcommittees: transparent file access, network IPC, remote procedure call, OSI/MAP services, X.400 mail gateway, and directory services.

This report is a summary of the activity in the network IPC subcommittee, which is currently working on two potential interfaces, a "detailed" interface (DNI) and a "simple" interface (SNI). DNI is roughly (though not exclusively) at the transport level. SNI is intended to be somewhat simpler to use than DNI, but at roughly the same level.

At this meeting, presentations of DNI and SNI were made at the EEC (Common Market) headquarters in Brussels. Discussions on DNI (definitions) and SNI (routines) continued. The main topics of discussion were:
1. DNI, SNI presentation to EEC
2. DNI definitions
3. SNI routines
4. Schedule
5. Security
6. P1003.8/2 → full POSIX committee

Detail

1. DNI, SNI presentation to EEC

Keith Sklower and Steve Head gave presentations on DNI and SNI respectively to POSIX attendees at EEC (Common Market) headquarters. This meeting was scheduled in Brussels primarily to obtain European input. The presentations went well, and attendees included X/Open and EEC representatives.

No significant differences of opinion or direction were noted between the committee and other attendees. This indicates some degree of success (?). (Other networking groups, such as directory services, were not so fortunate.)

This meeting "broke the ice" with international organizations in the area of networking, and we now expect increased interaction with those organizations.

2. DNI definitions

The committee discussed DNI definitions. Steve Head presented a paper on the subject. Suggestions made at the meeting will be incorporated into a future version of the paper, which will be circulated via electronic mail. If no further significant issues are raised, it will be incorporated into the next DNI draft.

3. SNI routines

The committee discussed SNI routines, based on a paper from Keith Sklower. No conclusions were reached, however, this particular discussion was very useful since it brought a number of goals and requirements for SNI into clear focus.

SNI is adopting some characteristics of ISODE (the ISO Development Environment). This is probably beneficial since it means that SNI will be partially based on a working implementation instead of being entirely new. As such, it may gain importance as a migration strategy for transferring applications from TCP/IP to ISO. (ISODE stands for the ISO Development Environment, a publicly available collection of networking software that runs over either TCP/IP or ISO transport and allows higher level applications to be oblivious to the type of transport a given system provides.)

4. Schedule

The working schedule has been delayed by the need to make presentations at Brussels, instead of doing "real work." Originally, we had scheduled the topics of connection setup,
connection tear-down, and name resolution for this meeting. These topics were not discussed, and our schedule has been shifted back a quarter to reflect this. These topics will be discussed at the next meeting.

5. Security

We held another joint meeting with the POSIX security group, P1003.6. An electronic mailing list was created for the topic of network security. For more info or to be put on the list, please contact Mike Ressler (bellcorempr or mpr@bellcore.com). A list of topics on networking security to begin discussions on was initiated.

6. P1003.8/2 → full POSIX committee

The decision to make P1003.8/2 a full POSIX committee was postponed by the POSIX executive committee (SEC). This subject will be re-addressed at the next POSIX meeting in January.

---

Reports on the January 1990 Meeting in New Orleans

Report on IEEE 1003.0: POSIX Guide

Charles Severance <crs@convex.cl.msu.edu> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Dot zero is producing a guide to the POSIX Open System Environment (OSE). The guide will bring existing and evolving standards together to provide specifications for all aspects of an OSE - everything from application programming interfaces to user interfaces and system management. It will give users an overview of the 1003, and other related standards, describe their interrelationships, and help them select the subset of available standards necessary for any particular application.

Draft Six Review

The group reviewed draft six, and points of special interest were:

- the formal definition of “open system”
- internationalization
- an editorial review of the entire document to ensure a consistent style
- a review of some high-level architecture diagrams, proposed to make Chapter 3 (“Overall Architecture”) easier to understand

The only one of these discussed by the entire group was the definition of “Open System.” To simplify the definition we created a definition for “Open Standard” which was used in the Open System definition. Here is the definition we finally agreed on:

Open System: A system that implements sufficient Open Specifications for interfaces, services, and supporting formats which enable properly engineered applications software: a) to be ported across a wide range of systems with minimal changes, b) to interoperate with other applications on local and remote systems, and c) to interact with users in a style which facilitates user portability.

Open Specification: A public specification which is maintained by an open, public, consensus process to accommodate new technologies over time and consistent with international standards.

The group won’t define “user portability” until next meeting, but the idea is that users should see a consistent interface from application to application, both within and across systems. Public user-interface standards should simplify both user training and vendor documentation.

The other issues were handled in small working groups.

1. Internationalization. This group identified parts of the document affected by internationalization and other “cross-component” issues, such as system management and security. They promise to present new draft text for the
internationalization sections by the next meeting.

2. Editorial review. This group tackled the no-fun jobs of reviewing the entire draft for style and identifying areas that had too much, or too little, detail. Along the way, they proposed a style guide and template for sections of Chapter 4.

3. Architectural overview. This group continued work on Chapter 3 to complete the text of the chapter, and worked to simplify it, and make it easier to understand. The CCTA (UK) presented a high-level classification scheme called “MUSIC” (Management, User Interface, System Interface, Information Interchange, and Communication) as a potential contribution to chapter 3. The chapter will have extensive modifications and additions for the next meeting.

Application profiles

Next meeting we’ll discuss exactly what must be in a POSIX Application Environment Profile (AEP). Profiles will affect and generate procurement issues, so this will be a key discussion.

Profiles specify a set of standards for specific computing areas, such as supercomputing. Not all standards will be required for all areas; a profile lists the subset of the standards necessary for a particular area.

The biggest point of contention in this discussion will probably be whether 1003.1 [Editor: the system interfaces set out in the Ugly Green Book] will be required for all profiles. Should vendors be allowed to advertise compliance to, say, 1003.11 (transaction processing), if they’ve implemented that standard on an underlying system that doesn’t support lower-level POSIX calls like fopen()? (There isn’t a standard for 1003.11 yet, but you get the idea.)

One argument advanced for requiring 1003.1 is that it will force vendors to adopt it more quickly. I don’t think that 1003.1 needs any help in that area. Another is that requiring compliance will ensure that vendors who want to advertise POSIX-compliant systems are following the general POSIX direction and not just implementing the simplest standard so they can claim that their system implements “some POSIX.”

An argument made against the requirement is that it may damage implementations. For example, real-time systems may lack even a file system, and may want a very limited subset of the POSIX interface to keep the implementation as small as possible. If all of 1003.1 is required, vendors may have to add costly and unnecessary features just to claim POSIX compatibility.

When the dust settles, I think 1003.1 will be strongly suggested but not required, because 1003.1 is a pretty arbitrary subset of any list of “required system interfaces.”

[Editor: We disagree. 1003.1 is a set of applications programming interfaces carefully chosen to be necessary and sufficient to make an operating system UNIX-like for the C programmer. Providing standards for a UNIX-like operating system should be the goal of the POSIX standards, and attempts by vendors uncomfortable with UNIX to dilute the effort should be cut off at the pass.]

[Author: POSIX must evolve a set of independent standards that have UNIX as their heritage. POSIX standards are all evolving as UNIX-like standards. Why discourage a vendor from implementing some subset of UNIX-like standards just because the vendor is not ready to provide a complete 1003.1 implementation?]

Want to go to a POSIX meeting?

This was my first POSIX meeting. In case you haven’t been and are thinking of going, here are a couple of things you’ll want to know.

New people are welcomed. As a practical matter, it helps to stick with a group for the entire week. It’s tough to understand much if you come into an advanced discussion cold. It would help if each group summarized its purpose and listed the big issues at the beginning of each meeting, to get everyone in the proper
frame of mind. Still, you'll be granted a sort of first-time armor to protect you when you ask naive questions or need clarification. For extra insurance, use the phrase "I will take an action item..." often.

Report on IEEE 1003.1:
System services interface

Mark Doran <md@inset.co.uk> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Most published standards inevitably require updating through corrective supplements. P1003.1 has now reached that stage. The first supplement, P1003.1a, is at an advanced stage and was the central issue at the New Orleans meeting.

Also on the agenda were:
- further talks with the group working on transparent file access;
- more language-independent-specification work; and
- a run-through of the material in the embryonic second corrective supplement, P1003.1b.

P1003.1a Ballot Resolution

The first corrective supplement to IEEE 1003.1-1988 (POSIX.1) is intended to correct errors and oversights in the first publication with a view to clarifying the intent. It is definitely not meant to introduce new functionality or behavior into the standard.

This work received its second recirculation ballot during the week preceding the New Orleans meeting. Donn Terry, chair of P1003.1, hopes that one, or at most two, more recirculations will bring the document to a publishable state. Accomplishing this will send it off to ISO, who will ballot it for six months. (That's right, six months; an IEEE recirculation ballot lasts ten days – does this seem a little lopsided to you?)

The details of the content of P1003.1a and its ballot resolution are long and complex, so I won't repeat them here. However, there is one issue worth raising which the ballot brought to light. On the subject of changes relating to the support of split baud rates, one balloter commented:

While we do not agree with the direction this issue is obviously taking, we will abide with the decision of POSIX insofar as split baud rates are concerned.

But we would be remiss in our responsibilities if we did not express our complete outrage with the provincial attitudes expressed by a number of the ballot comments we have had the pleasure to review during this recirculation period.

Split baud rates ARE NOT uncommon with a great number of the community of users of these standards. Obviously, many of those submitting ballots have not had the opportunity to consider the needs or requirements of users outside their own immediate view. We abhor such a limited, irresponsible scope, especially considering the nature of the tasks we are charged with resolving. It is our hope that we shall do better in the future.

Only rarely are standards meetings graced with such florid language, and the balloter clearly has at least the tip of his tongue in his cheek. However, there is, underneath this bonhomie, a serious point being made.

The IEEE is an ANSI-accredited standards-developing body, responsible for making standards pronouncements for use in the USA. All POSIX standards are being passed to ISO for potential adoption as international standards. The POSIX steering committee (SEC) has declared that POSIX would like to think of itself as an internationally accessible organization. If POSIX is indeed to be internationally accessible then the attitudes of some of those who attend will have to change. Take for instance, the split baud rate issue mentioned above.

Working group discussions revealed that split baud rate support, though a non-issue in the USA, is important in Europe. (The reasons for this stem from the way the PTTs in Europe structure their charges for communications lines – PTTs are Europe's little AT&T
equivalents.) To cut a long story short, the P1003.1 working group decided that split baud rates are not important enough to require explicit support for them in the standard.

This may be quite reasonable. What is unacceptable is the apparent scorn with which these proposals were regarded by a minority of the participants in the discussions. If POSIX proceedings are to lead toward internationally useful standards then all participants should be mindful of the fact that there is a flourishing community of users who do not live in the USA.

Split baud rates are not of earth-shattering importance, nor even terribly interesting; were this an isolated incident, it would not even be worth mentioning. Unfortunately, I have encountered this type of attitude on many occasions. Let’s hope that ballot comments like that presented above reduce this frequency. (“What are split baud rates?” the American reader is asking. Serial lines like those plugged into the back of “dumb” terminals can be set to transmit at high-speed while receiving at a lower speed, e.g., 9600 and 75 baud; this can be useful if you regularly send screenfuls of data to a terminal but only expect the odd character or two back in the other direction. POSIX supports this by supplying cfi(set, get)(i,o)speed() and tcg(set, get)attr() – that’s six interfaces, see? : -)

Transparent File Access (TFA)
The TFA group (now P1003.8) presented several detailed questions about the behavior that P1003.1 would like to see from a TFA implementation in several “corner cases.” Dot one’s response is that a few compromises can be made where there are serious performance issues, but the spirit of the original POSIX.1 should be retained wherever possible.

On a more interesting note, at a TFA BOF (Birds Of A Feather session – that’s a cozy chat after hours), it was suggested that a subset TFA specification might be produced before the full TFA specification. Such a specification would not provide full POSIX.1 behavior but would probably be enough to allow POSIX implementations to connect with existing FTAM and NFS file server machines, which should suffice for many applications.

Language-Independent Specifications
In my last report, I hadn’t heard a worthwhile justification for this work or the approach being taken. I still haven’t.

P1003.1b
This supplement, still being formed, will be the first to introduce new functionality into POSIX.1. Highlights so far include symbolic links, and file-tree walking (more ways to find files and directories in file systems). If you have a favorite interface which has not yet made it into a POSIX standard, you might be able to get it in by proposing it for inclusion in P1003.1b.

Report on IEEE 1003.2:
Shell and tools
Randall Howard <rand@mks.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Background on POSIX.2
The POSIX.2 standard deals with the shell programming language and utilities. Currently, it is divided into two pieces:

• POSIX.2, the base standard, deals with the basic shell programming language and a set of utilities required for application portability. “Application portability” essentially means portability of shell scripts and excludes most interactive features. In an analogy to the ANSI C standard, the POSIX.2 shell command language is the counterpart of the C programming language, while the utilities play, roughly, the role of the C library. POSIX.2 also standardizes command-line and function interfaces of some POSIX.2 utilities (e.g., popen(), regular expressions, etc.) This document is also known as Dot 2 Classic.
• POSIX.2a, the User Portability Extension or UPE, supplements the base POSIX.2 standard; it will become a non-normative (optional) chapter of some future draft of the base document. The UPE standardizes commands, such as screen editors, that might not appear in shell scripts but that users must learn on any real system. An interactive standard, it attempts to reduce retraining costs incurred by system-to-system variation.

For utilities that have interactive as well as non-interactive features, the UPE defines extensions from the base POSIX.2 utility. One example is the shell, for which the UPE defines job control, history, and aliases.

In my previous report, I noted two serious strategic problems with the UPE:
• lack of coherence, and
• lack of support.

The problems haven’t disappeared. (See the previous report for further information.)

Features used both interactively and in scripts tend to be defined in the base standard.

Status of POSIX.2 Balloting

Dot 2 Classic remains in its second round of balloting on Draft 9. Hal Jespersen, the POSIX.2 Technical Editor, thinks the forthcoming Draft 10 will go to ballot in June or July. Some early subsets of Draft 10 were in evidence at the working group meeting, but circulation is still restricted to those feeding changes into the Technical Review Process (so, no, you won’t be able to get one yet). Draft 10 involves fewer people than the ten or so technical reviewers that produced Draft 9. On one hand, fewer people mean fewer ulcers for Hal Jespersen, who must co-ordinate myriad changes from as many quarters. On the other, too few people produces a closed process, which extends the number of rounds of balloting required for final resolution.

Because the first round of balloting (Draft 8) produced many objections that were only partially resolved by Draft 9, and because issues often have several sides to consider, the Draft 10 balloting, starting this summer, has only a slim chance of creating the final standard. That said, Dot 2 Classic’s contentious areas appear to be narrowing to a small set of new inventions (create, hexdump, locale, localedef, etc). I expect the objections to Draft 10 to be far fewer, and that the process is likely to converge to a full-use standard by Draft 11, late in 1990 or early in 1991.

On the UPE front, Draft 4 is scheduled to appear in February or March, so that a mock ballot may be held for the April meeting. A mock ballot is a rehearsal for the real ballot – real comments and objections are both prepared and resolved by the working group. A real ballot shifts the focus from the working group to the balloting group. The mock ballot is an excellent exercise, but communications within the working group tend to be excellent. The process becomes more obscured once formal balloting begins, as shown by the extended balloting on Dot 2 Classic. Nonetheless, having distinct balloting and working groups ensures that the process has comments from all parties.

Formal (real) balloting for the future Draft 5 of the UPE should commence this fall. A much smaller standard, the UPE is approaching the level of review that Dot 2 Classic had before it entered formal balloting.

Status of the New Orleans Meeting

Apart from a status report on the balloting of Dot 2 Classic, the New Orleans meeting focused on readying all UPE utility descriptions for mock balloting. The working group reviewed existing utility descriptions in small groups of between three and six persons. One group spent much of the week fleshing out arcane details of vi, only occasionally relieved by work on simpler utilities.

A group I worked in made the surprising discovery that uuencode, a utility traditionally used to convert binary files to a printable form to pass through mailers, is a utility to “encode a binary file into a different binary file.” This complexity arises from internationalization, where there are always several ways of looking
at any problem. Delve deeply into POSIX and ANSI C internationalization issues, and you'll always discover topics that the committees have not yet dealt with. This is not a criticism of the internationalization standardization groups; much work is still needed and solutions to many problems remain elusive. In the uuencode example, we felt the output of uuencode should be code set invariant (i.e., uuencode on an EBCDIC system should produce the same results as uuencode on an ASCII or ISO 646 character system). To achieve this, ' ' through ' through ' must be expressed as 0x20 through 0x5F and "begin" must be expressed as 0x62 0x65 0x67 0x69 0x6E (the hex equivalents of 'b' 'e' 'g' 'i' 'n' in ASCII). POSIX appears to offer no standard way to convert a file from one code set to another.

Attendance at the UPE working group was, again, relatively small - around a dozen people. One reason is PAR proliferation. Most companies cannot afford to send one committee member to each working group. (I, for example, also had to cover TFA, POSIX.1b, and the internationalization efforts.) [Editor: Readers should note that that being spread thin didn't stop Randall from turning out a clear, thoughtful report. Thanks, Randall.] Another reason is that there is less enthusiasm for the UPE than for Dot 2 Classic. Even Hal Jespersen has said that "...basically the NIST put our feet to the fire to do the UPE."

Some people want the UPE to include an EMACS editor description as well as one for vi. Unfortunately, although there was talk of an EMIN proposal, none was submitted to the working group. If you EMACS fans want it included in the ever-expanding UPE, then submit a proposal. [Editor: Listen up, folks. He's serious.] (Of course, some devotees feel that standardization would be inappropriate for an extensible environment like EMACS.)

"Revision/Source Code Control Software" is a much-shuffled area of future standardization within the overall POSIX.2 PAR. Fearing another Tar Wars-like clash between fanatic supporters of of SCCS and RCS, the topic was removed from Dot 2 Classic and deferred to the UPE. The Source Code Control System (SCCS) is the original UNIX source code control system which was implemented in the mid 1970's, modeled after mainframe systems of the time. The more modern (no bias here...) Revision Control System (RCS), by Walter Tichy of Purdue University, claims to have improved on SCCS. Each has its proponents; SCCS appears to have a stronger following because of commercial support by vendors, but RCS appears to have a more devoted underground following. The working group is divided between those who want either SCCS or RCS and those who want neither, arguing that source control is a vendor-specific application. Unfortunately, the UPE working group has had problems resolving the controversy and Hal Jespersen has proposed that POSIX.2c (yes, you heard it right, .2c) be assigned as a PAR for working on this topic. (What happened to .2b? POSIX.2b is the working group that will prepare revisions and clarifications of Dot 2 Classic – which isn’t even finished balloting.)

Report on IEEE 1003.3: Test Methods

Doris Lebovits <lebovits@attunix.att.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Dot three's job is to do test methods for all of the other 1003 standards. This was the working group’s fifteenth meeting. We reviewed the ballot status of P1003.1 test methods, worked on P1003.2 test methods, and created a steering committee.

Review of ballot status and Dot two verification

The P1003.3 standard will consist of several parts: Part I is generic test methods, and part II is test methods for measuring P1003.1 conformance, including test assertions. Part III of P1003.3 will contain test methods and assertions for measuring P1003.2 conformance. As other P1003 standards evolve, they will be covered as separate parts in the P1003.3 standard.

Each day was divided into two sessions: mornings, we did technical review of parts I
and II, afternoons were spent writing assertions for part III. AT&T, NIST, OSF, Mindcraft, IBM, DEC, HP, Data General, Cray Research, Unisys, Perennial, and Unisoft Ltd. were represented. [Editor's complaint: I see no user representation at all.]

It took twelve meetings of the previous P1003.3 working group to prepare the draft that is now balloting. The technical review for the Draft 10 ballot was completed. Draft 11 was re-circulated late February 1990 and closed March 23, 1990. The balloting group is approximately ninety members. X/Open submitted a list of assertions for P1003.1a. This list was included as an appendix to Draft 11. Balloters were expected to review this appendix as part of their ballot. We anticipate an approved P1003.3 standard in the third quarter of 1990.

This is the third meeting for developing a verification standard against the P1003.2 standard. The P1003.2 assertion writing and review were done in small groups. Some of the assertions were based upon P1003.2 Draft 9.

A steering committee and some new officers.

The chair, Roger Martin, instigated the creation of a test-methods steering committee to help alleviate the increasing dot-three work load all the other proliferating groups are creating. The committee will coordinate the activities of all test-methods groups, monitor the groups' conformance to test methods, and write and approve Project Authorization Requests (PARs). Membership will be dynamic, limited to four to six, and new members will be chosen based on long term commitment, new ideas, and technical/managerial skills. Roger suggested an initial makeup - Roger Martin (NIST, Steering Committee Chair), Anita Mundkur (HP), Andrew Twigger (Unisoft), Bruce Weiner (Mindcraft), and Lowell Johnson (Unisys) - and the working group approved. It's a non-controversial mix of established P1003.3 members.

The Standards Executive Committee (SEC) has approved both the committee and its membership. Their first assignment is to document procedures.

In addition, new officers were chosen for the P1003.2 Test Methods activities. Ray Wilkes, of Unisys, is Chair, Jim Moe of Cray Research, is Co-chair, Lowell Johnson of Unisys is Secretary, and Andrew Twigger of Unisoft Ltd is Technical Editor.

Report on IEEE 1003.4: Real-time Extensions

Rick Greer <rick@ism.isc.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

1003.4 goes to ballot

The big news in 1003.4 is that some of it is ready for balloting. The current draft is a 330-page, eclectic collection of real-time features. Some (e.g., asynchronous event notification) address significant deficiencies in the dot-1 base, but others (e.g., IPC message passing) seem to be of limited value. It remains to be seen whether the limited applicability of some of the proposed features is enough to shoot down the entire ballot.

One area that may cause trouble is the shared-memory model in the Language-Specific Requirements section. While this language-independent model addresses a real need - serialization of reads and writes in the presence of simultaneous updates to a common store - it does so rather formally; people uncomfortable with formal mathematical models may be put off by it. The fact remains, however, that both dot 1 and the ANSI C standard failed to address this problem, which is critically important in shared-memory multiprocessor architectures.

Threads

The threads proposal is only an appendix in the current draft, and won't be subject to formal ballot. Though there were too many loose ends in the threads proposal to send it to
ballot in this round, most of them were tied up in New Orleans. We should have a ballotable draft ready after the April meeting.

Meanwhile, the active membership in the threads "small group" is changing. Representation from the Ada community has grown from two to six; almost a quarter of the active membership is now familiar with Ada and its multitasking model. Most threads people, including me, are also becoming active in the new multiprocessor study group.

Discussion within the multiprocessor group promises to be quite lively, since the threads group's more contentious issues (e.g., signals) were skirted by defining high-level interfaces, leaving details of low-level behavior unspecified. The multiprocessor group, on the other hand, must deal with the low-level behavior of multiprocessor configurations, and many of the old arguments have already resurfaced (e.g., should signal state be maintained per-process or per-thread?). Using high-level interface specifications to dodge low-level implementation issues does have its problems, though. People unaware of more subtle implementation issues tend to view new high-level interfaces as unnecessary complications. It's difficult to convince them that, even if consensus could be reached regarding the behavior of primitive functions, we would still need high-level interfaces (or rigid coding disciplines) to guarantee that independently developed routines use primitives consistently when addressing common problems. The real sticker here has been how to asynchronously terminate a thread and cause it to execute cleanup code. Everyone agrees that this is necessary. Some members, particularly those from AT&T/USO, feel that the best way to provide this facility is by minor enhancements to traditional UNIX signals, but most of the group feels that the best way to deal with notorious signal races in a uniform, language-independent manner, is to adopt a high-level interface, modeled after one used by DEC/SRC.

1003.4 turns into .4, .4A, .4B, .4C, and .14

There are three other major, ongoing efforts in dot 4: language-independent specification of the real-time extensions, identification and specification of other important non-threads, real-time extensions that didn't make it into the current ballot, and specification of a real-time application profile. The first is farthest along, but none is anywhere near completion. Recognizing that these efforts were separate from the current proposal, and from one another, the working group submitted four new Program Action Requests (PARs). The Sponsor Executive Committee (SEC) approved all four, and decided that the application-profile effort was so distinct that it needed a new number. The working group's five PARs are now:

- the current ballot 1003.4
- threads 1003.4A
- language independence 1003.4B
- further real-time extensions 1003.4C
- real-time application profile 1003.14

Report on IEEE 1003.7: System Administration

Martin Kirk <mkirk@axion.bt.co.uk> reports on the January 8-12, 1990 meeting in New Orleans, LA:

The System Administration working group is developing portable interfaces for administering computer systems, which will provide traditional systems-administration functions such as managing users, file systems, and devices.

The working group began with a base document similar to the draft System Administration FIPS produced by NIST in September 1988, containing a set of commands based on existing functionality. It addressed only the single machine case, and the group quickly saw that it formed an inadequate basis for extension to networked systems.
Three competing models were advanced to cope with heterogeneous networks. All three assumed that there would be a standard interface, but differed in the scope of the underlying administrative database and the degree of interoperability. To update a network of 100 systems, supplied by five different vendors, the three models had:

1. one database per system, requiring any operation to be performed 100 times
2. one database per vendor, requiring each operation to be performed five times
3. one database for the entire network requiring each operation to be performed only once.

The working group chose Model 3, which offered the greatest interoperability, the most benefit, and the biggest technical challenge. The working group also chose an object-oriented approach. [Editor: USENIX can take some credit for this, having prepared a whitepaper that recommended precisely this approach.]

Because system administration is closely related to network administration, in that both are concerned with managing objects distributed across a heterogeneous network, the group adopted an object template based on the work of the OSI Network Management Forum. The template uses Abstract Syntax Notation One (ASN-1), to specify the attributes and characteristics of objects.

Currently, the group's major task is to develop object class definitions. Some of the object classes, such as the user object class, seem relatively straightforward, with attributes such as login-name, numeric-uid, group-id, home directory, and login shell. Others, such as the device object class, introduce major questions: How far is it appropriate to go in defining sub-classes such as disk-devices and tape-devices?

The standard will not specify implementations. Information about a user can be stored in whatever fashion seems appropriate: in a traditional place, such as /etc/passwd, or in a database.

When the object-class definitions are complete, the next task will be to specify both a command-line interface and a programmatic interface to manipulate the objects. The latter will have both a language-independent specification and a C-language binding. All objects will support a core set of four operations – create, delete, set-attribute, and get-attribute – and probably a fifth to check consistency. In addition, there will be operations specific to particular object classes, such as a mount operation for file systems.

I am happy with the general approach, but there may be trouble ahead on the command interface front. At present, this is the canonical form:

<object> -o <operation> <attributes>

such as

user -o add name=jsh,uid=423,group=editors

or something of that general style. I expect that there will be complaints once it sinks home that this removes old favorites such as "mount" from the system administration canon.

Though the standard is designed for heterogeneous network administration, the working group has not really tackled interoperability. Someone must address this critical area, but it may ultimately be the IEEE TCOS networking groups.

Dot seven is currently aiming at a mock ballot in 1991, and a full ballot in either 1992 or 1993.

Disclaimer: The views contained herein are my personal opinions and do not necessarily have any relation to those of my employer.

Report on IEEE 1003.8: Transparent File Access

Jason Zions <jason@cnd.hp.COM> reports on the January 8-12, 1990 meeting in New Orleans, LA:
1003.8 breaks up
The networking work has been reorganized; what was one committee is now five. At this meeting, the Sponsors’ Executive Committee (SEC) approved all the networking Project Authorization Requests (PARs) and forwarded them to the IEEE Standards Board for final approval. In the past, 1003.8 was responsible for half-a-dozen types of networking issues. From now on, 1003.8 will restrict itself to transparent file access (TFA); the other work will be distributed to four new groups. The new structure is:

1003.8 TFA
Transparent File Access
1003.12 PII or P2P
Protocol Independent Interfaces, or Process to Process
1003.13 RPC
Remote Procedure Call
12xx PDI
Protocol Dependent Interfaces, a.k.a. s-1OSI
FT AM and ACSE
12yy NS/DS
Name Spaces and Directory Services, maybe X.500

The SEC tentatively assigned 1200-series numbers to NS/DS and PDI, because they intend these standards to apply to any operating system, not just one that's UNIX-like. (There's one exception: NS/DS must identify the name spaces required by the 1003 standards and determine some means of managing them.)

TFA decides what to do about NFS
The meeting was a landmark for TFA. Until now, no consensus on overall direction had been achieved. We spent a great deal of time discussing the philosophy and goals for a Full TFA and Subset TFA, but no common understanding had been reached in the minds of all members; we wandered between extremes of, “Full means 1003.1!” and, “But NFS sure seems to be good enough for users; after all, they're still buying it.” It became clear that some agreement had to be reached for progress to be made. Many TFA attendees had never worked on a POSIX committee before and didn't quite understand the POSIX consensus process, but after a joint meeting of 1003.1 and TFA, the exact scope and structure of work were finally hashed out. The group's work items are described below.

1. Full TFA
This piece will contain minor additions and changes to 1003.1-1988 to specify its behavior when operating on remote files. Work will include extending already-defined interfaces (e.g., new stat() information), defining new errors, defining failure and recovery semantics, and so on.

Semantically, a remote file accessed under Full TFA will be indistinguishable from a local file. A strictly conforming POSIX application will run completely unaltered in a Full TFA environment.

2. Subset TFA
This piece will define both a core subset of 1003.1-1988 that can work correctly over a variety of remote-file-access protocols (“the Core”) and a number of additional optional feature sets. The specification will form additional text for IS 9945-1 (ISO’s version of 1003.1).

The intent is to have Subset TFA work on the widest variety of protocols consistent with a useful Core; if a remote-file-access protocol is so constraining that any Core based on it would be too small to support useful applications, it will be excluded.

FTAM, the International Standard File Transfer and Access Method (IS 8571), will shape decisions about what will go into the Core, for a variety of reasons.

• It is the weakest common mechanism for remote file access.

• The standard has little chance of success at the ISO level unless it is clearly cognizant of FTAM.
Nothing weaker than FTAM is likely to prove useful to application writers.

People are clamoring for a simple interface to FTAM; the open/read/write/close style of Subset TFA meets that need.

The difference in functionality between the Core and Full interfaces will be divided into blocks of capabilities (the "feature sets" mentioned above), which might be provided by other commonly used file-sharing mechanisms. A Core-conforming application will be able to inquire (via pathconf) what functionality, over and above the Core, is available on a per-file basis, and alter its behavior accordingly.

The Core will meet an expressed need to know "what doesn't work right" over common file sharing protocols. For example, Sun might define NFS's functionality in terms like, "NFS provides Core Subset functionality, plus the _PC_LOCKING, _PC_DIRECTORIES, and _PC_TIMES capability sets." An application programmer could use such a specification to determine exactly what features of 1003.1-1988 were safe to use in an NFS environment.

This scheme also permits continued development of remote-file-access protocols. Any mechanism that supports at least the Core will conform to the standard. This encourages vendors and researchers to develop mechanisms that combine the Core and its options with other advantages (very high performance, very high robustness, good behavior over WANs, etc.), while giving users a well-defined interface for applications that will work in all such environments.

3. A Data-Stream Encoding (DSE) supporting the Full TFA Interface

This will provide the mechanism necessary for interoperation of client and server systems. 1003.8 will only develop the encoding; no binding to any particular protocol stack or suite is planned. (Such bindings will be done by working groups chartered to develop profiles to satisfy particular needs.)

Work on the DSE will probably not begin for at least another six months. There are now two existing proprietary mechanisms that provide the appropriate functionality: SVR4 RFS and the Andrew File System (AFS v.3 from Transarc). The committee hopes at least one (if not both) of these products' DSEs will be released to POSIX for standardization. If both are, there will probably be a gun-battle over which to base the standard on.

There was good progress on the first two work items. The group hopes to have a meaningful draft available by April, and would like to go to ballot by the end of the year. This quick ballot will help compensate for the small working group by bringing major ballot objections to the surface early. (Much coordination with other 1003 working groups, especially 1003.1, will also help.) The balloting process will probably be quite lengthy: on the order of 12-15 months.

Report on IEEE 1003.11: Transaction Processing

Bob Snead <bobs@ico.isc.com> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Context

Our charter is to develop an application profile for POSIX Transaction Processing (TP). We're wrestling with both the content of our profile and the idea of a profile, since profiles are new to POSIX. [Editor: Jim Isaak reviewed application profiles in the February issue of IEEE Computer.]

The content is influenced by two other TP efforts: OSI's DTP and X/Open's XTP. We must handle OSI DTP, just to gain ISO acceptance — a goal of all the 1003 efforts. In theory, XTP is just another proprietary concern. In fact, XTP's ongoing deliberations are currently confidential. Moreover, X/Open isn't an official standards body so we can't officially reference XTP in our profile. Nevertheless, XTP will carry considerable weight, since it will be a multi-vendor consensus on how to do UNIX TP.
Models

As at previous meetings, we spent much time discussing TP models. For the most part these discussions were based on a snapshot of XTP's model released to non-X/Open members some time ago. Each model we discussed consisted of three or four of the following elements: Application Programs (APs), Resource Managers (RMs, like database managers), Communications Managers (CMs, like TCP/IP), and Transaction Managers (TMs, which enforce the transaction protocol among APs, CMs and RMs). Here, in chronological order, were the major topics of discussion.

We discussed whether a CM might just be an instance of an RM (viewing an instance of a communications protocol or link as a resource), but concluded that attributes of CMs make them fundamentally different beasts (though, to be honest, it's still not clear to me why).

We considered several models based on XTP, but differing from one another in the roles of the CM and the interfaces between the AP and CM. We concluded that each communications protocol would have to have its own CM, and that our model must support multiple concurrently active CMs. A CM, though, is more than just its protocol support. It has to include support for additional functionality required for DTP. We never concluded whether or not an AP should talk directly to a CM, or to a CM via the TM.

Requirements

In the course of the model discussions, it became clear that many of us had different requirements in mind, so we shifted our focus to requirements to try to reach some consensus. Ultimately, we decided that POSIX TP must:

1. be mappable onto OSI DTP,
2. support global (distributed) transactions,
3. support chained and unchained transactions,
4. support a conversational mode,
5. provide data conversion (e.g., ASN.1),
6. ensure that POSIX RPC supports DTP semantics,
7. ensure that DTP can be accomplished through RPC,
8. provide for location independence via directory services, and
9. provide for security of data.

Exercises

We decided to break the modeling deadlock by focusing on the AP/TM interface and ignoring communication. We worked several examples, following ISO DTP services but using an RPC paradigm, and concluded that an API based in RPCs would need at least four services:

- one for a caller to start a transaction,
- one for a callee to find out if it is participating in a transaction,
- one for a callee to abort a transaction,
- one for a caller to commit or abort a transaction.

We also identified the following assumptions for TP via RPC:

- A thread of control (TOC) can be in at most one transaction at any given time.
- If one TOC communicates with another, the latter joins the former's transaction by default.
- No nested transactions are permitted.
- A GTRID (Global TRansaction ID) can be associated with multiple TOCs and multiple RMs.
- A transaction has only one initiator and only the initiator can issue commit. Any TOC may abort.
Report on IEEE 1003.12: Inter-Process Communication

Steve Head <smh@hpda.HP.COM> reports on the January 8-12, 1990 meeting in New Orleans, LA:

Overview

P1003.12 is the IEEE POSIX Network Inter-Process Communication (IPC) committee (formerly P1003.8/2). The committee is currently working on two potential interfaces: a detailed interface (DNI) and a simple interface (SNI).

At this meeting, the group arrived at a high-level description of a name-to-address translation facility, and decided the question of XTI versus sockets versus "something else" in favor of "something else." The group began discussing connection setup, and continued discussing SNI. Finally, the POSIX steering committee (SEC) changed the group's name to P1003.12.

There were about twelve attendees.

Detail

1. SNI reviewed

A UC Berkeley SNI proposal is gradually taking shape. The proposal describes both objects and functions that act on them. Some of these objects and functions have analogues in the socket world, while most of the others are composites.

The most recent additions are sni_save() and sni_restore(). sni_save() takes a snapshot of an endpoint and saves it in a string, suitable for passing to a child process through an argument or the environment. sni_restore() restores the library state of an endpoint from that string.

The committee has had two goals for SNI. For naive users, it should simplify the networking interface. For vendors, it should allow implementation of interfaces over complex protocol stacks (such as ACSE – or something above ACSE – over OSI-7).

One issue that came up was what the application programmer would target for. If DNI and SNI retain distinct differences, SNI-based applications risk outgrowing SNI's capabilities. One alternative would be to combine DNI and (the current) SNI to allow seamless expansion into protocol-specific hooks, without recoding of applications.

Next meeting, UNISYS is expected to present an alternative SNI proposal.

2. Naming

The group discussed name-to-address translation for DNI in detail, specified an interface at a high level, and intends to pass it to the naming group. The specification is:

```plaintext
given:
hostname/"entity"
service/"facility"
type/"context"
protocol or protocol family

return:
set of {
  address
  any input parameters that were
  completely or partially wild-carded
}
```

SNI might need something similar, but without the protocol / protocol-family / address-family parameter. (SNI is protocol-independent.)

The interface lets applications defer deciding which protocol- or address-family to use until after the query. It will also permit load-balancing, a technique to optimize data-transfer performance over slower interfaces (such as multiple serial point-to-point links).

The group deferred discussing both performance (time and memory) and which input parameters could be wild-carded.

3. XTI versus sockets

The XTI versus sockets issue came up briefly while discussing passive-endpoint functions. The group resolved to incorporate the best of XTI, sockets, and possibly other extensions, into DNI.
The group decided not to require full XTI-type functionality, and accepts the risk that porting XTI-based applications to DNI may require source code changes. A potential advantage of this decision is that the standard can leave out the mistakes of XTI and sockets. Also, vendors remain free to supply the older interfaces on the side.

A UCB representative will prepare a new DNI proposal between now and the next meeting.

4. P1003.8/2 → P1003.12

The SEC gave network IPC its own separate number: P1003.12. This change will be formally approved at the IEEE standards board meeting, a couple of months from now.

5. Potential overlaps with P1003.4

For several meetings, both P1003.12 and P1003.4 have been aware of their potentially overlapping coverage of process-to-process communication on a single, local system. Since there should be only one interface for common functions, and any characteristics peculiar to local IPC can be supported by protocol-specific options under DNI, P1003.12’s position is that it should handle all IPC. The group has asked the networking steering committee chair, Tim Baker, to relay this position to the SEC.

Future Meetings and Significant Dates:

The Spring 1990 meeting will address SNI/DNI connection setup/tear-down and SNI/DNI data transfer.

Report on IEEE 1201: User Interface

Peter H. Salus <peter@uunet.uu.net> reports on the January 8-12, 1990 meeting in New Orleans, LA:

What’s happening?

P1201 purports to concern itself with the user interface. As of the New Orleans meeting, P1201 comprised .1 (Applications Programming Interface), .2 (Graphical User Interface), .3 (Human-Computer Interaction), and .4 (XLib) subgroups.

Working backwards through these, 1201 has recommended that XLib go to ballot directly, a proposal which seems to have so shocked the SEC that they put off deciding on balloting till April. Steve Jobs told the USENIX audience in Phoenix, in June 1987, that X was “brain-damaged.” Whether that’s true or not, X has won, and just putting XLib to a vote makes good sense.

1201.3, under the chairmanship of Richard Seacord, has had a number of interesting discussions and presentations (of which I attended several, though not all). The major problem here is that we are nowhere near knowing what the “standard” for an interface might really require. However, the explorations are valuable, and a forum like this can be informative.

This leaves me with the GUI and the API. Both in Brussels and in New Orleans were skirmishes in the GUI wars: battalions of employees of OSF and its member companies arrayed in opposition to those of UI or USO and theirs, with a pair of observers from NeXT and Apple taking and placing bets on the sidelines.

I assure readers that have never attended these meetings that acrimonious backbiting and vituperation are the order of the day in both camps. Though a former employee of OSF, I wouldn’t hesitate to condemn the behavior of both sides, but the blame rests elsewhere. Where? In the tourists. See below, but for my money, too many folks like to travel and too many people have caught the “open systems/open standards” bug.

So long as the market remains unsettled about Motif, NeXTStep, OPEN LOOK, and Presentation Manager (to say nothing of Apple’s MacIntosh interface and IBM’s CUA) [Editor: That’s “Common User Application”, a part of SAA], the meetings of 1201.1 and 1201.2 will serve as tilting grounds, not occasions for useful discussion.
From my point of view, until the market (which means the big boys and the users) has a shake-out, .1 and .2 can only serve as debate platforms or end up recommending standards that are either the intersection of OPEN LOOK and Motif or their union. It might be that .2 can come to some sort of conclusion on the various style guides without .1, but I see the products being waved, not the function banners.

Why is it turning out this way?

All of this is prologue ("The past is prologue," writes Shakespeare in The Tempest) to a commentary on the TCOS-standards industry. [Editor: TCOS, the Technical Committee on Operating Systems, is the IEEE organization under which both 1201 and 1003 fall.]

Over the past 40 years, ISO has approved or accepted over 20,000 standards, which concern almost everything imaginable from hockey masks to medical prostheses to the hinging of radar masts on inland-waterway vessels. The standards have arisen in a variety of ways, most emanating from one of the regional or 70-odd national standards bodies. Typically, it has taken from four to ten years to progress from raising a committee to approving a standard. The result of this has been general agreement within the concerned industry prior to the issuance of an international standard. Wall plugs are an excellent example of what happens when the engineers and bureaucrats issue a standard without industry consensus.

I am far from convinced that the ever-increasing number of 1003 and 1201 (sub)committees is productive or useful, and embarrassed and appalled at their continuing proliferation. There are currently at least six or seven standards for diskettes. Do we really need that many for graphical user interfaces? I think not. Might we get what happened in the record industry (i.e., 45s for short cuts; 33s for long works and anthologies) if we wait? I think so.

Moreover, does the standards process really require more than two or three folks per company? There were 38 in attendance at the ISO/IEC Joint Technical Committee on Application Portability meeting in September (including the secretariat); there were nearly 300 in New Orleans. My perception is that going to a POSIX meeting is a perk. Holding the meetings in Hawaii, New Orleans, and Snowbird does little to dissuade me. The New Orleans host was OSF; the Snowbird host is Unisys. Though the new Unisys is a big entity, I didn't realize they had a site in Snowbird; nor OSF one in New Orleans.

C'mon, lets get back to work, not meetings for the holiday or for the sake of meetings. 1003.1 did good, solid work. Some of the other groups are doing work, too. Partying ain't part of it. Bah!

Report on ANSI X3J11:

C Programming Language

Doug Gwyn <gwyn@brl.MIL> reports on the state of ANSI C:

There is now a C standard

After the one appeal of CBEMA X3's approval of the proposed ANSI C standard was eventually voluntarily withdrawn by the appellant, the ANSI Board of Standards Review approved the proposed standard on December 14, 1989. (CBEMA is the Computer and Business Equipment Manufacturers' Association, the organization that sponsors X3.)

No appeals were received by ANSI within the time allotted, so there is now an official American National Standard for Programming Language C: ANSI X3.159-1989. The technical content of the ANS is identical to that of the December 1988 X3J11 draft.

The X3J11 technical committee will enter an "interpretations" mode at the March 1990 meeting in New York City. During this phase, the committee will be considering requests for clarification and interpretation of the standard. It is anticipated that Technical Information Bulletins will be issued from time to time when it is felt that clarification of the intent of
the standard needs to be published. Such bulletins would not technically be considered part of the official standard; however, they should provide valuable guidance to both C implementors and C programmers.

USENIX Standards Watchdog Committee Update

Jeffrey S. Haemer <jsh@ico.isc.com> reports on winter-quarter activities of the watchdog committee:

1003.0: A Guide to POSIX-Based Open Systems

Dot zero, the POSIX guide group, continues to suffer from bureaucratic inertia. It complains that its forty or so attendees are insufficient to allow rapid progress, yet in a year-and-a-half they've just created a table of contents. Some people think this reflects badly on the group. I think this is completely wrong.

Admittedly, the economics of producing the POSIX guide itself are unfavorable. A large fraction of the attendees are highly-placed or key employees of large corporations and influential organizations. A back-of-the-envelope calculation puts salary expenditures alone, for each one week dot zero meeting, close to six figures. Had the committee delegated the entire task to one or two full-time people, it would be done. The fine overviews UniForum occasionally publishes are proofs by example.

How, then, does dot zero benefit the user community? The meetings give influential people from the most important corporations in the commercial UNIX arena a way to get together in the same room (or after hours in the same city) and discuss the direction of UNIX without risking an anti-trust suit.

USENIX meetings serve a similar purpose for more technical segments of the UNIX community. To some degree, UniForum meetings serve an analogous purpose for other segments of the industry. But where else is there such a concentration of high-level, UNIX vendor management except, perhaps, at meetings of the Hamilton or Archer groups, or of the board of directors of X/Open? Attendees support POSIX, and influence their companies to become involved. Because POSIX is a good thing, so are dot zero meetings.

1003.1: System Services and C Language Binding

Dot one is well ahead of the rest of 1003; look here to see the future. The initial standard is done, published, and government-approved as FIPS 151-1. The group is now working on supplements, which come in two flavors: nit-picks and corrections (1003.1a) and real additions (1003.1b). But to speak of "the group" is misleading; these two working groups have a strikingly different makeup from the group that created dot one. Many who were passionately and intimately involved in the production of the Ugly Green Book have moved on, either to other committees or out of the standards game. The working groups are now small numbers of hard-core, dot-one devotees. For .1a, this isn't a problem - that's exactly the kind of person needed for nit-picking.

Watch .1b like a hawk, though. Any new functionality, slipped into supplements and appendices, carries the same risks as riders on congressional bills; if it can be slipped in unobtrusively enough, or with the right timing, it can be awful and still ride on the coattails of the main body. Bad deeds done here will both inflict irresistible harm, and diminish the credibility of dot 1.

I recommend resisting any effort to add functionality for which there aren't existing implementations in wide use, and about which there isn't already general consensus. Design-by-standards-committee efforts should be deferred to other more ignorable standards.
1003.2: Shell and Utilities

Dot 2 is still firmly in the dot one mold. Dot 2 Classic is balloting away, and should soon be both done, government approved and FIPSified, with a set of test assertions that companies like Mindcraft can sell test suites for. When this is done, a large number of systems will advertise compliance with 1003.1, 1003.2, and X3.159 and provide, for most users, a standard "UNIX."

Even the controversial UPE is mostly codifying existing practice. Arguments are over places where more than one practice is widespread, for example, source-code control, where partisans of SCCS struggle with partisans of RCS. (Actually, that's not true. What's really happening is that the group's shying away from this area because they're worried about a struggle. "Tar wars" seems to have spoiled the industry's appetite for making difficult decisions about contentious topics.)

Parenthetically, I'll admit to being mystified by the dim view some folks take of the UPE. I actually put programmer portability above program portability, since, when I go looking for new jobs I can't take my software with me, but do want to be sure that I can still use vi. (Of course, most members of working groups are sponsored by vendors.)

The equivalent of .la already has a name: .2b. Even the bad of dot one is mirrored here. Truly controversial proposals are being pushed off to the as-yet unborn .2c, which should produce a deja vu feeling in those already watching .1b. ("But," you remark, "you always say that.") And, just as .1 sometimes shied away from real decisions, in order to avoid upsetting anyone, .2 occasionally reacts to vendor inconsistency by proposing solutions that avoid upsetting any vendor by penalizing all users. As an example, the committee proposes requiring a C compiler (good), and naming it c89 (bad, but I complained about this loud and long last time). An important motivation for the new name is that cc already invokes the K&R C compiler on many vendors' platforms, and specifying a flag to choose one behavior or the other would conflict with someone's existing implementation; any given letter is already preempted by some vendor.

I'm not convinced by this argument. I have consulted the Ouija® board in my office, normally used only for project scheduling, and will now predict the effects of this sidestep, if approved:

- In two years, everyone will have a c89 compiler, to comply with a government FIPS. Shell scripts and makefiles will continue to invoke cc, but be less portable than they are now.
- On a few conformant machines, there will be no cc command. This will break an enormous number of programs, and solutions will vary from user to user, project to project, and installation to installation.
- On other machines, cc will produce one flavor or the other. Most, but not all, machines will link cc to c89. This will break a variety of things, but not consistently enough to allow a portable solution.
- On some of these machines, flags will make c89 compile K&R C. The flag will vary from vendor to vendor.

In short, we who do ports will have to keep track of how to invoke the C compiler on each of our target machines; .2 will not have enhanced portability in this area of our work.

Finally, like .1, my unease over a small number of problems stands in stark relief to the generally high opinion I have of the work done by this group.

1003.3: Test Methods

Dot three, a tiny mirror of the overall POSIX effort, is proliferating because it has no choice. It will now have a subcommittee to develop test assertions for each of the other POSIX efforts, and has acquired a steering committee to oversee the subgroups. Whether this is a better choice than having each POSIX committee develop its own test assertions isn't clear - I see plusses and minuses for each approach. Still, all in all, the group seems to know what it's doing, and is willing to do it.
Dot three isn’t always popular; one hears complaints that they come up with interpretations that seem contrary to the intention of the original standards committees. On the other hand, that seems as good a reason as any for their existence. They form a combination system-test and quality assurance group for the other committees, generating all the friction one expects from any such organization.

A dot three member did take the time to divulge an unexpected answer to a question I raised in my last report – what motivates someone to be in dot three? For a few folks, it’s obvious: MindCraft employees attend because their company develops and sells test suites. Others are also there because they’re really interested in testing. But think: if you want an overview of all of POSIX, what group should you attend? There are three candidates: dot zero, but then you’d have to buy an expensive wardrobe; the SEC, but that group is mostly institutional representatives, officers, and overworked committee chairs; or dot three, which examines each standard in detail as it nears completion. If you’re thinking of joining a working group, and want this sort of vantage point, I’m certain the group has plenty of work to hand out.

1003.4: Real-Time Extensions

The real-time group now has five PARs: .4, .4a, .4b, .4c, and .14. The first of these went to ballot after the New Orleans meeting. Threads, controversial enough to be omitted from .4, has been pushed into .4a. (Things too controversial to go into threads will be pushed into the multiprocessor group, which should be a lot of fun.)

(The remarks below in brackets and with -SP are taken from a response posted to comp.std.unix by Simon Patience of the Open Software Foundation; see also the article of comment by John S. Quarterman that follows. -Ed.)

[This is not actually true. Pthreads was never in the draft of 1003.4 proper but was an appendix. After New Orleans when .4 was ready to ballot, pthreads was not and so could not become a real chapter of its own within .4 and so got its own PAR. It had nothing to do with being controversial. Your parenthetical comment is pure fantasy also. -SP].

The threads subgroup (1003.4A) has attempted to kill the .4 ballot by a block vote for rejection. One correspondent says they are doing this because .4 is no good without threads. (I’m told that two “large, non-vendor organizations” are part of the coalition against the 1003.4 ballot. There is rumored to be a special, invitation-only, threads-strategy meeting by these two groups immediately preceding the Utah meeting. Can anyone confirm this and supply more details?)

[More misinformation here. The Common Reference Ballot was written by a number of people from different organisations some of whom attended the threads group and some didn’t. The endorsements for it came from a significantly wider audience than the threads group, some of whom I believe have not been to a .4 meeting either, or at least regularly. The objections were not related to threads except where an interface was impossible to be used in a multi-threaded environment.

The rumor of a pre-Utah meeting is completely overblown. OSF and UI regularly meet, with representatives of our respective member organizations, to discuss technical matters to try and maximize commonality between our two systems, especially at the interface level. The subjects include threads as this is an emerging technology area, but it is certainly not restricted to threads. As the people involved in this also attend POSIX meetings, it is natural to take advantage of the fact that we are all going to be in the same place. The meetings take place regularly and more frequently than POSIX meetings. We think this level of cooperation is the sort of thing the industry would expect us to do, especially the end user community, rather than indulge in the UNIX wars that are restricted to the Trade Press. -SP]

University of California’s Computer Science Research Group (the folks who bring us Berkley UNIX) is also voting against the .4
ballot as a block. This stand has nothing to do with the lack of a threads proposal; the vote objects to the working group’s addition of completely new and (their words) “lame” features to UNIX. An amusing twist, this. To a traditional standards activity, one vendor block voting against another; POSIX adds one research group (CSRG) voting against another (.4).

[I believe that this was just an endorsement of the Common Reference Ballot mentioned above, which was submitted by someone at Berkeley. –SP]

The threads group itself is divided over whether they are doing an interface to OS-kernel services or an applications library. They are also divided about whether they are doing an interface to language-independent, concurrent programming services, or just a C-language extension.

In general, .4A seems to be a small core of activists pushing ahead with a clear agenda, with an opposition that complains but appears incapable of putting together a detailed unified counter-proposal. Both the rush to go to ballot, and the move to tie success of the rest of 1003.4 to threads, should be causes for scrutiny.

[I can’t think where you get this idea from. There is no desire that I know of to tie threads to the rest of .4. The people involved are highly motivated and think that the time is right to standardize on a thread interface before the industry become too divergent. It is felt be many people that there is enough experience in the industry and academia to write a good usable standard and are trying to do so. –SP]

Interestingly, if threads are forced back into the base .4 standard, it may end up causing another problem. The ACM’s ARTEWG (the special interest group on Ada’s runtime environment working group) is likely to vote in a block against 1003.4 if it contains a threads proposal that does not support Ada in a natural way.

[This is not likely to happen as I said above. The threads group are talking to the Ada people (constantly it feels like :-) and it is hoped that when the draft is ready for balloting most of the Ada folks will be happy. There is a problem with scope which has never really been properly defined with respect to Ada, especially Ada runtime.

Your overall tone was one of suspicion that there is a subversive plot going on and that half of POSIX is being taken over by a small number of people in the threads group. This is clearly ridiculous as it could never happen; the consensus process prohibits it. –SP]

The Ada folks are concerned that there be an underlying, OS-level model of concurrency consistent with both the C-threads and Ada tasking models. This seems especially important to them if Ada applications want to use standard services written using C libraries which are implemented using C-threads (e.g., windowing and database access). Such a model would also be important for support of Ada compilation systems, which are typically produced by independent software houses to operate on a variety of operating systems and machine architectures.

Dot 4b is a language-independence effort. What’s interesting here is that real-time was one of the groups that the SEC grandfathered out of the requirement that POSIX standards be language-independent. (Other exemptions included other standards well along, like .1, and standards that were intrinsically language-dependent, like .9, FORTRAN bindings). Despite that exemption, real-time may be the first group to write a language-independent binding.

Real-time also has PARs for .4C, a place to put stuff that didn’t make it into .4 (i.e., .4 is to .4C as .1 is to .1B), and .14, the real-time profile.

Language-independence Study Group

I want to straighten out something I was confused about in the last summary report. (Thanks to Jeff Kimmel, of the language-independence study group, for taking the time
to explain this.) Language-independence is a sop to ISO. Two prices we pay to gain rapid international approval of the POSIX standards are an agreement to hand ISO standards formatted in their preferred style, to which end the IEEE is providing editorial assistance, and a commitment to a direction ISO intends to take for all its standards: language independence.

And, to clear up another misconception, Steve McDowell worried, in his last .7 snitch report, that ISO requires language-independent specification languages to themselves be standardized. This would force POSIX to use something frightening like VDL. Fortunately, that turns out only to be true for formal specification languages: languages from which one can derive correctness proofs. ISO isn’t interested in proofs, only in divorcing specifications from specific programming languages. They don’t want to give an unfair advantage to languages in which the things being standardized are likely to be initially implemented, like C or FORTRAN, over more international languages, like ALGOL-66. In other words, POSIX will probably produce specs in ASN.1 or even English. (That’s “language independent.” Get it?)

1003.5: Ada Bindings

Dot five didn’t officially meet in New Orleans, partly to give .5 members more time to attend other groups. Dot five members kept saying things to puzzled members of other committees like, “We’re not really meeting,” “I’m not really here,” and “Well, I am here, but don’t tell our chair, Steve Deller.” One member graciously volunteers this short, but timely, update:

“The Ada binding group (P1003.5) just finished an intensive working meeting at Florida State, in Tallahassee. The meeting went very smoothly. We resolved all the issues brought up by the recent mock ballot, and expect to have a revised draft ready for the April POSIX meeting. That draft is supposed to be given some finishing touches at the meeting, and then sent out for formal ballot.”

1003.8: Transparent File Access

As expected, what used to be dot 8 has split into several groups. There was a meeting on the last day, in which chairs of each of the newly-formed POSIX networking-related groups gave status reports. At that meeting, one attendee objected that the models and APIs that come out of these groups increase portability, but do little or nothing to ensure interoperability. Surely, networking standards should have interoperability as a primary goal, he complained. While the current groups don’t have solving this problem as part of their charter, many attendees agreed that the complaint is valid, and something should be done on this front. Keep your eye on this problem.

While the other subgroups have new numbers, the group standardizing transparent file access (TFA) retains the dot 8 name.

Six months ago, TFA was torn between a faction wanting to canonize NFS, and another insisting on something that supports full dot 1 semantics. Now, the group has achieved consensus. They’ll provide several standards: a core subset with which FTAM will comply, a set of extensions to the core with which various versions of NFS will comply to various degrees, and a full standard that will support full dot 1 semantics. This compromise recognizes the \textit{de facto} international standard without sacrificing a commitment to dot 1.

1003.9: FORTRAN Bindings

Dot 9 is in the middle of editorial cleanup in preparation for balloting. Emphasis until now has been on content, so the draft developed with many styles and formats. Much of the last meeting was spent trying to even things up.

Since things are drawing to a close, you might expect meetings to be sedate. If you read the .9 postings in \texttt{comp.std.unix}, you’ll know that’s not true. When I walked in on the .9 meeting the group was in the middle of a heated discussion. Someone had proposed adding several functions to increase portability of FORTRAN programs. One specific example
was a function that would return the maximum REAL for the implementation. While there is little question of the utility of such a function, there were two sorts of illuminating objections:

1. Some members of the group objected that the standard was not intended to increase portability of FORTRAN programs, only to provide FORTRAN bindings to the .1 standard. (Indeed, unlike .5, .9 makes no attempt to be a stand-alone document. It freely uses pointers into .1.) Others countered that the section being discussed corresponds to section 8, *Language-Specific Service for the C Programming Language*, of the Ugly Green Book; that the group's goal is improving application portability; and that additions that further that goal are completely within the group's charter.

2. One member objected strenuously that many of these additions required REAL support. I was utterly mystified by this objection, until the group patiently explained that, though .9 is an F77 binding, it won't require F77 compliance, and won't use all the features of F77. For example, these new functions were .9's first use of REALs. What the member was objecting to was that without the added functions, a vendor could advertise .9 compliance with an integer-only FORTRAN compiler. Adding these new functions would require that the vendor's FORTRAN compiler actually handle REALs. Think about that.

The ultimate (and, in my opinion, correct) decision was to add the functions, but you can see that there are interesting philosophical divisions in this group. Similar divisions actually exist in all the groups, but the discussions in .9 seem to be more direct and get resolved more quickly. Chalk it up to more programmers, fewer politicians.

**1003.10: Study Group on Supercomputing**

Dot ten has two subgroups, Profile and Batch, each working on a document.

The Supercomputing Application Environment Profile specifies a set of standards, along with options and parameters needed for supercomputing application environments. The current draft, 1.0, is still rough, but specifies most of the required standards. At the April meeting, the Profile subgroup will hold a joint session with dot 0 and the other profile working groups (.11, .14, and the multiprocessing study group) to discuss profiles.

*Batch Extensions for Portable Operating Systems* describes a standard batch management system based on NQS (the Network Queuing System, available from NASA Ames). The batch subgroup began its work within /usr/group's supercomputing working group, has been meeting eight times a year, and is now on draft 1.2. When complete, the document will specify required extensions to POSIX, including interfaces for checkpoint/restart and resource control, utilities for job submission/management and batch system administration, and a network application-level protocol. The subgroup has submitted a PAR for the batch work, which the SEC will consider at their April meeting.

**1003.11: Transaction Processing Study Group**

Good news in transaction processing. Dot 11 has been trying to work out what model of transaction processing to adopt. Because many committee members are also active in other committees specifying other TP models, the committee had a running start, but progress has been slowed somewhat because there are at least three camps: those who favor the ISO model, those who favor the X/Open model, and those who believe that discussion of concrete models is premature.

Part way through the New Orleans meeting the committee took a break from modeling to explore what an API to a transaction processing system might look like. This, finally, provided a fairly uncontentious topic on which all members could collaborate, and the committee seems to have been able to generate real agreement rather quickly. Success breeds success, and this may smooth the way to find other areas that the committee can make progress.

One warning: working out a sample API may serve only to clarify the committee's
thinking about the requirements of their application profile, but I wouldn’t be shocked to see the committee eventually submit a PAR for the work. If that happens, ask yourself whether the committee should be designing APIs for an area where there isn’t yet industry consensus.

1003.12: Protocol Independent Application Interfaces

Dot 12, process to process communication, is one of the groups derived from the division of the old dot 8 group. The big news from this group is that they’ve made a real decision in the struggle between XTI and sockets. The group has decided to invent a new interface, which they hope will combine the best of both and avoid the mistakes of each. This is important. It is the first time since the beginning of the committee (several years ago, counting its origins in /usr/group) that it has actually taken a stand on the question. The issue has come up often in past meetings, but until now been deferred by the group.

On other fronts, the group is still trying to produce two APIs: a detailed network interface and a simple network interface. I worry a bit about having two disjoint interface standards in the same area. Are two standards better than none? (On the other hand, having two raises the possibility of splitting the group into two separate, numbered groups at some later date, a popular POSIX pastime.) Recognizing the danger in this split approach, some members of the group are considering whether it might be possible to specify a single expandable interface.

12xx: Protocol Dependent Interfaces for OSI

This new dot 8 spin-off, chaired by Kester Fong, is looking at protocol-dependent networking interfaces. They’ll begin by concentrating on FTAM. I predict this group will make rapid progress, because its composition is dominated by users.

To help prevent its work from being an Aristotelian exercise in abstract design, the group has begun to collect all the examples it can find of applications based on FTAM. If you have, or know of, any such examples, please pass them on. Kester’s e-mail address is FONG%AESv01.GM@HAC2ARPA.HAC.COM.

1201: User Interface

1201 is growing to four groups: .1 (Applications Programming Interface), .2 (Graphical User Interface), .3 (Human-Computer Interaction), and .4 (XLib). This serves as a focus for an interesting philosophical issue.

As many readers realize, there is widespread sentiment outside of these groups that 1201 should, instead, shrink to zero groups – that standards in this area are premature. Even more interesting is that the same sentiment is widespread inside the groups. The level of dissatisfaction does vary from group to group. Out of curiosity, I requested a vote for dissolution at the first New Orleans meeting of 1201.3. Fewer than one-third of the attendees voted to dissolve. This contrasts with a similar vote in Brussels in 1201.2, where nearly half of the attendees voted to dissolve. With this much anti-1201 sentiment, isn’t there a way to get the IEEE to reconsider the activity? Apparently not.

At the last USENIX, in Washington D.C., Jim Isaak, the SEC chair, explained to the well-attended standards BOF that there is really no easy way to dissolve a committee. If volunteers show up to staff the working group, follow the IEEE rules, and eventually circulate a ballot that passes, they’ve created an IEEE standard. This means, if you don’t like the idea, you currently have only three options.

1. Join the balloting group and vote any proposal down. Not easy; you have to have a good reason for voting no. Of course, “This standard is premature; the direction of industry is too unclear” may be good enough.

2. Join the working group and filibuster until the direction the standard should take does become clear. (Of course, that would be expensive, and lose you popularity points.)

3. Let the group declare a standard and hope everyone ignores it. This one’s dangerous because NIST won’t, which means the vendors
can't, which means users probably won't be permitted to, and will, at least, have to carry the code around as excess baggage.

So, I'm curious. If you don't like what's going on here, which do you intend to do? (Okay, I'm not that picky. If you like what 1201's doing but object to some other portion of what Doug Gwyn calls "the standards juggernaut," what are you doing about it?)

**X3J11: C Language Standard**

Closing on an upbeat note, we have a C standard. What more newsworthy item could you ask for?

From: John S. Quarterman, USENIX Standards Liaison, <jsq@usenix.org>.

The summary report from Jeff Haemer, the USENIX Standards Watchdog Committee Report Editor, is in general just the kind of thing we try to publish. However, there were a few problems with it. In particular, the comments about a supposed block vote against 1003.4 originated by a threads subgroup were inaccurate. There was in fact a common reference ballot that originated with UCB CSRG. It addressed many points throughout the 1003.4 draft document. It was referenced in numerous negative ballots, including several from Institutional Representatives. (USENIX did not reference it in a ballot, but only due to time pressure: USENIX supports it in principal.)

These errors in Jeff's report were due to inadequate review before publication, which occurred because I was out of the country as he finished the report. It was important to get the summary posted on the networks before the Utah standards committee meeting, and turnaround time to substitute reviewers turned out to be greater than anticipated. My apologies for this coordination problem. We will attempt to prevent this kind of situation in the future by more thorough review, including having each section about a specific committee reviewed by the corresponding Watchdog Committee volunteer in addition to being reviewed by me.
CSUUG Established

Peter Pronay

On March 6, the UNIX users in Czechoslovakia met in Brno and established the Czechoslovak UNIX Systems Users Group – CSUUG. They approved the by-laws of the national group and elected the board. The chairman of the board is Mr. Zdenek Jirkovec. He is a systems programmer at CSAD Prague (CSAD is the principal bus & truck transport company in Czechoslovakia). The board has 4 more members.

One of the decisions of the first meeting was that the national backbone of EUnet will be in the Institute of Applied Cybernetics in Bratislava (sitename “iaccs” on EUnet). Gejza Buechler and Peter Pronay will be running the backbone. Mr. Vladimir Verosta from KS Brno will be editing the CSUUG non-periodical. Mr. Jiri Bek from the Economical University in Prague will be looking after the finances.

More than 35 institutions expressed interest in membership in the CSUUG. However, the administrative aspect of creating organizations in Czechoslovakia is complicated recently. Hundreds of different groups, political parties, and movements are waiting to be registered and the bureaucratic “machine” cannot handle the load quickly enough. So, the “official birth” of CSUUG had not yet taken place when I wrote this, but it very likely might now. (We just learned that EUnet and EUUG have both accepted CSUUG’s membership application. - Ed.) Since hard currency is a problem in the country, CSUUG will not know its real number of members before the membership dues are paid. And that cannot happen before the organization is registered by the government.

As you see, life can be hard even when the revolution is soft. Nevertheless, from the point of view of the UNIX community, the CSUUG is here and well off.

You can contact the CSUUG chairman at:
Zdenek Jirkovec +422 228642, ext. 434
CSAD KNV Praha
Krizikova 4-6
186 50 PRAHA Czechoslovakia

May/June 1990
CONGRESS & WORKSHOP PROCEEDINGS

Please enter my one-year subscription to the 1990 USENIX Conference & Workshop Proceedings, which include:

Washington, DC Conference - January 1990
C++ Conference - April 1990
Anaheim Conference - June 1990
UNIX Security Workshop - August 1990
Mach Workshop - October 1990
Large Installation Systems Administration IV Conference - October 1990

COST: $110.00


PAYMENT OPTIONS
☐ Check enclosed payable to USENIX Association. ☐ Purchase order enclosed.
☐ Please charge my: ☐ Visa ☐ MasterCard
Account # ____________________ Exp. Date ____________________
Signature ____________________

Overseas? Please make your payment in U.S. currency by one of the following:
* Charge (Visa, MasterCard, or foreign equivalent)
* International postal money order
* Check - issued by a local branch of a U.S. Bank

Name ____________________________
Address __________________________
City __________________ State/Country ______ Zip/Postal Code ______

Please mail this order form to: USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

This offer is good through December 31, 1990
**ORDER FORM**

**CONFERENCE & WORKSHOP PROCEEDINGS**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Proceedings</th>
<th>Price</th>
<th>Total</th>
<th>Foreign Postage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anaheim Conference</td>
<td>June '90</td>
<td>$22</td>
<td>$15</td>
<td>$37</td>
</tr>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Apr. '90</td>
<td>28</td>
<td>$18</td>
<td>$46</td>
</tr>
<tr>
<td></td>
<td>Washington DC Conference</td>
<td>Jan. '90</td>
<td>25</td>
<td>$15</td>
<td>$40</td>
</tr>
<tr>
<td></td>
<td>Graphics Workshop V</td>
<td>Nov. '89</td>
<td>18</td>
<td>$10</td>
<td>$28</td>
</tr>
<tr>
<td></td>
<td>Distributed &amp; Multiprocessor Sys. Workshop</td>
<td>Oct. '89</td>
<td>30</td>
<td>$20</td>
<td>$50</td>
</tr>
<tr>
<td></td>
<td>Large Installation Sys. Admin. III Workshop</td>
<td>Sept. '89</td>
<td>13</td>
<td>$9</td>
<td>$22</td>
</tr>
<tr>
<td></td>
<td>Baltimore Conference</td>
<td>June '89</td>
<td>20</td>
<td>$15</td>
<td>$35</td>
</tr>
<tr>
<td></td>
<td>UNIX Transaction Processing Workshop</td>
<td>May '89</td>
<td>12</td>
<td>$8</td>
<td>$20</td>
</tr>
<tr>
<td></td>
<td>Software Management Workshop</td>
<td>Apr. '89</td>
<td>20</td>
<td>$15</td>
<td>$35</td>
</tr>
<tr>
<td></td>
<td>San Diego Conference</td>
<td>Feb. '89</td>
<td>30</td>
<td>$20</td>
<td>$50</td>
</tr>
<tr>
<td></td>
<td>C++ Conference</td>
<td>Oct. '88</td>
<td>30</td>
<td>$20</td>
<td>$50</td>
</tr>
<tr>
<td></td>
<td>C++ Workshop</td>
<td>Nov. '87</td>
<td>30</td>
<td>$20</td>
<td>$50</td>
</tr>
<tr>
<td></td>
<td>Graphics Workshop IV</td>
<td>Oct. '87</td>
<td>10</td>
<td>$15</td>
<td>$25</td>
</tr>
<tr>
<td></td>
<td>Washington DC Conference</td>
<td>Jan. '87</td>
<td>10</td>
<td>$20</td>
<td>$30</td>
</tr>
<tr>
<td></td>
<td>Graphics Workshop III</td>
<td>Dec. '86</td>
<td>10</td>
<td>$15</td>
<td>$25</td>
</tr>
</tbody>
</table>

Total price of Proceedings: __________
Calif. residents only add applicable sales tax
Total foreign postage: __________
Total enclosed: __________

* Discounts are available for bulk orders. Please inquire.

**PAYMENT OPTIONS**

☐ Check enclosed payable to USENIX Association. ☐ Purchase order enclosed.
☐ Please charge my: ☐ Visa ☐ MasterCard ☐

Account # ___________________________ Exp.Date ___________________________

Signature ___________________________

Overseas? Please make your payment in U.S. currency by one of the following:
* Charge (Visa, MasterCard, or foreign equivalent)
* International postal money order
* Check - issued by a local branch of a U.S. Bank

**Shipping Information**

5/90

Orders to U.S. and Canada are shipped via printed matter. Please allow 2 weeks for delivery. Foreign orders are shipped via air printed matter. Please allow 10-14 days for delivery. Shipment by UPS, first class, or airmail is available upon request.

Ship to: ____________________________

_______________________________

Please mail this order form to: USENIX Association

Suite 215

2560 Ninth Street

Berkeley, CA 94710

May/June 1990

57
Local User Groups

The Association will support local user groups by doing a 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. Send additions and corrections to login@usenix.org.

CA - Fresno: the Central California UNIX Users Group consists of a uuerc-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 294-4373
brent@CSUFresno.edu or csufres!brent
Commercial institutions or individuals:
Gordon Crumal (209) 875-8755
csfres!gordon (209) 298-8393

CO - Boulder: the Front Range UNIX Users Group meets monthly at different sites.
Steve Gaede gaede@tda.com
Software Design & Analysis (303) 499-4782
P.O. Box 3521
Boulder, CO 80303

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

FL - Fort Lauderdale/Miami: The South Florida UNIX Users Group meets the 2nd Tuesday of each month.
Tony Vincent, John McLaughlin (305) 776-7770
(sun,novavax,gould!sunvax!tony
jmlaughlin@sun.COM
John O'Brien (305) 475-7633
gatech!uflorida!novavax!john

FL - Jacksonville/Northeast: UNIX Users of Jacksonville (uujax) meet the 2nd Thursday of each month.
Tom Blakely (904) 646-2820
uforida!unf7!tfb
Emilie Olsen (904) 390-3621

FL - Melbourne: the Space Coast UNIX Users Group meets at 8pm on the 3rd Wednesday of each month at the Florida Institute of Technology.
Bill Davis (407) 242-4449
bill@ccd.harris.com

FL - Orlando: the Central Florida UNIX Users Group meets the 3rd Thursday of each month.
Mike Geldner (407) 862-0949
codas!sunfla!mike
Ben Goldfarb (407) 275-2790
goldfarb@hc9.uct.edu
Mikel Manitius (codas,atmail)!mikel

FL - Tampa Bay: the Tampa UNIX Users Group meets the 1st Thursday of each month in Largo.
Bill Hargen (813) 530-8655
uunet!pdn!hargen
George W. Leach (813) 530-2376
uunet!pdn!reggie

GA - Atlanta: meets on the 1st 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

MI - Detroit/Ann Arbor: The SouthEastern Michigan Sun Local Users Group meets jointly with the Nameless UNIX Group on the 2nd Thursday of each month in Ann Arbor.
Steve Simmons (313) 426-8981
home: (313) 426-8981
office: (313) 769-4086
K. Richard McGill Bill Bulley
rich@sendai.ann-arbor.mi.us web@applga.uucp

MI - Detroit/Ann Arbor: dinner meetings the 1st Wednesday of each month.
Linda Mason (313) 855-4220
michigan@usr/group
P.O. Box 189602
Farmington Hills, MI 48018-9602

MN - Minneapolis/St. Paul: meets the 1st Wednesday of each month.
UNIX Users of Minnesota Robert A. Monio
17130 Jordan Court pnussutt@nis.mn.org
Lakeville, MN 55044 (612) 895-7007

May/June 1990
MO - St. Louis: St. Louis UNIX Users Group
Eric Kiebler
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105
(314) 725-9492

NE - Omaha: meets the 2nd Thursday of each month.
Kenton Landfield
P.O. Box 44112
Omaha, NE 68144
(402) 291-8300

New England - Northern: meets monthly at different sites.
Peter Schmitt
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755

Pat Parseghian
Dept. of Computer Science
Princeton University
Princeton, NJ 08544

NY - New York City:
Unigroup of New York
G.P.O. Box 1931
New York, NY 10116
Ed Taylor
(212) 513-7777

OH - Columbus: The Columbus Local UNIX Group (CLUG) meets the 1st Monday of each month.
Mark Verber
Physics Department
Ohio State University
Columbus, OH 43210
(614) 292-8002

OK - Tulsa: the Tulsa UNIX Users Group, $USR, meets the 2nd Wednesday of each month.
Stan Mason
(918) 560-5329
tulsix@drd.com
Mark Lawrence
(918) 743-3013
mark@drd.com

PA - Philadelphia: the UNIX SIG of the Philadelphia Area Computer Society (PACS) meets the morning of the 3rd Saturday of each month.
G. Baun, UNIX SIG
rutgers!{bpa,cbmvax}!
c/o PACS
{attunix,philabs}@pencom!taylor
wax
c/o PACS
tikal!cameo!bill

TX - Austin: CACTUS meets the 3rd Thursday of each month.
Capital Area Central Texas UNIX Society
P.O. Box 9786
Austin, TX 78766-9786
officers@peyote.cactus.org
James Johnson
(512) 331-3781

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

TX - Houston: the Houston UNIX Users Group (Hounix) meets the 3rd Tuesday of each month.
Hounix answering machine
(713) 270-8124
Chuck Bentley, vice-president
(713) 789-8928
chuckb@hounix.uucp

TX - San Antonio: the San Antonio UNIX Users (SATUU) meets the 3rd Thursday of each month.
Jeff Mason
gatech!petro!hsatb!jeff
14100 San Pedro
San Antonio, TX 78232

WA - Seattle: meets monthly.
Bill Campbell
(206) 232-4164
Seattle UNIX Group Membership Information
P.O. Box 820
Mercer Island, WA 98040-0820
uw-beaver!tikal!cameo!bill

Washington, D.C.: meets the 1st Tuesday of each month.
Washington Area UNIX Users Group
2070 Chain Bridge Road, Suite 350
Vienna, VA. 22182
Alan Fedder
(703) 448-1908

May/June 1990
Election Results
International Standardization
Calls for Papers
Summary of Board Meeting