KS

# NEWSLETTER

# EUUG

European UNIX† systems User Group

Volume 6, No. 1
SPRING 1986

## CONTENTS

# EUUG

**European UNIX† Systems User Group**

## Newsletter Vol 6 No 1

## Spring 1986



† UNIX is a Trademark of Bell Laboratories.

Copyright (c) 1985. This document may contain information covered by one or more licences, copyrights and non-disclosure agreements. Copying without fee is permitted provided that copies are not made or distributed for commercial advantage and credit to the source is given; abstracting with credit is permitted. All other circulation or reproduction is prohibited without the prior permission of the EUUG.

# EUUG

## European UNIX† systems User Group

# SPRING CONFERENCE & EXHIBITION

Florence, Italy, April 21-24 1986

The EUUG Spring Conference and Exhibition will be held in Florence, April 21st – 24th, at the Conference and Exhibition Centre. The Conference will include a series of state-of-the-art presentations, papers and seminars supported by an Exhibition of UNIX-related products and services, and is open to both members and non-members of the EUUG.

There will be a 3-day **Technical Programme,** an **Industrial Programme,** and a series of **Advanced Tutorials.**

### The Technical Programme

This will run from April 22nd to April 24th, and topics under discussion will include the design and implementation of distributed file systems; management of a large software distribution; database management systems; ISO/OSI implementations for UNIX; and process management under Berkeley 4.2. Speakers for the Technical Programme will include Peter Weinberger from Bell Laboratories, one of the originators of AWK, who will speak on the implementation of the Eighth Edition Remote File System.

### The Industrial Programme

This runs in parallel with the Technical Programme and will concentrate on trends in UNIX System V, looking at Standards, its use in major projects, and the problems incurred in different environments. Applications examined will include UNIX in the European Institutions.

### Advanced Tutorials

Following the success of previous Tutorials, there will be a day of 'advanced' UNIX tutorials, on April 21st. Topics discussed will include: Porting UNIX to New Machines; System V Inter Process Communications; the Source Code Control System, SCCS; and Advanced Features of the Bourne Shell.

### Exhibition

From April 22nd – 24th the conference will also host an Exhibition of UNIX-related hardware, software and support services at which the majority of UNIX vendors in Europe and the US will be represented. A number of technical demonstrations are planned for the Exhibition.

### IEEE Standard for UNIX-like systems

Preceding the conference, from the 16th to the 18th of April there will be a meeting of the IEEE-P1003 Working Group. This Group is developing a "proposed IEEE standard for operating systems that are functionally compatible with the UNIX operating system", (formerly the /usr/group STANDARD). This first meeting of the working group outside the United States, is being held to promote contact with those Europeans involved in the development and standardisation of UNIX.

### For full details, contact:

Helen Gibbons
European Unix Systems User Group
Owles Hall,
Buntingford,
Herts. SG9 9PL,
ENGLAND.
Tel: + 44 763 73039

# Call Me Madam (Editor)

*Jean Wood*

This edition of the EUUG Newsletter represents my first efforts as it's editor. The European UNIX Systems Users Group is growing, making the transition from small friendly user group to large professional (but still friendly) technical organisation; its newsletter must also grow and accomodate those changes.

The newsletter is the medium through which the Governing and Executive Boards of EUUG can reach all EUUG members; as their activities expand, they must keep the membership informed. In this issue, Teus Hagen begins this effort with his report on some of the work of the EUUG.

Similarly, many of the national groups are moving into new areas and the newsletter provides the venue for these groups to share information about common interests, such as conference planning and network administration. Frances Brazier discusses the new approach NLUUG has taken in conference planning. I hope other groups will likewise share their experiences.

Conference reports have traditionally filled these pages, and as you will see, this issue is no different; reports covering several aspects of the Copenhagen meeting are included. In the past, copies of the proceedings have sometimes found their way into this publication. With the exception of selected papers, this is a 'tradition' I believe has ended. Proceedings will be available for those who do not attend the meetings, however, their publication and sale will, for the most part, be distinct and different from this volume.

The EUUGN cannot grow and continue to represent the professional technical nature of EUUG without the continued contributions of it's members. It provides the most visible means for communication amoung all members of the European Unix systems Users Group. Serious technical papers, opinionated essays (Does EUnet eliminate the need for EUUGN?), original cartoons, suggestions and critisizms should be sent electronically (if possible) to mcvax!euugn or mailed to me,

Jean Wood
Editor, EUUGN
Digital Equipment
Centre Technique (European)
B.P. 29 Sophia-Antipolis
06562 Valbonne FRANCE

Contributions for this edition were mailed to 3 different locations; others were sent electronically to 4 others. It is but one of the reasons for the tardiness of this edition, but the one I hope to eliminate first ... before the next time.

Please accept my apologies for the delay in getting this out, the reasons range from inablity to cope with blizzard snows and gale force winds on the French Riviera to inability to cope, at times, with UUCP.

I have come to think of this as my White Rabbit edition, because like like the rabbit in Alice's Wonderland, these days I've often been heard to mutter to myself 'I'm late, I'm late, I'm late, I'm late, I'm late ...'

Most sincerely,

*jean*

# The UNIX Trademark has Changed

*Teus Hagen*

EUUG Executive Board

Since the day that UNIX Europe Ltd (UEL) settled into England and EUUG committee members began meeting regularly with the men in the AT&T/Olivetti caps, I tried furiously to get every occurance of the word 'UNIX' in printed documents shown with an asterisk. Of course, you all know that the asterisk indicates that you should look at the bottom of the sheet of paper and see that UNIX is something or other from Bell Laboratories. Times are changing.

At our last meeting UEL told us that the trademark notice should be changed to "UNIX is a trademark of AT&T in the U.S.A. and other countries". They also made it clear that they are now clearly separated from Olivetti and fully represent AT&T. I still do not know if there is a correlation between these two remarks. But things have changed at UEL and they will look more closely now at what is happening in Europe.

As promised, UEL now seems to be in much better shape with respect to the time delays for licensing. The time delays for 'Educational licensees' are said to be 'improved'. Additionally, UEL can now be reached electronically via EUnet. Yes, UEL is a member of the EUUG!

Perhaps all that is needed is for DEC to join the X/OPEN group. With X/OPEN we are approaching a strong industry voice with respect to UNIX. The first I.E.E.E. UNIX system call standard proposals date from January 1986, and we might expect the proposals for the UNIX tools soon. AT&T have released the next edition of the System V Interface Description (SVID), volumes 1 and 2.

The EUUG is actively participating in the efforts to internationalize UNIX. EUUG members Mike Banahan and Keld Jørn Simonsen are following this on behalf of the EUUG and are actively coordinating with others to make the views of the EUUG known to those preparing standards or implementations in this area.

Organising a meeting for a hundred attendees was easy. But we are now organising very large conferences and that means a very large cash turnover and a greater financial risk. The costs per person have increased significantly. In spite of our rapid growth, we are learning quickly and and with the help of many volunteers we can handle the conferences smoothly.

Conferences must now be planned one to two years in advance. In the past the negotiations about AT&T speakers who were to be invited to EUUG conferences were short and took place several months before the conference. The only major discussion was one month later about the costs. AT&T has changed. So has the EUUG. We need to find keynote speakers well in advance who can boost the conference advance publicity.

We can boast of successful conferences with a high level of technical talks. For new-comers, there is the added possibility to learn via the tutorials. For business, the conference represents an opportunity to get in touch with people involved with UNIX products at all levels. Old hackers get their chance to collect new information from each other. Even though, there is relatively little actual selling going on, companies can use this forum to update and broaden the knowlege of a large number of users in their product areas.

Personally, I do not know a better place than an EUUG conference to collect good computer science information, and this opinion is not confined to UNIX topics. For some it may seem like a reunion, but for most people working with UNIX it is one of the best ways to get the information they need and share ideas in an informal setting.

EUnet is now one of the largest networks in Europe. Thanks to the amazing amount of work done by the backbone site managers, it is in good shape. Several megabytes of data are shipped across Europe and the US every week. EUnet also has links to Japan and Israel and work is in progress to hook up to CSnet (and of course ARPA). This will mean that EUUG members can access ARPA sites in the future. Many meetings and discussions were needed to arrange to have this feature for

# We lead others follow

*genuine*

## The only 32 bit parallel processor available today

Balance 8000 is the practical parallel computer that delivers rather than promises the future. Running from 2 to 20 32 bit processors and offering up to 14 MIPs, Balance gives unrivalled price/performance on a single expandable UNIX™ System that will preserve your initial investment in software, spares and training as your system grows. And it's here. Today.

Call a Sequent Distributor - they'll be pleased to tell you more and to arrange a demonstration.

**SEQUENT COMPUTER SYSTEMS, INC.**

Portland, Oregon, U S A

U.K. Distributors

**Compass Peripheral Systems**
Bridge House, Faraday Road,
Newbury, Berks RG13 2DH
(0635) 33933 Telex: 846301 Compas G

**Quartz Ltd.,**
Studio 5, intec Two,
Wade Road, Basingstoke,
Hants RG24 ONL (0256) 463535

™ UNIX is a Trademark of Bell Labs

the EUUG. More is to come. Discussions are also taking place with the representatives of other networks such as DFN in Germany. The link from EUnet to the United States costs us a lot of money. Most of that money is going right into the hands of the PTT's. Currently we are studying alternatives so that those costs may drop a lot and create better and faster turnaround as well. Hopefully in Florence much more can be said about this and also we can discuss the arrangements for the address domaining. Thanks to all those EUUG members who made EUnet work and still keep it rolling.

The books 'UNIX micro systems' and 'UNIX products for Europe' might look small, but with more than 250 UNIX product entries, these books are hard ones to miss. It was a tremendous effort to collect the basic data for the items and verify that it was correct. Entering the data into a machine was another large task. Luckily part of this work is done by volunteers and an other part by the EUUG secretary. All this made it possible to keep the price of the books rather low. Thanks to Springer Verlag and Samson, the books are available (for a much higher price) in the books shops. No UNIX book has had a wider distribution through Europe.

Clearly, writing programs is something we all know how to do. The number of programs distributed today is phenomenal. Expressing ourselves in words for others to read is a different matter. It seems very hard for people to do. Nevertheless it is one of the ways in which one can receive much attention. Finding people to write for the EUUG newsletter is one of the endless problems which faces the EUUG.

A attempt has been made to keep the rumours from the UNIX community separate from the newsletter. Not only rumours but a lot of UNIX related news will soon be handled by an outside publication and subscriptions to that publication will be made very attractive to EUUG members (25%!).

All these services and others, such as the EUUG software distributions, represent an enormous amount of work done by EUUG. Much is done on the national level, but somethings can be better accomplished from within the EUUG structure. Often local efforts can be beneficial to other national groups. It also seems senseless to replicate some actions in every country in Europe. With the EUUG umbrella, organisation actions can be better coordinated and local activities can be picked up and used elsewhere.

Being European costs money. To have a truly European UNIX organisation means more people involved and bigger financial risks. Budget management in the EUUG has changed to better handle these risks. The first move was to try to have all services basically made self-supporting. This has meant some redefinition of the budget categories. We want all budgeting decisions to be clear to everybody and we need to have a clear understanding of what everything costs us. If someone does something for the EUUG, then that person is given a budget to work within and a responsibility to provide an accurate accounting.

Much, much more work is done by EUUG members than I have included here. It is an active group to which its members love to belong where we can all contribute and benefit.

Thanks to you all.

*Teus Hagen.*

# Image Synthesis with UNIX

*Etienne BEEKER*

Institut National de la Communication Audiovisuelle (INA)
4, Avenue de l'Europe
94360 BRY/ MARNE (France)
33-1-875-84-07

*ABSTRACT*

For two years, the "Institut de la Communication Audiovisuelle" (INA), has developed a collection of programs to help producing three dimensional picture sequences. This software is designed to be used by producers and animators with no previous background in computers. This paper exposes the major programs used, i.e. the modeling, functions, the way we build animation and the rendering program.

## 1. Introduction

With the aim to create new special effects for video applications, our team has developed a set of programs for producing what is now generally called "3D animation". Three classes of programs have been written :

- Modeling : this software provides interactive, graphical input of polygons with the help of menu, tablet and mouse, and the transformation of these polygons into bicubic patches, with or without continuity of the derivatives between adjacent surfaces [BEE84].
- Animation : the same tools also provide interactive input of trajectories applied to objects, to the camera, to lights, etc., with a data base as a background result and immediate visualization of animated wire frames (line test). [BHN84]. The databases generated by these two classes of software are entirely compatible with the third class.
- Rendering : three types of programs are available (scan-line, ray-tracing, particle systems). Only the scan-line program will be exposed here. Its main characteristics are :
  - simultaneous rendering of bicubic patches, polygons and backgrounds
  - texture mapping
  - casted shadows
  - antialiassing
  - transparences
  - reflections on flat surfaces, etc...

All this software is written in C under UNIX For the reason of portability and because programs are in constant development and enhancement, no assembly langage is used.

A diagram of these differents programs is shown in Fig. 1, with the type of data transferred between them.

## 2. Modeling of objects

The most natural way to build a shape is to approximate its surface with as many polygons it is necessary for a given precision. This technique, although it has the advantage of simplicity, presents several drawbacks :

— irregularities appear if the point of view is too close to the object,

— the data base can be huge for complex objects,

— special effects like texture mapping are not easy.

For these reasons, we limit this technique to geometrical objects like cubes, prismas and more

generally polyhedrons, or to objects which are to appear rather small on the screen.

More complex surfaces were introduced by Bezier [BEZ72] in the sixties for automobile design and have known since a wide development for all applications in CAD. This kind of patch allows modelling free form surfaces and is of particular interest when used for designing objects where few industrial production constraints are involved. For these reasons, this type of patch was selected for designing 3D objects as a step towards building animated sequences in a realistic way.

The general equation of a bicubic patch is :

$$P = (w^3\ w^2\ w\ 1)(B)(P)(B)(u^3\ u^2\ u\ 1)^t \qquad u,w\ \ member[0,1]$$

where

$$B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

is the Bernstein matrix, and

$$P = \begin{bmatrix} P_{00}P_{01}P_{02}P_{03} \\ P_{10}P_{11}P_{12}P_{13} \\ P_{20}P_{21}P_{22}P_{23} \\ P_{30}P_{31}P_{32}P_{33} \end{bmatrix}$$

the control points matrix ( see fig.2 )

The 4 points $P_{00}$, $P_{03}$, $P_{30}$ and $P_{33}$ called "corners" belong to the patch surfaces.

Other points control either the slope of boundary curves, or the surface of the patch. These points are very meaningful even for a non mathematician and moving one of then give a foreseable result.

The method we developed in order to build objects out of patches is composed of three steps :

1) Design of the general shape of objects from polygons, with a CAD program. For historical and coherency reasons, we wrote our own program (many functions like the menu driver or the wire frame display routines are common to the patch designing program). The data structure chosen for the polygon description is very usual thus allowing any other construction. Polygons are not necessarely flat but (only constraint) they must have 4 edges.

2) Automatic transformation of polygons into Bernstein patches. Arbitrary default control points are chosen with bilinear interpolations.

3) Design of nicer shapes with another CAD program taking these patches as input. The freedom given by the new created control points allows to smooth the surface with the aid of mathematical constraints. This smoothing is done automatically by the program, which has first analyzed the different topological positions of the patches (if one corner belows to 2, 3 or 4 or more patches, if it is an object edge, etc.). The designer, while moving the control points interactively, can continually check the corresponding smoothing like playing with some kind of mathematical model clay.

For example, a simple sphere can be designed with 18 bicubic patches, where 1000 polygons would be insufficient if the point of view were too close.

Other advantages include texture mapping and easy object deformations. Texture mapping can be done by a correspondance between the image mesh (512 x 512 for instance) and the regular mesh of patches isoparametrics (u or w constants).See figure 3 which shows the correspondance between the 2 meshes.

Object deformations are easier than with polygons because of the lower number of significant control points (the "corners") and conservation of topological coherency.

All modelling programs run under Unix on a French super-microcomputer called SM90 (designed by CNET). This machine, based on a 68000 and equiped with a bitmap display, is used as a personal graphic workstation.

## 3. Rendering

The major problems to render mathematically defined objects in a realistic way, are the elimination of hidden surfaces and the implementation of a shading model, i.e. formulas that calculate the interaction of light with the object surfaces. Many solutions have been given to these problems and the one we chose can be found in [LCW80], [BLI77], [COO81] and [PHO75].

The next problem is to give to the user an ergonomic interface to input all parameters necessary to simulate nicely the behavior of various physical phenomena :

— manner in which lights are reflected on surfaces (metallic, plastic, rough, transparent, etc.),

— casted shadows,

— reflections and mirroring effects.

Our answer was to create a specialized language to describe these parameters. This language (using a lexicon and a grammar) insures a large degree of homogeneity in the treatment of many particular cases as well as being open to all subsequent evolutions and extensions [NIC84].

This language will not be extensively described here, but an example is given below :

**chair : 1 -> 60**
**table : 61 -> 140**

means that the chair is made of patches 1 through 60 and the table from patches 61 to 140.

The nature of these objects is given as :

**chair , flat_plastic clear_white**
**table , flat_plastic pure_red**

where clear_white and pure_red have been defined in a file "colors" :

**define (clear_white, color : 255 255 255)**
**define (pure_red, color : 255 0 0)**

and flat_plastic in a file "materials" :

**define (flat_plastic, plastic_specular : 0 ambiance 100)**

These files are included at the begining of the parameters file :

**include (colors)**
**include (materials)**

which is passed through the M4 processor for expansion.

The result is :

```
Surface_nature :

    1 -> 60,
        plastic_specular : 0
        ambiance : 100
        color : 255 255 255

    61 -> 140
        plastic_specular : 0
        ambiance : 100
        color : 255   0   0
```

In a second pass, this text is given to Lex and Yacc which verify its syntax and coherency and send the input data to the rendering program. Default parameters are set when they are not specified and error messages generated for situations without solutions.

Many materials and colors can be predefined which are more complex than those of the example. For instance, the word plastic corresponds to the Phong's model, but a more general model of shading is implemented (Blinn and Cook's model) that can simulate metallic reflections. Transparency, roughness, texture, etc., can also be easily described as well as surfaces that carry or/and generate shadows [HOU 84].

Unfortunately, this input is textual and not graphic. Our designer-user cannot be totally ignorant of computers. He must at least know something about our favorite editors ed, vi or emacs. We are presently thinking about a more ergonomic input interface, using menus and a mouse, but this is not straightforward.

Rendering programs are rather heavy : almost a hundred million floating point operations are necessary to calculate an image, pixel by pixel. Our program (written in C of course) runs on a Perkin-Elmer 3210 without any special enhancements except a floating point accelerator. It can run on any 32-bit machine. We tested it on a Vax 780 and a Gould 3297. On this computer, complex images can be obtained in less than 5 minutes.

## 4. Animation

The difficulty in 3D animation paradoxically arises from the vast number of possibilities. Movement of the point of view, for instance, does not affect the trajectory, speed and acceleration of objects not even those objects that are crossed through. During motion all variations in the view direction or in the focal can be envisaged.

The problem is then to propose to animators interactive tools for animation design that allow them to control step by step the movement of the virtual camera, the movement of objects, their possible deformations, and all conceptual operations which result is sometimes "unconceivable".

The method developped at INA is based both on interactivity and graphical design of trajectories. To animate the point of view, the animator sets key-points of the movement with the help of a digitizing tablet, in the scene space represented on the screen. Focal, direction of view are determined in the same way.

Object movements are more difficult : they must be bound to a cartesian reference. Their deformations are done in their own reference. A reference can be in translation and rotation with its father reference, and all references are linked together in a tree structure. The movement of an arm, for instance, can be defined in the space of it's body, and the movement of the body in the space of the scene. Complex movements can thus be defined (the arm regarding the entire scene) by simple decompositions.

To avoid rough motions, interpolation between key-parameters is done with "splines" which ensure a good continuity in movement and speed variations.

A spline is shown on Fig. 4, with its control points. Moving a control point makes the spline follow the same direction. Splines are $C^2$ continuous.

On a physical trajectory, equal spaces of time are set for motion timing. Variations of speed are also interpolated by splines, thus preventing infinite accelerations.

The animator can visualize instantaneously the scene in wire frame at any instant of the sequence and modify, if needed, any previously defined parameter. Unfortunately, the animation cannot be reproduced in real-time, at the speed of 25 images per second, but some minutes are

sufficient to obtain a line-test in video, with the help of an image per image recording system.

This system of animation design is written in continuity with the rendering program and bound to visualization programs, thus allowing to calculate automatically all images composing the sequence.

## 5. Conclusion

A method for building 3D animated sequences has been presented. Several aspects have not been discussed such as a ray-tracing rendering method, a fractal object generator, and a particle system program which are now under development, and will allow new effects.


## Acknowledgements

All this system could not have been built without the work of a whole team : Jean-Charles Hourcade was mostly involved in the scan-line program, A. Nicolas in the animation program, Daniel Borenstein in worrying about object modelling and Bruno Tezenas du Montcel in ray_tracing and shading calculus. Special thanks also to Denis Freyd who until now acted as production coordinator, and to Michel Gien who helped me to rewrite this paper.


## References


[BHN84] D. Borenstein, J.C. Hourcade, A. Nicolas
"Graphical tools for animation"
(to appear)

[BEZ72] P. Bezier
"Numerical control - Mathematics and applications"
John Wiley and Son, London 1972

[LCW80] J. Lane, L. Carpenter, T. Whitted, J. Blinn
"Scan line method for displaying parametrically defined surfaces"
Communication of the ACM, Vol. 15, No. 3, Jan. 80, pp. 23-24

[BEE84] E. Beeker
"Smoothing of shapes designed with free-form surfaces"
(to appear)

[BLI77] J. Blinn
"Models of light reflection for computer synthesized pictures"
ACM Computer Graphics, Vol. 11, No. 2 1977

[COO81] R. Cook, K. Torrance
"A reflection model for computer graphics"
ACM Computer Graphics, Vol. 15, No. 3, Aug. 1981

[PHO75] B.T. Phong
"Illumination for computer generated pictures"
ACM Computer Graphics, Vol. 18, 1975

[NIC84] A. Nicolas, J.C. Hourcade
"Unix based languages for synthetic images"
(to appear)

[HOU84] J.C. Hourcade, A. Nicolas
"Algorithms for Antialiased Casted Shadows"
(to appear)

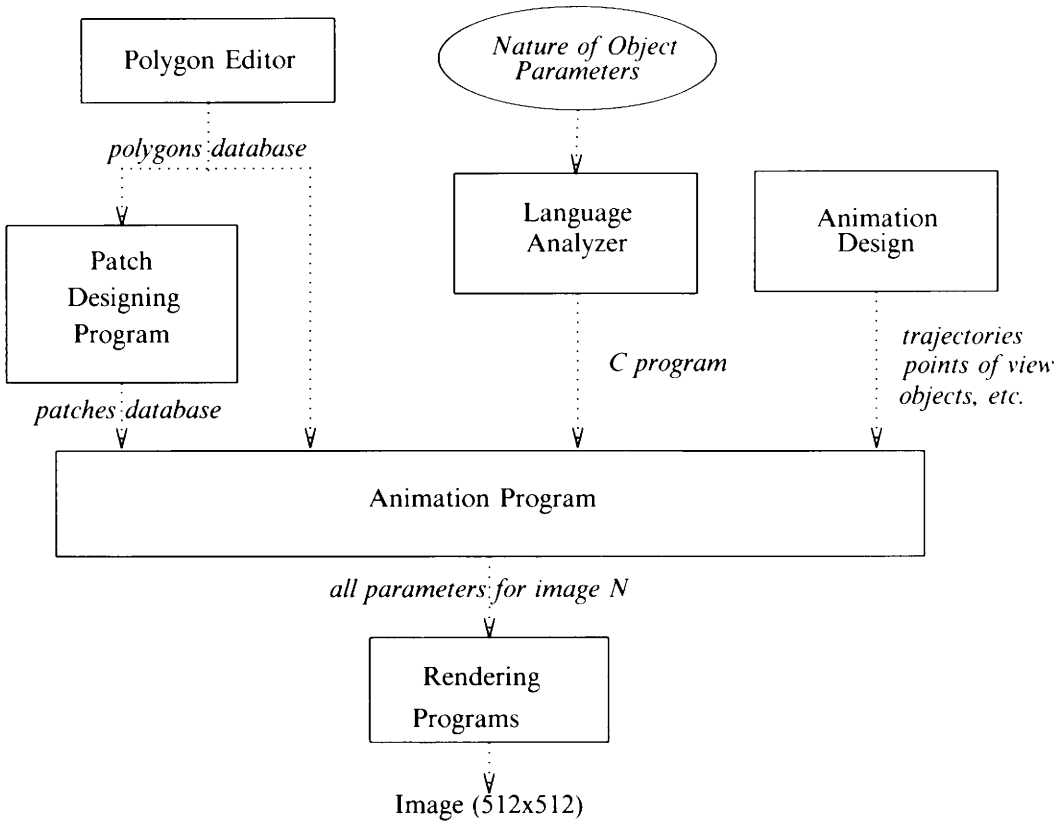


Figure 1: Diagram of main programs used

# Two Unique References — for UNIX™ Users


Price £16

Price £7.50

Hundreds of entries • Valuable technical information • Addresses of manufacturers • Quick reference indexes • Easy-to-follow classified entries.

If you want to get the latest information on UNIX hardware, software and service, then order your copies NOW!

## Contact:

**The Secretary, EUUG, Owles Hall, Buntingford, Herts, England SG9 9PL.**
**Tel: 0763 73039**

# European UNIX System User Group Meeting
# Bella Center, Copenhagen — 10th - 13th September 1985

*Kim F. Storm & Arne Facius,*
*University of Copenhagen, Denmark.*
*edited by Peter Collinson*
*Secretary*

## Introduction

I didn't manage to get to Copenhagen and so two willing volunteers were found to take some notes on what went on. The report below contains the abstracts of the conference talks plus some comments from Kim and Arne. Many thanks to them for the work which they have done.

## Tutorial sessions, Tuesday 10 September

### UNIX System V administration and overview

**Jim Joyce**
**International Technical Seminars, USA**

*Abstract*

This course is designed to introduce essential tools that make the UNIX system administration for System V more efficient and secure. Among the topics are: shell scripts for effective system maintenance, monitoring files that grow automatically, customised system management, correct selection of system permissions, device configuration.

### Fast Prototyping Techniques

**Rich Morin**
**Canta Forda Computer Lab, USA**

*Abstract*

This course learns to combine tools with proven techniques to do fast prototyping under UNIX, saving month in production time. The powerful toolkit of utilities link together in a pipeline or in a shell script, to create applications in a fraction of the time it would take to code the equivalent functionality in C. Among the topics: packaging; from prototype to production model; application design tactics in the UNIX environment; idioms; shortcuts to results; channeling data flow effectively; debugging tips and techniques.

### LEX and YACC

**The Instruction Set Ltd, GB**

*Abstract*

The tools *Lex*, a lexical analyser generator, and *YACC*, Yet Another Compiler Compiler are invariably ignored by all but compiler writers; indeed their names do not cause immediate interest on the part of the ordinary UNIX user.

In this tutorial we examine the tools and their application to everyday activities.

**Writing Device Drivers**
**The Instruction Set Ltd, GB**

*Abstract*

In this tutorial we examine the principles of operation of UNIX device drivers. Their overall structure is studied and their interaction with the rest of the system explained.

It is assumed that attendees are aware of the basic characteristics of hardware and the way it is typically driven by software. It is further assumed that a thorough understanding of the UNIX system at the user level exists.

**Functions in the System V Shell**
**The Instruction Set Ltd, GB**

*Abstract*

The Bourne shell had not altered significantly since its introduction in Version 7; not that is until functions were introduced into the System V.2 shell.

In this seminar we explain the basic principles of operation of shell functions and examine some functions that are in regular use.

The distinction between shell functions and shell scripts will be explained.

## Technical Sessions, Wednesday 11 September

**AWK as a General-Purpose Programming Language**
**Brian W. Kernighan**
**AT&T Bell Laboratories, USA**

*Abstract*

*Awk* is a pattern-action language with convenient facilities for processing strings and numbers. It was originally intended for simple data validation, report generation and data transformation tasks; *awk* programs for such tasks can often be expressed in only a few lines of code.

*Awk* has often been pressed into service in ways far beyond what was originally intended. Recent changes to *awk* make it more suitable as a general-purpose programming language. This paper will discuss new features — new built-in functions, multi-file input, dynamic regular expressions and text substitution, and user-definable functions — and illustrate some interesting applications that they allow.

*Comment*

The original purpose of *awk* was to solve simple data manipulation problems for which people normally would not write a special program. The programs written in *awk* were intended to be small. But larger and larger programs have been written because of the strength of the language. Programs of 1000 lines of code are not unusual. The new version of *awk* has several new features which will ease the task of writing large programs.

Some of the new features are:

- One can close an open file.
- A 'system' function.
- Trigonometric functions like 'sin' and 'cos'.
- The command line arguments can be accessed through *ARGC* and *ARGV* variables.
- A 'getline' function which can read ahead in the input, or read from a file or a pipe.
- Substitution functions 'sub' and 'gsub' (equivalent to s/../../ and s/../../g).
- The field separator (FS) can now be a regular expression.
- New functions may be defined. Parameter passing is call by value for scalars and call by reference for arrays, as in C. To avoid declarations arguments are local while the rest of the names are global. †

*Awk* has been used for several unrelated purposes, e.g.

- Fast prototyping (but with rather slow execution).

† is this really a language for large programs? KS

- Report generation, data transformation, data validation, information retrieval.
- Compiler implementation language.
- Object language for compilers.
- As peoples first programming language, because of easy transition from idea to program.

Some questions resulted in the following answers:

- Only one pipe open at a time, because of limitations in **popen**.
- There is a manual.
- The new awk is NOT available (but ask UEL).
- Better error messages (at last).
- Array elements may be deleted.

**The Streams facility in UNIX System V rel. 3**
**Bob Duncanson**
**Unix Europe Ltd., GB**

*Abstract*

*Streams* is a set of mechanisms within the UNIX System Kernel that allow character I/O to be implemented in a modular way. As a result, the drivers for character-type devices have a structured interface to the UNIX system kernel. The *streams* mechanisms also provide a flexible framework within which different networking architectures can easily be designed and implemented.

*Streams* now includes a conceptual model of character I/O, a well defined interface with the rest of the UNIX system, and a library of utility programs, other utilities (such as an administration driver) and several new (or modified) UNIX system calls.

*Comment*

Historically the optimization of character input and output did not draw much attention, because in those days the TTY-33 was state of the art. Nowadays, we do not only have fast terminals. But terminals may be connected through networks, and one terminal may talk to several machines. Streams are based on message passing, and they are used in System V (R3) to split the kernel into modules, giving a clean interface between drivers and the rest of the kernel.

One can add functionality to a driver by inserting modules between the driver and the kernel. Such modules may for instance implement a specific protocol, for UUCP say, and just by replacing this module a new procotol can be used, which leads to network independent user-level routines. Future versions of UUCP will be independent of the specific protocol used. Streams are incompatible with sockets, but work is going on to implement sockets on top of streams.

**Object Formats and Memory Management techniques in UNIX**
**Martijn de Lange & Hans van Someren**
**Associated Computer Exp**

*Abstract*

The talk will give an overview of memory management techniques encountered in modern (UNIX) computer systems, and of the relation between object formats and memory management. After an extensive introduction on these general topics, the ACE tailored System V kernel memory management will be described.

The ACE UNIX System V kernel for MC680X0 based systems is portable over a wide range of Memory Management units. Internally, the system uses a flexible layered database for hiding the MMU dependent parts and for the segmentation housekeeping details. On top of a small interface to the MMU dependent parts, and together with this flexible database, a powerful memory management system has been implemented. It improved the internal structure of the system, the performance and the user's interface to the segmentation.

Fundamental to the user's interface is the System V COFF format (Common Object File Format). The S-V COFF format is accepted as a standard load format, but COFF extensions to record the segment attributes made the multi-segmentation management system much more powerful.

The management system allows swap on demand and highly efficient process forking using copy on write segments. The user is able to allocate segments to any extent, to allocate segments dynamically, and to use shared run time library segments. With the extended COFF format it is possible to

specify by means of indirect sections that the segment description is to be taken from another COFF file. A process uses a set of segments, which is described by it's private view with as many as 16 attributes for each individual segment.

The advantages are obvious: reduced memory usage, swap on demand, memory sharing, sharing of libraries, and an improved system performance and maintainability.

The extended COFF is implemented for all MMU's supported by the ACE tailored System V kernels for single and multiprocessor Motorola MC680X0 based systems. In the future, a multi-segmentation system with paging might be expected.

*Comment*

The abstract says it all. We can only add that they report disk space savings upto 30% because of the shared libraries.

However Martijn de Lange gave a new interpretation of the evolution of UNIX, of which we can only give a defective summary: UNIX was a kid born in a wealthy family (Bell Labs) with a happy childhood through the ages of 5, 6, and 7. After that the kid went to the university (UCB), got a middle age crisis between 41 and 42, went into therapy and found back to the age of 3 (III), and now it has a split personality, one at 43 and one at 5.

**An improved UNIX Kermit file transfer program**
**Barbro Malm & Johan Helsingius**

*Abstract*

While local area networks and packet switching long-haul networks are getting increasingly popular, a surprising amount of file transfer work is still being done over slow and unreliable asynchronous lines.

In the UNIX environment, this is usually done by the UUCP system, but often only "transient" file transfer capability is needed, and the configuration of UUCP links can get too complicated, especially for casual users.

When attempting to move data between computers using incompatible operating systems, the situation is much more complicated. Things like incompatible character sets, buffering techniques, half and full duplex transmission, and interpretation of special characters all make communication a definitely non-trivial task.

One solution is a public-domain file transfer protocol, called KERMIT (KL10 Error-free Reciprocal Micro Interconnect over TTY lines). Kermit was originally developed at the University of Columbia, but its development work has since spread to sites all over the world.

Implementations currently exist for a remarkably wide range of systems, from Commodore and Apple home computers to CDC and IBM mainframes. University of Columbia, as well as many other sites, maintains and distributes the latest releases of the various Kermit versions.

Kermit file transfers are done using cooperating programs running at both ends of the communication line, passing small, checksummed packets over regular TTY lines. Facilities exist for transmitting 8-bit data over 7-bit lines, as well as terminal emulation capability.

Our recent work has been aimed at extending the current Kermit protocol to incorporate some more advanced transmission protocol techniques, such as 'floating windows', and simultaneous full-duplex bi-way file transfer. A prototype 'advanced' UNIX kermit has been implemented using, among other tools, LEX and YACC.

One important possibility offered by this development is the capability of providing UUCP-like network services to non-UNIX machines, such as PC's. Some of the implications of this will also be covered in this talk.

The UNIX Kermit program (as well as Kermits for other operating systems) is available for free distribution to all potential users, except for explicitly commercial purposes.

*Comment*

This was the first speaker after lunch, and we were all likely to go to sleep; Johan assured us that we would. For the sake of the readers, we kept ourself awake, revealing that the abstract covers the contents of the lecture perfectly. We can only add that the extended Kermit protocol has a 90-95% efficiency compared to the 50-70% efficiency of the standard protocol. He claimed that his participation at the conference was sponsored by the Finnish tourist agency, and that he had to do some

advertising to get EUUG to hold the next conference in Finland. He showed us some very fine slides, which at least convinced me, that Finland would be a good place to hold a conference.

**Networking with ISO and UNIX**
**Sylvain Langlois & Alain Gauteron**
**Project Rose, Bull, F**

*Abstract*

During the last 5 years, operating systems have been evolving towards new concepts and design in order to integrate inter-system communications. The work done by standardization bodies within ISO has been conceptualized into the Basic Reference Model for Open Systems Interconnection. Protocols based on this Model have now reached a level where they are quite well specified (state of International Standards), and thus can be incorporated into operating systems.

This communication presents an application of the OSI Model to the case of a Local Area Network environment. It mainly covers integration of the 4 lower layers of the Model into the UNIX kernel. A brief overview of the gateway to a public network, which is currently in progress, is also given.

The LAN architecture is based on an ETHERNET, using the ISO Internet Protocol (DIS 8473), and the Class 4 Transport Protocol (IS 8073), which has been slightly simplified to operate on top of a connectionless Network Protocol. Nevertheless, it is important to note that the resulting Transport implementation enforces ISO conformance rules; the simplifications can rather be seen as implementation choices for optimization in a local area networking context. The gateway is a Transport level relay to ISO Class 3 Transport sitting on top of an X.25 network.

After a short presentation of this architecture, and especially modifications which have been made to Transport Protocol specifications to adapt it to a connectionless LAN, we describe the implementation in more detail.

The ease of adding communication facilities to the UNIX System strongly depends on which version of UNIX is chosen. IPC and memory management mechanisms developed by Berkeley, and work done for TCP/IP implementation, made the 4.2BSD version more attractive for such a development. We believed that it was possible to use this particular system architecture for what we wanted to do, without too many changes in the existing facilities offered by 4.2BSD. Even if we have been obliged to make some small changes for implementing the Transport Service, we would like to consider this presentation as a kind of example of adding new communication domains in a 4.2BSD kernel.

Finally, we would like to address some problems we are now encountering when trying to incorporate a connection oriented network protocol, such as X.25, in the 4.2BSD kernel.

*Comment*

The ROSE project funded by the EEC ESPRIT. Its overall purpose is to establish electronic mail, file transfer, remote login, and teleconferencing for the other ESPRIT projects. The aim is to implement ISO/OSI under UNIX. The project schedule says that an operational connection between UNIX (4.2BSD) machines should be ready in 1986.

They use the standard socket interface of 4.2, but with some modifications to set/getsockopt, and a new confirm system call. They have a new address family called AF_ISO, which is almost compatible with the standard 4.2 code.

The system they use is a V7 UNIX with 4.2 network enhancements; TCP/IP is still present in the system, but it is not working. They do not have any performance figures yet.

To the question why they did not use System V, since some of the participants in the project were also members of X/OPEN, he answered that it might be the next step, but that it would be done by another group.

**Sendmail now, and its next generation**
**Miriam Amos**
**DEC, Ultrix Engineering Group, Berkeley Division, USA**

*Abstract*

Berkeley's Sendmail has become the comparative yardstick for UNIX mail systems. It provides a reliable mail delivery system. However, its extensive size and flexibility may be considered overkill. This is a discussion of Berkeley's current mail system and the next generation of the mail system to

come.

*Comment*

Miriam would like to hear what Sendmail should do from everybody. She talked about the improvements that are planned for future Sendmail systems.

The design goals for a new Sendmail are: compatibility with existing mail facilities, reliability (no lost mail), expandable (possible to write code for something that might exist), no configuration information should be compiled into the programs, individual forwarding and accessing of mailing lists, and finally optimisation of network traffic.

The current Sendmail has aliasing and forwarding, it does its own queueing, the SMTP protocol is built into the system, it has a flexible configuration, and can do automatic routing to network gateways. However, Sendmail is not 'UNIX like', because it does too much in a single program, trying to gateway between all sorts of formats and systems. The 'rewriting rules' used in the configuration are simple in themselves, but their connection is cryptic. Maintainance is also a headache, it is 50000 lines of code, and changing the configuration demands the presence on an expert. The plans are to break Sendmail into 'one problem — one program' thunks; a 'mail crossbar' which does gateweying between dedicated mailers, a generalized queue management like lpr, separate programs for maintaining the aliasing files, and a separate SMTP server. 4.3 will still have the old Sendmail, but it is faster, better, and perhaps with better documentation.

**Remote File Systems on UNIX**
**Douglas P. Kingston III**
**Centrum voor Wiskunde en Informatica, NL**

*Abstract*

In the past few years, the nature of computing has changed dramatically as the technology has made it possible to provide computers for small groups or individual users, while sharing more expensive resources via networking. Unfortunately this has also created problems since it is still desirable to easily access data belonging to others that may now reside on another system. When this capability is provided across a network in such a way that the remote files retain the basic attributes of a local file, it is referred to a Remote File System or *RFS*.

While not in widespread use, there have been a large number of attempts at developing a RFS capability for the UNIX operating system. The scope of such work has varied greatly between different attempts. This reflects changes in the design criteria, the underlying capabilities expected in the UNIX system, and the extent to which people were willing to alter UNIX itself. The various implementations have varied greatly in their transparency and efficiency, the two most important qualities of a RFS. In addition, the proprietary nature of many of the RFS implementations has greatly hampered their widespread acceptance by the UNIX community.

The ideal RFS is totally transparent to all UNIX user processes, has no noticable effect on operating system performance, and is available to be implemented on a variety of systems.

The paper will survey many of the RFS on UNIX, explain their basic designs, and comment on how well they approximate the ideal RFS, at what cost and with what disadvantages.

Based on the preceding review, and the implementations the CWI has been able to obtain, we have undertaken to implement a RFS for 4.3BSD that will be generally available to the UNIX community. We will detail the design and implementation choices and report on current progress and current or anticipated performance. The CWI RFS is implemented in the kernel for transparency and efficiency. Implementation has begun and a test system was completed two months ago.

*Comment*

There are zillions of RFS for UNIX, but none of the publicly available are really "good"; either they are primitive or their performance is too bad for practical use. The public domain CWI RFS will be based on the idea of having a **/net/hosts** directory where the remote file systems are mounted, which is similar to the systems from Lucas Films and Harvard. The plan is to implement a canonical file system interface in 4.3, and then implement RFS on top of this interface.

*Q.* What is wrong with '/..' as net prefix?

*A.* Nothing, but in this way '/..' is no longer equal to '/'.

*Q.* Shouldn't a really portable system be implemented in user space?

*A.* It is not reasonable not to touch the kernel.

*Q.* But not everyone has a source licence?

*A.* That is true, but even the Newcastle Connection may run in the kernel for efficiency.

**EUUG General Membership Meeting**

Teus Hagen started things off by making a short statement.

There are now national user groups in 11 countries. There are associated groups in Switzerland, Ireland, and Austria. The DECUS UNIX group also listen in on the EUUG activities. The governing board is composed of representatives from the national groups. Individual members have no influence on the Governing board.

The Newsletter is now edited by Jean Wood, who has taken over from Jim McKie.

The EUUG treasurer is now Nigel Martin, who has taken over from Mike Banahan.

EUUG actvities are:-

The EUUG Newsletter, with 4 issues per year.

The EUUG Conferences. They are planned two years ahead. The plans are to put more weight on the technical part of the conferences, and have smaller (or no) exhibitions, and make the industrial sessions larger.

There are talks with AT&T, and frequent contacts to UEL. EUUG participate in standardisation (X/OPEN) and internationalisation. EUUG also interchange information with its sister organisation USENIX.

EUUG helps members in licencing questions.

EUUG is running the international EUNET network. It administrates the backbone structure, software maintainance and distribution, accounting, and establish gateways to other network, e.g. EARN, CSNET, USENET, and ROSE.

EUUG has published two catalogues on UNIX system micros and UNIX products in Europe.

A major part of the EUUG work is the tape distributions, with more than 100 tapes per year. There are tapes with UUCP, GNU EMACS, and USENIX software.

A future plan is to cooperate with Apt Data Services in the production of a weekly newsletter 'UNIGRAM'. Members will obtain a discounted subcription to the newsletter.

EUUG try to stimulate UNIX activities, and UEL offers awards for student papers, e.g. free travel and participation to conferences. The project results must be free for the UNIX community.

The next EUUG conference will be held in Florence, April 21-24, 1986.

The Autumn 1986 conference will be held in London, September 9-12, 1986. There will be no exhibition, but more workshops.

The venue for future conferences has not been decided, and one idea is to alter the conference frequency to one per year.

There were then some questions and answers.

*Q.* Can EUUG make the ISO transport layers, e.g. from ROSE, available?

*A.* Are working on it, but no success yet.

*Q.* Who are members of the EUUG?

*A.* All national members, including students at the universities, are automatically members of the EUUG.

*Q.* Can we enforce deadlines for abstracts and papers to the conferences? it would be nice to have the proceedings before the conference.

*A.* There is always a conflict; to get interesting talks at the conferences, you often have to beg people to give them. The current programme was not ready until 14 days before the conference.

*Q.* The proceedings should be made available to non-participants and non-EUUG members sooner.

*A.* One thought is to put the proceedings in the newsletter.

Here, a vote showed that nobody was interrested in letting a professional publisher print the proceedings, while the majority was in favor of publishing them in the Newsletter.

By a show of hands it was clear that most of the participants only attended the technical sessions.

The impact of an exhibition is that the conference must be held in a place that is large enough to hold the exhibition, and conference centres are quite expensive. EUUG made a lot of money in Nijmegen, lost it all in Paris, and will make a small profit on the Copenhagen conference.

It was suggested that the exhibitors should be allowed to send technical people only, that is — no sales people, but exhibitors like to sell computers.

Some attendees cannot come if there is no exhibition to wave in front of their management's noses as a reason for participation.

The final remark was to ask the exhibitors what they think.

## Technical Sessions, Thursday 12 September

**Overview of 4.3 BSD**
**Kevin J. Dunlap**
**DEC, Ultrix Engineering Group, Berkeley Division, USA**

*Abstract*

4.2BSD provided new functionality, but due to the lack of time was not tuned to the level the developers would have liked. 4.3BSD is the next release of Berkeley's Unix offering. This release includes the system tuning that time restraints prohibited on 4.2BSD as well as additional functionality.

Performance improvements were provided by use of cacheing, optimization of existing algorithms, selection of more efficient search algorithms, and utilizing the more efficient facilities provided by 4.2BSD. Some of the new functionality added were expansion of the network capabilities to handle subnets and gateways, support for windows and system logging. As well as extending the libraries and utilities to handle the new Internet name server, new system management tools, and Pascal support for dbx.

*Comment*

An overview of the optimizations performed in 4.3BSD was presented. Of specific points to mention is:

Kernel optimization:

- Process management & scheduling: break list into 3 parts: active,zombie,unused.
- Filesystem:
  find big block instead of many small 4/8 K blocks.

Improvements to libraries & utilities:

- hashed databases
- buffered i/o
- mail system
- C run time library
- network servers
- csh

Kernel extensions:

- increased number of file descriptors per process (from 20 to 64)
- increased kernel limits
- memory management
- signals
- system logging
- windows

Functional extensions to libraries & utilities:

- name servers
- system management
- routing

- compilers

*Q.* What about string copying?

*A.* VAX specific.

*Q.* Will it run out of process descriptors?

*A.* No.

*Q.* Running out of swap space?

*A.* Believe it's improved.

*Q.* In Ultrix?

*A.* At some point, probably.

*Q.* UDA driver?

*A.* Has been fixed in 4.3.

**The 'cat -v is dangerous' attitude is itself dangerous**
**David M. Tilbrook**
**Imperial Software Technology, GB**

*Abstract*

'This feels like a Republican victory party ...'

Vic Vyssotsky in keynote address at a USENIX conference

The so-called UNIX-philosophy has been preached from the pulpits of UNIX conferences by the high-priests of orthodoxy at many a conference. Does this zealous fervour have any connection with the failure of UNIX to make any significant advances in the recent years?

Why are there large areas of computer science that seem to be ignored by the UNIX world or why is that when some areas are attempted on UNIX they prove to be as unworkable or cumbersome as they were on the more traditional environments?

This presentation is a highly personal view by one who has been dismaid by the failure of the UNIX community (himself included) to make any significant advances in real-time systems (what ever they may be), software engineering (something palatable at least) and a variety of other problems.

*Comment*

David was sorry that Peter Collinson wasn't present, because his notes were so good. We were sorry too! To keep up with the speed of this speech and the number of foreign words wasn't easy, but here's the best we could do.

David started to point out that he really didn't have any content to motivate people to yell at him. He quoted an article in the Guardian about 'group thinking' comparing this to the UNIX community. As Tom Duff said at Toronto, 1975: "...the first clue that something is wrong with APL is that when two APL programmers come together they always start discussing extensions" — this is no argument against the talk — but a lot of effort is expended on minor things, the major problems are not solved by West Coast. A usual way today are to build a tool, use it for some time, then throw it away and begin again, what about doing this with UNIX? Anyone against? This is of course threatening to a lot of people, like those in the room here. But the next generation is not reached by expanding UNIX. In doing so, we must preserve some kind of Phoenix, rescue the good and throw away the bad.

A few personal views:

In the beginning, Kenneth said: "let there be a hierarchical file system" and it was good. A small system, not necessarily new ideas, real-time system and IPC left out at will, and then the problems began! The two man project was now beginning to evolve towards a Mega-project. The nature of the community meant that new things were included, but who knew what to avoid? Large areas are not developed, and if they are it's done by Unix novices, e.g. SCCS.

Eleven years ago he made a large system, a system with signals, pipes and file system, and I'd better be careful with those pipes! Today we have a large system with signals, pipes and file system, and I'd better be careful with those pipes still.

**/usr/man** was a remarkable database of documentation eleven years ago. At least when you were coming from a system with either doc's in libraries or non existant. It was so good, that we are still using it today, but it hasn't changed in eleven years. In the next iteration, I want some improvements. But, of course, *make*, a small set of SCCS, phone_numbers etc. was good. UNIX has grown a

lot, but not gracefully and not in compatibility with the original system. STONEMAN — the name showing the intellect of the man having made it — has borrowed a lot of ideas from Unix. The UNIX community are not moving forward any more, there is a need for radical change, a change of direction, a need to look for a successor. I put it out, that this forum should find out how to make it!

*Q.* (Mike O'Dell) You are right for all the wrong reasons. The facts are right, but one need to separate implementations of UNIX from UNIX itself. People adapted UNIX, but they didn't know what was behind it. If 4.2 and SUN etc. was put together...

*A.* Isn't this just another iteration?

*Q.* (Stroustrup) Supersetting is a bad idea! UNIX has the problem that none will be touching a new system because expectations have grown enormously.

*A.* Not clear.

*Q.* Happy that UNIX doesn't do any real-time handling. It wasn't designed to.

*A.* I didn't mean real-time should be put in, shoe-horned in as one could call it.

*Q.* Would it be a good idea to have a new project like redoing V7 and put it in the public domain, without AT&T behind and controlling it.

*A.* Standardization for those who want to keep with it — rethink for the ones to go on!

*Q.* What are the motivation behind your speech. Are you practising what you are preaching?

*A.* Can't do it myself. I'm an application person, but will support it. Where we were and why it went that way is the philosophy.

*Q.* There is a time constraint. The longer they dig, the bigger grave they dig.

*A.* Military dependent, but don't care about them digging their own grave — sorry! — but we can't keep the two things together.

*Q.* But the money is coming from the industry?

*A.* *Research* are not being commercially directed. If Ken was asked 15 years ago, he wouldn't have been able to.

*Q.* (de Ridder) Some researchers are looking into making the successor of UNIX. It's not a technical problem to develop it, but it will take some time.

**Recent work in UNIX Document Preparation Tools**
**Brian W. Kernighan**
**AT&T Bell Laboratories, USA**

*Abstract*

Document preparation based on *troff* continues to be an active area of research. This paper describes a new tool, *grap*, a program for typesetting graphs. *Grap* is a preprocessor for *pic*, rather than the usual *troff* preprocessor. Although originally intended only for document preparation, it has also been used for algorithm animation and exploratory data analysis, and has served as an "assembly language" for several compilers for specialized graphs.

The paper also describes enhancements to *pic*, particularly built-in functions and control-flow primitives, that permit the creation of figures of some complexity.

*Comment*

Kernighan introduced the speech by noting that we all owe something to D.Knuth (TEX and METAFONT). He then went on to talk about the possibilities in the UNIX environmenht.

*Tbl* can be used to create large tables, which contain lots of information, which cannot be recognized easily. Graphs are much easier to comprehend quickly. This was introduced with a number of unsuccessful tries:

```
                options
numbers           v
------->      GRAPH      ------>      plot      ------>   PLTROFF   ---->   TROFF   -->
              language

                options
numbers           v
------->      GRAPHIC    ------>      pic       ------>     PIC     ---->   TROFF   -->
              language

              macro defs.
macro             v
------->        PIC      ------>    TROFF     -->
 calls
```

Should have written a new language — done now:

```
grap
------->   GRAP   ------>   PIC   ------>   TROFF   -->
input
```

Basic idea:Should give a **minimum** specification of a graph — given just numbers, it produces a graph, with suitable axis, scaling etc.
Can control axis labels, ticks on axis, line types, and may even escape to *pic* commands to draw arrows etc. Axis may be logarithmic.

In fact it is a small programming language, not rich but useful. Still another preprocessor to *grap*, *scatmat*, which is are used to plot scatter matrices.

*grap* and *pic* shares a lot of code, and Brian has now integrated *grap* into *pic*, which has been modified also, e.g. more typesetter independent, no internal array limits and other enrichments.

Examples:

```
.G1
1911  255.4
1915  252.5
.     .
.     .

grap [file] | pic | troff

.G1
label bottom "World ...."
label left ....
ticks left from 220 to 260 by 10
ticks bottom at 1920,1940,1960
draw solid
copy "mile.d"
```

*Pic* changes:

- features from *grap*
- output in inches, not in units
  (".... could easily be changed to centimeters for more advanced countries", as Brian stated).

TROFF changes:

- 30% faster
  (width cache, hashed name tables)
- ```
  \s+(dd                  for new eqn
  if line-number filename    like #line in cpp
  \X'whatever'            passed through to output uninterpreted
  \C'longname'             longer character names
  \N'nn'                  absolute number
  ```
  ```

.fp n XX long-font-name  synonym for long font names

*Nroff* changes:
Revised to use ASCII terminal table instead of reading a.out file format: ".... done because they were going to set up a committee to discuss ways to do this" — Brian did it in an afternoon.

*Q.* Anywhere in the future where we can see what we get on the screen?

*A.* There are such systems, but not sure the WYSIWYG is useful in all cases, e.g. complicated graphs and linguistics.

*Q.* What about other languages, e.g. in *troff?*

*A.* Yes, it would be a great improvement, the notion of a preprocessor has been tried. But, it may break many things.

*Q.* What about improvements on *eqn?*

*A.* Yes, several — in danger of being shot by the session leader (Helen), he showed another slide.

*Q.* Availability?

*A.* In Writers Workbench II — perhaps in the end of the year.

*Q.* Changes to intermediate language output by *troff?*

*A.* Only trivial extensions (superset).

*Q.* X/OPEN — want more fonts!

**Principles of Font Design for Personal Workstations**
**Charles Bigelow**
**Stanford University, USA**

*Abstract*

The personal workstation offers powerful tools for the computer literate, but these tools depend on typography: legible fonts in meaningful compositions. Digital typography is constrained by the limited resolutions of current screen displays and printers, which cannot faithfully reproduce traditional analog letterforms. To optimize the legibility of digital text, new fonts must be designed for screens and printers. These designs will be most effective if they consider the mechanisms of the human visual system, the evolutionary principles that shaped our modern alphabets, the technology of the digital image, and the conceptual structures underlying typographic variations and compositions. The workstation requires a digital typography that preserves the fundamental features of literacy while expressing them with new clarity in a new medium.

*Comment*

David Tilbrook was very enthusiastic about him coming, especially for his gaily coloured tie!

There are three levels to consider:

- readability
- legibility
- deciferability

In history, development has gone from iconic to symbolic signs or abstract entities. Today we have gone 3000 years back using icons to 'name' entities, in fact the opposite way of our ancestors.

Letters have a very long history, not very many tech. stuff has survived that long. They are very archaic and we still are able to read roman stone carved letters. To begin with, it was only big letters, but handwritten text became lowercase letters due to physiognomy. Later both upper and lowercase letters were introduced. With printing, a development of letters to very fine sculptured (art) letters began.

Nowadays, what we want is a regular rhythm of smooth shapes, black on white, formed by nice sculptured letters — what we get are "computer printout" and screens!

Font design must be based on **how the eye works**. An example of a dot-matrix printed word with a big distance between the dots was shown compared to the same word in ordinary print, where only part of the letters were seen. It was very clear, that the latter was much easier recognizable. The eye recognizes the smooth curved edges to a high degree.

Furthermore, there is an optimal spacing between types, which is not linear over sizes and parts of the letter which traditional typesetting did just right, but which is lost in current electronic typesetting.

A list of what we need:

- better fonts for workstations
- should not remove curves and diagonals
- differentiation of size and of style
- device-independent outlines or device-dependent bit-maps
- point variations
- how prevent degradation on laserprinters
- METAFONT — Knuth designed it for his own purposes
  "....mathematician who is still a programmer!"
- designed by hand — and then digitized, a system of curved splines
- split letters in fundamental parts

*Q.* (Mike O'Dell) When can I get the fonts you showed for my laserprinter system?

*A.* They are available for other devices.

*Q.* What about terminals, is black on white the best?

*A.* White on black gives often flicker. For people doing text processing it's very much better, because they refer to papers with black on white.

*Q.* Any advice for people doing low-resolution fonts for terminals?

*A.* Don't over simplify them. Get an even spacing. Use proportional or carefully tune mono-spaced. Build them by hand.

*Q.* Any experience with grey levels?

*A.* There is a lot of research. But problems, how to go from high-resolution to grey levels? Some research indicates that it may be more fatiguing than pure black & whited, the eye is looking for sharp edges.

**Smalltalk on UNIX Systems**
**Georg Heeg**
**Universitt Dortmund, D**

*Abstract*

The talk will describe experiences with the implementation and use of Smalltalk interpreters on MC68000-based Unix systems. It will try to relate some basic ideas of the Smalltalk programming environment in the process.

*Comment*

Heeg refered to the speech of Kernighan. In Smalltalk, you would have three windows, one with graph, one with data and one with commands. Changes should affect the graph immediately!
Needs one VAX per user, bitmap screen and a mouse.

Smalltalk-80:
**Goals:**

- support for nonspecialized computer users
- object-oriented paradigm
- class descriptions

**User interface** (the most famous thing):

- window and mouse — ("all later systems bears this inheritance") the meaning of the mouse buttons are uniform throughout the system.

**Objects (basic concept):**

- everything is an object
- belongs to classes — classes defines the actions to be done. Classes are also objects.

Provides a very advanced programming environment. Everything in the Smalltalk environnment is in **source**.

**Smalltalk -> UNIX access:**

1) file system

standard: directory and files can be accessed
advanced: structured filesystem editor with graphical representation

2) program execution (batch)
Smalltalk communicates with string to/from a UNIX filter

3) terminal emulation on certain advanced systems

4) thoughts about representing UNIX parts as Smalltalk objects.

*Q.* Can one use normal UNIX programs in the Smalltalk environment?

*A.* As filter, batch via files, or if it exists — via a terminal emulator.

*Q.* Any important applications?

*A.* Very advanced database/spreadsheet programs.

**Simula and C, a Comparison**
**Georg P. Philippot**
**NCR Education Nordic Area, N**

*Abstract*

A newcomer in the UNIX world, *Simula* is an object-oriented language of the same family as that of C. They are both block structured, machine independent, general purpose high level languages. Though in principle you may use any of the two for any given task, there are individual areas for which one is better that the other. This paper describes the similarities and differences between two great languages, assuming some basic knowledge of C. It is hoped that this may give the audience some appetite for learning more about *Simula* and how it fits into a UNIX environment.

*Comment*

Algol(father), Simula, C, Pascal and Ada are all related. They look a little bit different, but development and compilation of programs are the same.

An overview of common features, C advantages and Simula advantages was given.

Conclusion:

- Simula domain:
  safe development, complex programs
  fast development, simple programs
- C domain:
  when speed can really be increased by an order of magnitude
  binary file and bit manipulation

*Q.* Availability?

*A.* For the VAX and the SUN — through Simprog in Stockholm.

*Q.* Why not choose ADA?

*A.* ADA is far too complex (overkill).

*Q.* Are strings in Simula null-terminated?

*A.* No.

**A C++ Tutorial**
**Bjarne Stroustrup**
**AT&T Bell Laboratories, USA**

*Abstract*

This is a tutorial introduction to the C++ programming language. With few exceptions C++ is a superset of the C programming language. After the introduction, about a third of the text presents the more conventional features of C++: basic types, declarations, expressions, statements, and functions. The remainder concentrates on C++'s facilities for data abstraction: user-defined types, data-hiding, user-defined operators, and hierarchies of user-defined types. Finally there are a few

comments on program structure, compatibility with C, efficiency and a caveat.

*Comment*

To correct some mistakes, there are no languages called 'C' or 'Old C'. The majority of C-programmers at Bell Labs do not use C++, many of them are involved in maintaining old programs! But most new programs use C++.

C + Simula67 classes + type checking + operator overloading -> C++.

Basic idea is Simula class. In Smalltalk they took away any idea of efficiency and gave you user-interface! C++ goes the opposite way. C++ allows the programmer to express ideas clearly and precisely.

- programs should contain more info in the same space — i.e. shorter programs
- modularity
- fast — both in compilation, it finds errors faster — presently 30-100% slower in code generation — and execution.

For all practical purposes C++ is a superset of ANSI-C. More compatible than most will believe. A lot of work has been done to do this right.

A class declaration contain a private and a public part. It may contain function declarations, which contain argument types and they may be overloaded — determined on the argument types at call time.
Operators are declared in the same way as functions.
A new datatype called *references* is introduced. It may be used to implement *call by reference*. Stroustrup doesn't like call by reference, but examples were given showing that you have to use it sometimes.

New operators defined: *new* similiar to *malloc* but able to determine the size from the type itself. *delete* to free it again.
You can also have inline functions.
Generic types must be done by macros, which is unpleasant but doesn't cost at execution time.
Classes may be expanded to avoid the traditional 'copy and modify source' problem. They are called derived classes.

C++ is expressive, easy to learn and is based on proven concepts.

*Q.* Any protection against interrupts (signals)?

*A.* Not taken care of.

*Q.* Major changes since release A?

*A.* Default scope of a global name is the whole program.

*Q.* Availability?

*A.* Available to Universities from UEL.

**Nottingham's experience of X.25 under 4.2BSD**
**William Armitage**
**University of Nottingham, GB**

*Abstract*

When looking for an X.25 implementation, we came across the University of British Columbia's EAN X.400 mail system. Tucked away in a corner, hardly mentioned, was an implementation of the CCITT domain for the Berkeley 4.2 kernel. We have successfully adapted this code to the UK academic environment.

This talk first examines UBC's implementation of X.25, and goes on to describe our experience in adapting the code, covering layering protocols over X.25 (Yellow Book Transport Service and RFC877 IP Encapsulation) and performance issues.

*Comment*

A very technical description of the protocols and the interfaces was given. For a full understanding we must refer to the papers from the speaker.

**Error recovery for Yacc parsers**
**Julia Dain**
**University of Warwick, GB**

*Abstract*

We aim to improve error recovery in parsers generated by the LALR parser-generator *Yacc*. We describe an error recovery scheme which a new version of *Yacc* automatically builds into its parsers. The scheme uses state information to attempt to repair input which is syntactically incorrect. Repair by alteration of a single token is attempted first, followed by replacement of a phrase of the input. A parser for the C language is generated from existing specifications and tested on a collection of student programs. The quality of error recovery and diagnostic messages is found to be higher than that of the existing portable C compiler. The new version of *Yacc* may be used by any current user of *Yacc*, with minor modifications to their existing specifications, to produce systems with enhanced syntax error recovery.

**A Prolog description of a symmetric solution for automatic error-recovery in LL(1) and LALR(1) parser generators**

*Abstract*

It is shown that the LL(1) and LALR(1) parsing schemes are too close to each other to justify the quality gap between their respective error-handling capabilities. Even within the constraints of a one-symbol-lookahead strategy a reasonable error-recovery with clear and precise syntax-derived messages is possible without adding any explicit error information.

A considerable improvement is obtained by synchronizing on so-called fiducial symbols. Prolog was suitable to find a formal specification of the set of fiducials for any grammar. The fiducials optimize error-handling without disturbing the efficiency and robustness of the underlying parsing scheme. They are implemented by context parameters in LL(1) procedures and by automatic insertion of error tokens in LALR(1) grammars.

Available syntax descriptions in YACC of ADA and AWK are used to demonstrate the effectiveness of the method.

**Screen based History Substitution for the Shell**
**Mike Burrows**
**Churchill College, Cambridge, GB**

*Abstract*

Several UNIX command interpreters now incorporate a history mechanism to assist interactive users in repeating or correcting commands. Many of these are similar in style to Bill Joy's C Shell, which provides a simple line oriented interface and a numbered history list. More advanced shells, such as the Korn Shell, allow screen editor facilities, but still reference history items by an event number or an explicit pattern matching syntax. This paper describes an interface that allows history substitutions to be performed as a command is typed and with minimal user effort. Entire lines or single words may be substituted with equal facility. The technique has been fully integrated with more normal editing features and Tenex-style filename completion.

The interface is extremely simple to use even with large history lists, without requiring the user to repeat "event numbers" or to review the history list periodically. Versions of the interface have been added to various existing shells and have been in use for several months, proving popular with both novices and experienced users. The lastest version has been implemented as one of a number of enhancements to the System V.2 shell.

*Comment*

To use Mike's own words, he is one of those people who likes to add new features to the shell when he has nothing else to do.

The M-shell, as he calls it, has a built-in editor (uses termcap/terminfo), a history mechanism based on command completion, job control, tilde (~) expansion, and functions.

The editor is not based on either *vi* (no modes) or *emacs* (no windows), still it is very easy to use, because it will adapt itself to the keyboard of the terminal, using the special keys (e.g. insert char) if

they are available.

This talk focused on the history mechanism, which is completely different from the event number based mechanisms used in C-shell and Korn shell. The command completion is very simple to use: type in the first few characters of the command, hit the escape key, and the rest of the command is filled in from the history list. If the command from the history list was not the correct one, hit escape again, and the next command that matches the prefix is shown. Commands are shown from the newest to the oldest, with duplicates removed.

The same mechanism can be used to complete a command line argument: hitting escape brings forward all matching arguments from the commands in the history list one at a time. When all the argument lists are exhausted, matching file names from the current directory will be substituted.

Experience has shown that in most cases, one hit on escape is enough, and more than three is very unusual.

*Q.* Doesn't the automatic adaption to various keyboards make people confused?

*A.* If they become confused, they can always use a standard control sequence.

*Q.* Can the entire history list be edited?

*A.* It is possible.

*Q.* Start-up time?

*A.* Interactive start-up time is less than one second.

*Q.* Availability?

*A.* Needs a source licence, but mail me (mb@cl.cam.ac.uk)

**European Languages in UNIX**
**Conor Sexton**
**Motorola International Software Development Center, IRL**

*Abstract*

This document describes the approach adopted by Motorola to the internationalization of a UNIX System V derived operating system — Convergent Technologies CTIX 3.0. The initial goal was the provision within CTIX of character sets enabling easy use and interchangeability of U.S. English, French-Canadian and six European languages. The problem is broken down into its constituent parts, and the solution, as well as the scheme of character-set representations employed, is outlined. The many difficulties encountered during the development and testing are described. Finally, an insight is given into the direction of future Motorola development in the area of international UNIX.

*Comment*

The current implementation uses an 8 bit internal representation of the character set, which is not sufficient in general, but to go further, Motorola waits for AT&T to put 16 bit characters into System V. Even the limitation to 8 bit characters gave lots of problems with both the standard utilities, and application programs. For example none of the editors worked with 8 bit characters, which gave some bootstrapping problems. The hardest job was to test out all the utilities and applications to see whether the job was done.

Their system uses a special terminal with down-load facilities for keyboard mapping, fonts, and dead key sequences (e.g. 'e). The terminal uses three byte codes which represents a 16 bit character in a Xerox standard character set; these codes are converted to and from a single 8 bit character in the device driver.

*Q.* Does the sort utility work?

*A.* Not at all!

**The COSAC X.400 Message based Network**
**Claude Kintzig**
**CNET, F**

*Abstract*

COSAC is a message based network, that uses the principles as described in the CCITT X.400 documents. The talk will include a short introduction into the principles of X.400.

1) MHS-X.400
   - presentation of the model
   - services
   - P1 and P2 protocols
2) COSAC Version 5
   - COSAC under UNIX
   - functions: message transfer, mail, file transfer
   - COSAC port to IBM, VAX, MULTICS
   - network growth
3) The future
   - Additional functions: job transfer, distributed bulletin board, directories.

*Comment*

The current system is written in Pascal with a clean adapable interface to the operating system. There are mail interconnections between UNIX, MULTICS, EUROKOM, MISSIVE, VAX/VMS, and IBM. Present developments aim at interconnections between COSAC and EUNET, and between COSAC and CSNET.

*Q.* What network do you use?

*A.* X.25

*Q.* Have you interconnected to EAN, the X.400 system from University of British Columbia?

*A.* We are planning to do that.

*Q.* What UNIX version?

*A.* Version 7.

**Communications Solutions for Mainframe UNIX**
**Jim Hughes**
**Summit Operations, UNIX Systems Development Lab, Amda**

*Abstract*

Bringing UNIX to the world of high speed mainframe computers presents major challenges to communications. UNIX was developed around full duplex asynchronous terminals, which are not commonly used in large data centers.

In creating the UTS port of UNIX to the 370 architecture class of processors, special interfaces and protocols were established to allow the use of full duplex, asynchronous terminals as well as the more standard bisynchronous terminals.

With this full duplex support software users can interface with UNIX applications in exactly the same fashion as would be used on mini- or microprocessor based implementations of UNIX. Thus an editor such as *vi* is available.

This paper discusses the technical aspects of full duplex support software, as well as current interfaces for X.25, Ethernet, and other networks. Through these communications solutions and the power of UNIX on a mainframe computer, the end-user has an effective tool for creating and accessing corporate applications.

*Comment*

A long (and rather dull) presentation of full duplex/half duplex/echoplex (and he kept on and on with it!), with special attention to the problems on the mainframe side, was given. Amdahl had to make special code for the communication processor to make it work. Problems with ASCII contra EBCDIC on the mainframes was discussed too.

*Q.* Other comm.processor than 4705?

*A.* Yes, 7171!

*Q.* SNA?

*A.* Investigation going on, but X.25 is seen as a better solution.

*Q.* Ethernet solution?

*A.* Some users are looking into it, but it's new technology whereas 4705 is tested equipment.

**The ANSI Draft Standard for the C Programming Language**
**Mike Banahan**
**The Instruction Set, GB**

*Abstract*

The ANSI draft standard for the C programming language (ANSI X3J11) has been published. This talk will discuss the work of the committee, the major new features or changes that the standard includes, and the likely effect on current and future programs written in C.

Details will be given on how to obtain a copy of the draft, and where to send comments for consideration before publication of the 'final' proposed standard.

N.B. This is a talk, not a technical paper. No serious technical issues will be addressed, no Nobel-prize winning concepts introduced.

*Comment*

ANSI X3J11 C standard.
Who are they? ("....travelling for the dinner?")

- approx. 50 committee members
  - industry
    — users (some)
    — suppliers (lots)
  - educators
  - observers

Almost anybody may join in, there are presently 2-3 European members.

What is the standard about?

- define syntax of language
- standardize library routines
- consider external factors
  - character set
  - file handling

Aims:

- remain true to the spirit of C ('....but cleaned up')
- improve portability ('....hard to do')
- break fewest existing programs
- introduce limited enhancements
- remove ambiguities
  e.g extern-declaration in inner scope has been handled in 3 different ways in the past.

Conformance:
A conforming implementation will:

- guarantee mandatory semantics
- define certain others
  — right shift extension
- leave some undefined
  — a <<= BITS_PER_LONG+1;
- some things are not specified
  — f(getchar(), getchar());

Classes of programs:

- erroneous — use defined behaviour
- 'non-portable' — use implementation-definition behaviour
- conforming — use only defined behaviour

Language:

- complete rewrite of reference manual.
- Rules on **extern** and **static** declarations disambiguitized.

Identifiers:

- up to 31 characters
- different objects must have first 31 characters unique
- names of the same object should be equal in more then 31 characters
- external identifiers — 6 chars, monocase only — 'blaim brain-damaged archaic linkers'

Types:
Some new types

- signed char
- unsigned char
- char
- [signed/unsigned] [long/short] int
- [long] double

Storage classes:
New storage classes

- volatile — may be subject to external changes
- const — "I promise not to write to this"

Void
The type with no type (eh!). Already in modern C, but (void *) is the new universal pointer.

Arithmetic on float expressions, may now be done on float, not double.
enum IS int
structure members have different space.

Functions prototypes:
If used once, they must always be in scope of the calling function.
Arguments are converted — sqrt(4) is ok.
Scope of function argument names extended into first block:

```
f(x) { int x;    —    now gives an error
```

New function def.:

```
f(float f_arg, register int i_arg){
```

Libraries:
Two file types recognized

- text
- binary

binary — 1:1 correspondence to file contents
text — map \n to \r\n

Effects?
How much code does this break? — New keywords always breaks code!

*Q.* Is the manual readable by users?

*A.* The reference manual is for compiler writers.

*Q.* Does the function prototype names have to correspond to the formal names?

*A.* No, they are ignored in prototypes.

*Q.* Sizeof?

*A.* The size of *sizeof* is implementation dependent (defined in header file).

*Q.* Is *entry* not defined?

*A.* No, dropped as a reserved word.

*Q.* Union initialization?

*A.* Was not included.

**Standardised Art as a Vehicle for Enhancing Market Penetration**
**Mike O'Dell**
**Group L Corp., USA**

*Abstract*

This talk will examine the curious notion of "standardized software" with specific concern for the implications of this notion for groups interested in advancing the state-of-the-art. The approach will be to take a surprisingly valid analogy, and then push it well beyond the breaking point.

*Comment*

Marketing resembles system programming! More people do it than are capable of doing it. Any ad claims to be "standard" —

'From Now on, Consider it Standard'

looks nice, but what does this mean? What does standard mean? Let's look at some definitions from the encyclopedia:

1. A flag, banner, or ensign.
2. An acknowledged measure of comparison.
3. A pedestal, a base or stand.

(a list of over 10 definitions were given)

And as an adjective

- of average, but acceptable quality

"isn't English wonderful?"

Standards should be minimums not maximums for a system!

Let's look at Software:

- erroneous perception of portability
  - CP/M -> all ran on "the same" Z80 systems ("....simple, nothing to do!")
  - MS/DOS -> **all** PCclones
- vast array of software for sale everywhere
  - some programs are wonderful
  - many programs are of questionable quality, if not down-right worthless
  - cannot see their worth before you buy it
    - due to both number and marketing practice
    - why are there 150 word processors for IBM/PC — is this quality?
  - the captive software market has been incredible lucrative for some
    - if the customers still pay they must like what they get (?)

How does this apply to Art?
In some funny ways. How does it apply to UNIX?

- there are limits to compatibility
  *68000 binaries will not run on *86 machines!
- unmet needs — people needs networks
- there is a difference between utility and dogma
  "....you can't just look at a program to see if you look at it, like you can do with art"

Good things about the UNIX and C standards

- standards shouldn't innovate
- statement of minimum functionality
- stability

Bad things

- deify and encourage the assification .... antique and seriously inadequate technology.
- perpetuate widely held notion

Many people hope it will be lucrative (when UNIX becomes standardized), may it R.I.P. (Rest In Peace).

*Q.* Problem with standard! (?)

*Q.* Usenix lots of arguing — already 90% moveable — establish min. standard, not max.

*A.* Yes, we can easily move! Version doesn't matter this much.

*Q.* Standards are good in some ways and dangerous in others. The worst risk is that it may lock people to old technology.

**The X/OPEN standard**
**Jacques Febvre**
**Bull Sems, F**

*Abstract*

The X/OPEN group has been created by major suppliers of computer systems. Their objective is to establish a common applications environment providing portability of applications at the source code level on a wide range of machines.

This common application environment is described in the X/OPEN portability guide presented here. It contains:

- The UNIX System V interface, as described in the AT&T "System V Interface Definition"
- The C Language.
- The FORTRAN 77 Language.
- The ANSI 74 COBOL, extended with accept and display verbs as defined in Microfocus Level II COBOL.
- A definition for ISAM which is a major subset of the C-ISAM product published by Relational Database System Inc.
- A definition of 5.25" floppy format for source transfers

Benefits of this Common Application Environment for independent software vendors, end users and computer suppliers are also presented.

*Comment*

One of the marketing issues of the X/OPEN work is to increase the perceived credibility of UNIX, because the technical community is talking it down. For example, the existence of many versions make people complain about portability problems, whilst in reality, porting code between UNIX systems is magnitudes easier than porting between almost any other type of system. In Europe, the big manufacturers have not been able to cooperate by tradition, so the X/OPEN common technical infrastructure work is untraditional.

The X/OPEN standard is a minimum standard; so there is still room for competition. The kernel extensions specified in the SVID will be individual options, except that shared memory, semaphores, and messages may be omitted.

X/OPEN also participates in standardisation and internationalisation work, because the spirit of the cooperation is to use existing standards. The X/OPEN group is open for new members to join it.

*Q.* Why 5.25" floppies?

*A.* They are already there, and we need them now!

*Q.* Why have you ignored Berkeley?

*A.* We haven't. They are important input, but our aim is at the commercial market.

*Q.* What influence have AT&T had on the X/OPEN standard?

*A.* The X/OPEN portability guide was made in close connection with the AT&T SVID team.

*Q.* Why didn't you use the *de-facto* standard, Xenix?

*A.* The portability guide does consider Xenix.

*Q.* When will the X/OPEN members use the new standards?

*A.* They will move towards it in the future.

*Q.* You have to do this right now, or it will be too late!

*A.* We agree, so we are interrested in cooperation with the user communities.

*Q.* Why do you use *cpio*?

*A.* It is in SV, but we will support tar as well.

*Q.* (Jim McKie) Why is the X/OPEN book so expensive. To get people to comment on it, it should be given away for free; remember, we are here today because somebody gave something away for free...

*A.* The book was published by a professional publisher, which determined the high price.

**End of the Conference**

On behalf of the Conference organisers, Nigel Martin summed up the lessons from this conference:

EUUG will try to book one hotel in Florence.

The tape distribution was a success (but as was shown later, not without problems).

He thanked all that had participated in making this conference work, especially AT&T who flew over certain speakers from the US.

The proceedings from the conference will be sent out to all participants.

The conference ended by giving Jim McKie a present for his long and loyal service for the EUUG, now that he is leaving for the US. The present was, of course, the X/OPEN portability guide, so he got it for free after all.

**D2B** (a French UNIX software house based in the PARIS area) and its sister company **ORIANE** (the French source licensee of the **UNIFY DBMS**) are looking for:

**a UNIX specialist (minimum 3 years of experience)** having an in-depth knowledge of DBMs and willing to improve existing DBMs and support the development of new applications

**a UNIX specialist (minimum 3 years of experience)** with a knowledge of LANS and ISO standards

Both positions are based in the PARIS area
Write in the first instance to BOX NLS1, EUUG, Owles Hall, Buntingford, Herts. SG9 9PL, England

# Impressions from the Industrial Sessions given at the EUUG Autumn 1985 Conference in Copenhagen September 10-11, 1985

*Bo Svarre Nielsen, Hewlett-Packard Danmark A/S*

**September 10, 1985**

The first day started with some tutorials given by *The Instruction Set Ltd. (GB)* , - the first about Lex and Yacc presented by *Mr. Jim Olroyd.*

The tasks of a scanner and a parser were well explained. The idea of Lex and Yacc was presented on a conceptual level with some examples including the well known calculator example.

*Mr. Bill Fraser-Campbell* gave a tutorial primarily on device drivers and secondly on functions in 5.2 shell.

The tutorial on device drivers was quite technical and was mainly intended for people having a source license although other people (like me for example) got a better understanding of the way specially the tty driver works.

The explanation of functions in 5.2 shell was fine with a lot of good examples. There were lots of good ideas in the section on functions to implement user environments.

After a small lunch (very small, because all the resturants vere closed leaving most people to buy bisquits, apples etc.), the President of *Yates Ventures Inc. (USA), Mr. John Kiefer,* gave an overview of the UNIX market.

His main point was a theory that the UNIX market is in good growth although not as fast as some UNIX people believe. Single user systems like MS-DOS will have the highest growth rate, and UNIX is not very well recognised in the large operating systems world. However, UNIX will be the de facto standard for medium size multi-user operating systems. Standardization is extremely important in order to maintain the growth of UNIX. Presently there are more than 35 versions of UNIX.

The next session was an overview of the products from *Relational Database Systems Inc. (USA)* given by the Vice President, *Mr. Skip Hawk.*

The goal of this company is to have full compatibility from micro systems to mainframe for their data management software. Their product line consists of several data mangement packages, e.g. the well known C-ISAM file access method, the relational database system INFORMIX, the query language INFORMIX-SQL, a screen builder and a report generator.

*Mr. Neil Urquart, Spinx Ltd. (GB)* gave a personal view on the problems in distributing software. The main problem in distributing software was not really the number of different vendors, but the number of different software packages. These two multiplied together gave a fair number of combinations, but with a third dimension, which includes all different versions of hard- and software, the number of combinations became unmanageable.

The last speaker on that day was *Mr. Daniel Flickinger, CCL Datentechnik AG (BRD)* who talked about the problems of developing and maintaining a program system of 230.000 lines of C-code for several UNIX systems.

It turns out that in practice the UNIX programs are less portable than one might think. In order to keep programs compatible, it was necessary to use only a part of UNIX library functions and even rewrite some of them, e.g. the string handling routines.

**SEPTEMBER 11, 1985.**

In the morning session on September 11, *Mr. P. J. Cameron, Plessey Microsystems Ltd. (GB)* talked about "Development Methods for High Performance Commercial UNIX Systems".

He stressed the idea that high performance systems can only be made by a team of hard-, soft- and firmware specialists so that together they can work out the optimum design. He gave a couple of examples of I/O cards designed for UNIX systems by such a team.

*Mr. Peter Frenning, Altos Computer Systems A/S (DK)* gave a presentation of the Altos computers concentrating on network types and how the UNIX filesystem could be extended to run in a distributed system.

Just before lunch - and on that day we could get lunch - *Mr. Tony Heywood, Redwood International Ltd. (GB)* talked about integrated office software.

He pointed out the traditional differences between dedicated systems with emphasize on capability, but with little possibility for integration and integrated systems with less capability and how the product UNIPLEX tried to achieve both.

After lunch, *Mr. Peter Grøndahl Nielsen, Danware Data Aps (DK)* explained how DAM+, a database tool for UNIX, worked. Its most important characteristics were the duplication of data imbedding the usual transaction log in the database which assured that the database always was consistent.

Construction of user interfaces was the subject for the next two speakers, namely *Mr Henrik Lindberg, Carl Lamm Systems AB (Sweden)* and *Mr. Paul Clarke, Data Logic Ltd (GB).*

Using different approaches, both gave possibilities for construction of general purpose user interfaces.

*Mr Steve Bonniwell, Apollo Computer A/S (DK)* continued after a break to present the graphics product structure used in the Apollo computers.

Following this *Mr Ivan Costarov, ICCC Aps (DK)* presented a very technical talk about the problems in implementing UTS 4000 communications systems under UNIX.

The very long day ended with two speakers - *Mr. Mike Southon, The Instruction Set Ltd (GB)* about the criteria for selection of UNIX training and *Mr Basil Cousins, ICL (GB)* about the commercial relevance of X/OPEN to discuss standardization and the reasons for this.

# YET ANOTHER GOOD CONFERENCE

*Sebastian (Snoopy) Schmitz*

**The Purpose of this Report**

My goals are to amuse and inform (in that order, but in nearly equal proportions). It is also intended that it everybody who did not go will become insanely jealous and will make sure they attend the next conference.

All views expressed are my own and not my employers. All usual trademarks and footnotes are acknowledged. Shame that one always has to say this, but I am no friend of hot water...

**Day 0 (Tuesday)**

This was travelling day. Had my first frustration because I arrived at work, having forgotten my shampoo. This was terrible. A mad rush to the coiffeur at Munich airport was necessary. He demanded DM 7.50 for a shampoo that normally costs DM 3.00 !! I tried for a discount (i.e. batted my eyelids at him in a VERY obvious manner - but no avail. . .)

Then I rushed into the departure lounge, where I batted my eyelids at the stewardess at the checking in desk. This was rather more useful, as I found out that Duesseldorf was congested. I seized the opportunity and hopped on an earlier plane, as my margin for my connecting flight to Copenhagen was pretty small. Ah, yes. The stuff executives are made of: quick decisions ....

My first ever SAS flight was a very pleasurable one. The stewardesses are actually *nice* and they smile, and they joke around with the passengers. I just could not visualise the efficiency-ticking Lufthansa lass doing same. Any person, who is tall, blond and handsome (like moi) is addressed in 9600 Bd Danish and the first question is not: "Do you want something to drink ?" but rather "Do you want Whisky or Gin ?". The guy next to me had three beers and two whiskies. That would have killed me.

Arrival Copenhagen. Steady drizzle, friendly people.

**Day 1 (Wednesday)**

The day started off with another frustration: Danish hardware does not support style. That is to say that the plug of my hair-blower did not fit the wall receptacle in the hotel. So much for the trendy/new wavy hair cut. Sigh.

Arrived at the Bella Conference Center with wet hair, where I spotted lots of familiar faces - Achim Brede (DEC), Daniel Karrenberg (the guy who always bounces back our mail from unido and sends us all those bills.). Also met Jean Wood from DEC, who immediately clasped my neck firmly in her hands and started strangling me right there & then. (Achim & Daniel were too stunned to react, not that they would have). She shrieked: "So I finally got you, you who never says anything nice on the net about the MicroVAX2 !!! AAAAArrrggghhh...". I only managed a choked reply: the emotionality of the scene was getting to me. Jean is a really nice person, we got along well. Soft hands, too.

So then it started: I was actually there from the beginning...

**Welcome by Keld Simonsen/DKUUG**

This time there were about 350 attendees; the EUUG membership has reached 1700... The GUUG (the German chapter) is pronounced Gaagh, Keld politely refused to pronounce the name of the FUUG (Finnish UNIX users group).

He asked why people come to Copenhagen, and responded to his own question with:

- Food
- Cakes

- Ice cream
- Beer
- The Tivoli
- Jens Olsens World clock
- David Tilbrook

Of course, he got it wrong. I came for the SINGING for he sang a very spirited version of "Welcome to Copenhagen", a derivative of his latest hit-single "Welcome to Copenhagen". It was very nice.

**AWK by Brian Kernighan**

Brian Kernighan is one of the original superstars, who manages to fill the largest venues during his world tours... He gave a talk on the latest developments with AWK. AWK, as we all know, is something we cannot do without. Its most famous error message is "Bailing out near line 1". Actually he made the point that AWK was only intended for one line programs; now he has the problem that people come in with 1000 line programs. He enhanced AWK with the following :-

- A close statement for files
- A system call (like C)
- Trig functions
- Functions (most of those 1000 liners were repeated code due to lack of these).
- Argc & Argv
- Better input functions
- Dynamic regular expressions with variable field separators

Lots of the new features are really necessary and it sounds as if AWK is really getting its act together. Wish I could have it. Unfortunately for us 4.2 sites it is not yet available.

The more interesting point (for me) was that AWK stands for Aho, Weinberger and Kernighan (the three initiators). I seriously always believed it stood for awkward. Oh well - you never stop learning.

**Coffee and Tea and Kringeler**

Kringeler is Danish for goodies...

**Object Formats by Martijn de Lange**

Martijn is from ACE, a company that buys their coffee from money made by selling SysV kernels as binaries and support, consulting etc.

These people have extended the COFF (Common Object File Format). Object does not refer to modern art but rather binaries and relocatable files (a.outs and .o's). They have kept the SysV semantics (for all the good it will do them). He highlighted the differences between the two systems (SysV and 4.2) re. memory management. They have extended SysV with page attributes which make for easier dynamic loading of modules etc.

This was a good talk (i.e. Martijn knew what he was talking about), and he gave a witty UNIX history. He claims that Unix is having its midlife crisis "after it was full of Joy, but then it was in the SUN too long". Groan. But we all laughed.

**Lunch at the exciting Cafeteria**

Achim complained that the meals are too small. We tried to goad him into getting himself a second one - but no use.

**KERMIT by Johan Helsingius**

Johan is from the unspeakable FUUG (Finnish Unix Users Group). He gave an introduction to the KERMIT file transfer utility, which now runs (would you believe) on CRAYs. He claimed that KERMIT utilizes 90-95 % of the available serial line bandwidth, whereas uucp only uses 60-70%.

The highlight of his talk was the slides he showed us from Finland. The FUUG is trying to convince everyone that one of the next conferences should be held in Finland, under a midnight

SUN, no doubt. He showed very inspired pictures of DEC field service engineers (on sleds) on their way to customer sites (little green tents on vast sheets of ice) where the customers (moose) accumulate.

**ISO Networking by Sylvain Langlois**

"Panic Trap" Sylvain (of eunet.general fame), did a lot of work putting the ISO protocols into the UNIX kernel. His talk gave an introduction of what ISO is all about and the status of his work.

**Sendmail by Miriam Amos**

Miriam is from the Berkeley Division of DEC. This means she is employed by DEC, but is on loan to Berserkeley. Actually the idea of an "employee on loan" seems to be a mind-staggering concept for most Americans, as she told me later. Miri helps Berkeley getting 4.3 out the door and onto our VAXen.

She related to us, how happy she was to go to Berkeley, because she does not have to take care of sendmail at DEC anymore. She tap-danced all the way to Berkeley, because she got rid of it.

One of the problems with sendmail is that it can (and does) lose mail, if you kill the process at the right time (for example). This is bad: THOU SHALT NEVER LOSE MAIL !!!

The gist of her speech was: sendmail is too huge, too complex and it just does too much. In short its a real bitch of a program to maintain (mega-applause from Daniel et al!). Its configuration file is really mean too. She reckoned that the sendmail of the future would go back to the good old "KISS" rule. It should really be simpler. More cheers.

**Remote UNIX filesystems by Douglas Kingston**

Douglas is from mcvax (I say this because nobody knows what the Centrum for Wiskunde en Informatica is.).

He gave a very good talk, describing remote file systems and some of their approaches (i.e. a super root, like Newcastle connection, or remotely mountable file systems, like SUN's NFS). He discussed some of the problems on a higher plane like, software transparency (ideally you don't want to recompile anything), user validation, where do you do a core-dump ? What about binaries etc ?? All interesting, non-trivial questions...

**EUUG General Meeting**

A lot of controversy erupted over the idea of exhibitions. These cost us (the attendees) a lot of money because the conferences have to be held in sleek and expensive conference centers because of space and power requirements; nearby hotels are sleek and expensive as well.

The general opinion was that we should have no more exhibitions. They are not really worth it. I zipped around the Copenhagen one in about 30 seconds. Nothing really interesting to see.

**Evening Entertainment**

After the conference, Jean (the Copenhagen Strangler) gave Achim and me a lift to our hotels. Jean's registration materials arrived late and DIS Congress Service booked her into the SAS Hotel (most expensive one in Copenhagen), where you can buy a Bang & Olufsen Stereo and have it charged to your room !!!

Both of them had a massive go at me, because of my feelings towards the MicroVAX2. I managed to get the point across that I really did like the MicroVAX2, but that I did not like the DEC sales staff, because some of them were not very well informed.

In the resulting 'discussion' Jean somewhat took her eyes off the road and managed to nearly stab my eye out with the bumper of a huge SAAB truck, while she was busy chasing two very scared cyclists off the road...

Using the lure of vegetarian food, I managed to rope in several charming ladies for dinner. So we all set off, a right likely bunch: Jean, Achim, me, Ulrike (Weng-Beckmann from Siemens), Ruth Moulton (from Whitechapel, who make workstations) plus a few people (male) from Siemens, whose exact names I cannot remember.

The food place was called "De Gro/ne Ko/kken"; for all you non-compo's this means "The Green Kitchen". The food was very good: Ulrike and I had an "eat-as-you-please" buffet (very good), everyone else had Moussaka. Achim again complained that the portions were too small, but this time he could not get more, because the kitchen closed soon after we were served.

After the desserts we found ourselves on the tiles again. Achim, Jean and I went to the Tivoli. Jean wanted to go on all these rides, but Achim and I politely refused. She accused us of being 'boring Germans'. This is worse than being called a 'boring old fart'. Achim and I sulked.

We then sat down in a Tivoli beer bar, where we were joined by Daniel and others... We entertained each other for a while, and we talked about Unix security. Jean immediately came up with this great idea of a newsletter issue on Unix security, and conned all of us into contributing. Got home at midnight, and worked on my Security paper until about 2 am.

**Day 2 (Thursday)**

**4.3 - An Overview by Kevin Dunlap**

Started the day with Kevin Dunlap, who is also in DEC's Berkeley Division (with Miriam), giving an overview of the speed enhancements.

- namei caching etc. have been improved by about 85%.
- Process management is now hashed and in three queues.
- Improved real time clock handling
- Exec is 28% faster now because it uses an improved algorithm, to copy binaries.
- Setjmp and Lngjmp have 13 % less overhad per call now.
- Lots of improvements on the libraries and utulities.
- csh was improved. From 20 sys calls per prompt to only 10.
- Inet daemon instead of one for every network.

I was really happy. Our 785 is already very swift, and even if only some of the claims work out as said, then it should be a really nifty little system. I think we will all love it....

**Document Preparation by Brian Kernighan**

Brian's second paper was about yatp (yet another troff preprocessor). The GRAP language is used to define graphs etc. Its actually a Pic preprocessor. It was implemented using awk. You can tell Brian works for AT&T: he never throws anything away... Troff was made 30% faster, due to diverse improvements. Nroff as gotten better ascii terminal tables.

I liked this talk, even if I don't use pic/troff/eqn.

**Font Design by Art Bigelow**

Art looks a bit like John-boy Walton. He wore a really stunning tie (flourescent colours etc) so that we will remember him. He gave one of the best talks at the conference.

He said that 1012 - 1013 letter forms (i.e. chars) get produced in the US every day. He classifies them into three categories:

- readable - the chars are so nice, they actually 'pull' your eye along the page, keep up your interest.
- legible - they are fairly easily readable, but since they are more optimised for clarity, you have to concentrate on the reading. Used mainly in technical stuff.
- decipherable - the worst case (He was obviously not thinking of my handwriting, because that isn't. You really have to strain yourself to get the information.

So he claims that the right font can actually make text more interesting. He had lots of very good slides. One good one was how the hieroglyphic symbols became transformed into handwriting and ultimately print.

He also said that the use of icons for computers is basically a step backward some 3000 years. I bet he won't say that on net.micro.mac !

Letters are a lot better for communication: we can still read texts in capital letters, typeset 2000 years ago. No other form of communication has this endurance.

He introduced the concept of spatial frequency, which is the frequency of alteration of black and white marks on the paper. The optimised frequency for humans is roughly that of a 10 - 12

point font. The old typesetters from ages ago already "knew" this - it iS the size of what most of us call "normal" print (from newspapers etc.). Font scaling is really 'tuning' text to optimise the humans perceptiveness. (My phrase - pardon me if I got it wrong, Art). Being constrained to a grid (i.e. like on a 5*7 matrix for terminals), ruins this 'fine tuning'.

As far as most terminals are concerned: he thinks that most screens are unbearable. He is right, too. He sells fonts for workstations etc. (including the Mac). Although, he is new to all this, he is successful and enthusiastic. He was very convincing. And he knows a *lot* about typography.

**Simula & C comparison by Georg Phillppott**

Georg is from the FUUG. Quote of the day: "Simula does it with class". He compared the two languages - first syntactically (boringly similar - or simular ?) and then semantically, specifically in the way that Simula has processes, objects, semaphores etc.

I learned (from his vintage slides) that 'Kildecode' means source code (obvious, isn't it...).

**C++ Tutorial by Bjarne Stroustrup**

This was a quick introduction to C++. I don't really like the idea of type checking in C - I like to write uncorrupted core-corrupting C, like on the day it was born. Bjarne (and AT&T) obviously feel otherwise.

C++ changes a lot of things. It also uses streams so printf("Hello world\n") (probably the most famous computer science quote) comes out as:

cout << "Hello World\n"

EEEEEmygads....what is the world coming to !

C++ is a preprocessor to C - hence the C-compiler does not get messed with (much). I prefer Objective C - could we have someone do a talk on that sometime (Committee, please ??)

**Unix 4.3 BOF chaired by Miriam Amos and Kevin Dunlap**

Whats a BOF - well, BOFFF is the sound a bird makes when it hits the ground involuntarily, losing all its feathers...hence BOF stands for Birds Of a Feather (or a "Special Interest Group").

This was a question and answer session re. 4.3 where poor Miri and Kevin tried vainly to answer everything at once. Some very interesting points arose however...

> They have this rdist program, which handles distributing sources and maintaing software across a network. This means that it will distribute updated source files immediately and try to recompile and re-install the software on different machines.
>
> Berkeley is working on its own distributed filesystem. It will not take an existing one like Newcastle or Sun's NFS to integrate into 4.4.
>
> 4.3 should be binary compatible with 4.2 (except getsockopt syscall).
>
> I/O on files (fread & fwrite) have been made 10 times faster.
>
> dbx will now also debug pascal programs. All known dbx bugs have been fixed.
>
> And lots more....yours for only $500 !!!

This BOF was a sort of name server BOF as well, because Berkeley are also giving us a name server in 4.3. A lot of discussion arose regarding its stability. Someone asked "So what if someone bombs mcvax ??". At this point I really burst out laughing because I could just see poor Jim McKie, sitting on the smouldering wreckage of his ex-VAX, clothed by nothing but the charred remains of an EUUG T-shirt.

**Evening Entertainment**

The conference dinner was in the Tivoli. The talloc() (table allocation) routine panicked because we arrived and asked for a contiguous block of nine places. So in the end we all scattered. I was most delighted to find myself sitting between Jean Wood and Chris Holmes (who works for Bigelow and Holmes (the font people)). If Jean has got strong hands, well Chris has got very steady hands: she is a font artist, this means she actually draws the letters in big (and they then get reduced and digitized/mutilated).

In her own words: 'One morning, I had just gotten up and had a cup of coffee. There was a bee sitting on the breakfast table and it had a little drop of dew or water on its back. I flicked off

the drop with one finger - and the bee never noticed. THATS when I knew, I had steady hands...'. Another amazing profession. The dinner was good too (despite Achims comments re. portion size again...he really is insatiable), the company was enchanting and it was over far too soon. Theo de Ridder held a very funny speech (as usual), Dave Tilbrook made rude comments (as usual) and later the crowds disbanded to watch the fireworks.

We (Achim, Jean, Bjarne (of C++ fame), Daniel, Miriam and Kevin) went sauntering around the Tivoli. Well, we wound up in this beer place, where we had a good laugh, telling each other about "my worst ever system crash". I had lots of good ones to tell. When the Tivoli closed at Midnight, we all went off to the Stro/get, Copenhagens pedestrians delight. Good fun until Jean tried to extricate her care from the hands of a vigilant car park attendant, who wanted to be in bed ages ago. Thankfully Bjarne was around to handle the Danish negotiations.

**Day 3 (Friday)**

The final day. Very many people were late...

**Screen based History for the Shell by Mike Burrows**

Mike is one of the lads from Cambridge. He showed us his shell the MSH (pronounced Mush). This is a nice idea, because he does away with the idea of having to recall the command name by event number (i.e. !21 ). Instead you type the first few chars of the command name and the shell substitutes the rest. You trigger substitution by use of a single key, like up-arrow for example.

For example, suppose you did several commands with a c at the beginning, like a cat, a cc and a cal. Then you type a 'c' and msh would bung in 'cal'. Type ^ and msh will put in 'cc'; type ^ again and msh would substitute 'cat'. It then goes around the same iterations for the filenames.

This sounds so nice that I will mail him and ask for a version. He claims that people really like it. It seems like the best thing out of Cambridge since the Ring.

**Copenhagen**

The next few talks were really about standardisation and internationalisation. I could not really bear the thought of yet another AT&T extravaganza, so I decided to skip off and have a look around Ko/benhavn.

It's a lovely little town and I got along very well with the locals. Its got such a nice atmosphere that I could surely live there. There are plenty of antique bookshops (Groan, said the VISA card) and I managed to pick up several mint first editions of some old A.E. van Vogt and Delaney. That made me very happy indeed.

Then I returned to the Bella Centre, to meet Heidi from Unix Europe. I had this unresolved licensing problem and she offered to help me out. She did too. She must do miracles for AT&T's cash flow: she tells you very honestly and nicely that your problems are over if you just pay $12,000 - and thats a special offer ! - and you just HAVE to believe her. I do too. At least Ma Bell does not sell herself cheap...

After all these lengthy and hair-raising discussions I returned to the conference proper.

**The X/OPEN Standard by Jacques Febvre**

He discussed this very same book, which is being published. In the following discussion the old AT&T vs. 4.2 controversy resurfaced. A show of hands showed, that the proportion of one or the other was about half. Then why take AT&T as standard ? The answer is obviously that Berkeley does not own half the dollars as well. I guess the only chance would be a 4.2 Unix standard issued by the Coca Cola Corp. Then at least we would have enough money to back us. Does that mean we could have Classic Unix ?

**Farewell**

Generally the feeling was positive, except regarding the the DIS Congress Service.

So the conference motto for this one was:

Your hotel has been DISorganised !

**Rumours**

It is rumoured that Dave Tilbrook bought something "very nice" for Helen Gibbons in the sex shop, next to the Hotel Triton.

**Memorable Quotes**

"Hello Robert, by an innings, by an innings, for goodness sake."
- Dave Tilbrook

"I think this system is like LEGO...do you have LEGO in Denmark ???"
- Tony Highwood

"I will shoot the next person, who asks a question"
- Helen Gibbons

"UNIX is like AIDS: its not very interesting unless you've got it or you're a doctor"
- Name and Address withheld by request.

"I do not like people corrupting themselves, be it Pascal or any other drug."
- Bjarne Stroustrup

"Is there such a thing as TOUGH FISH ?"
- overheard at the Cafeteria Nord

## Comments on the Survey taken at Copenhagen

*Jesper L. Lauritsen*
University of Copenhagen, Center of Applied Datalogy

The survey was prepared by Keld Jørn Simonsen and myself. Thanks to Edit Kjer-Pedersen for help in entering the data.

The questionnaire was prepared in a hurry (as usual, I guess). It was finished the day before the start of the conference, and some of the questions could probably have done with a bit more consideration. Several probably interesting questions had to be edited out so that we could keep the questionnaire on one page. Anyway it seems like anyone having something extra to say wrote a comment or a short story. When readers are evaluating the results it must be remembered that almost half of the questionnaires were collected by Wednesday, the second day of the conference. This was done to allow the presentation of some preliminary results on the last day of the conference.

We received 175 answers which is more than 50% of the registrants, so the survey should be quite representative of the conference. 105 of these were collected on Wednesday and the rest on Thursday and Friday. When possible the results are compared to the ones from Cambridge in 1984, which is the latest survey from an EUUG conference.

| | Copenhagen | | Cambridge | |
|---|---|---|---|---|
| Question | yes | no | yes | no |
| First UNIX conference | 47 | 127 | 62 | 86 |
| First EUUG conference | 76 | 98 | 85 | 61 |
| Attend future EUUG conferences | 157 | 16 | 142 | 1 |
| How heard of this conference: | | | | |
| came to prev. EUUG conf. | 78 | 97 | | |
| read in periodicals | 28 | 147 | | |
| others | 80 | 94 | | |

Note the relatively low number of new people. One important reason is of course that Denmark is a small country, but the late final programme and registration form are probably also to blame. Quite a number of the registrations were not received by the organizer until the last few days before the start of the conference*, indicating that many barely had time to organize funding, days off, transportation etc.

Perhaps it would be an idea to do some more advertising in periodicals; few saw information about the conference here. The common way to hear of the conferences seems to be by previous participation and the nets.

| Position | industry | research | user | total |
|---|---|---|---|---|
| manager | 17 | 20 | 5 | 42 |
| senior | 39 | 21 | 2 | 62 |
| other | 19 | 29 | 5 | 53 |
| total | 75 | 70 | 12 | 157 |

* The late registrations almost gave Keld a gastric ulcer. Being the main force behind the conference, he would probably have shot himself if the number of registrants were less than 200. Eventually we reached 344 participants.

| UNIX experience | |
|---|---|
| yrs | num |
| 0 | 6 |
| 1 | 21 |
| 2 | 31 |
| 3 | 25 |
| 4 | 20 |
| 5 | 25 |
| >5 | 44 |
| total | 172 |

| Age | |
|---|---|
| yrs | num |
| <21 | 2 |
| 21-25 | 33 |
| 26-30 | 47 |
| 31-35 | 37 |
| 36-40 | 37 |
| 41-50 | 11 |
| >50 | 1 |
| total | 168 |

| Nationality | |
|---|---|
| country | num |
| GB | 59 |
| north† | 40 |
| D | 18 |
| NL | 12 |
| USA | 11 |
| F | 10 |
| IRL | 5 |
| others‡ | 16 |
| total | 171 |

For the first time, I believe, we asked some questions about who the conference participants were. It is noteworthy that most have several years of UNIX experience. There were a few 'users'. I am not sure what a user is in this connection but I guess it is somebody doing neither research, development nor system administration. Both points should be remembered when future conferences are planned; introductary talks probably have little interest.

Great Britain are still far the largest national group but several national user groups are growing fast, and if you take the Nordic countries together they are catching up. There were a few participants from countries without a national user group.

| Question | num | average |
|---|---|---|
| How reasonable was the cost of: | | |
| conference fee | 145 | 2.2 |
| special tutorials | 81 | 2.1 |
| dinner | 122 | 2.0 |
| hotel | 117 | 2.1 |
| transport | 112 | 2.5 |
| How relevant was the topic: | | |
| communication | 143 | 2.8 |
| text processing | 122 | 2.5 |
| languages | 124 | 2.6 |
| standards | 113 | 2.5 |
| kernel | 120 | 2.6 |
| How useful was your attendance | 164 | 2.8 |
| How good was organization | 161 | 2.6 |
| How big is the need for spouse prog. | 118 | 1.2 |

For each of these questions you could mark with points 0 to 4, where 4 = very much, 3 = much, 2 = some, 1 = little, 0 = not at all. Apparently this was somewhat ambiguous for some, specially in the question about "the cost of", so you should not place too much faith in that result.

All the main topics of the technical sessions were rated as very relevant with communication as the winner. The intention was to get an idea of what people wanted to hear, not a rating of the quality of the actual talks. Unfortunately it seems some misunderstood that.

EUUG gets a good rating for organization. Many had complaints about the conference organizer, DIS Congress Service. From the many comments, many of them very lengthy and very angry, it is clear that mainly two things went wrong. Acknowledgment by mail was sent much too late or not at all, and worse, hotel reservations went totally wrong. Hotel rooms were not registered, or were registered and later cancelled, many people had no room when they came to Copenhagen. DIS can

† Denmark, Sweden, Norway and Finland

‡ Italy, Austria, Switzerland and Canada are the most important.

UNIX is a ... etc. etc. (boring isn't it?)

be somewhat excused; they had to cancel most of the hotel bookings because of the very late registrations and there was too little time to send acknowledgments. EUUG's late posting of the conference bookings also played some part in causing the late registrations. There were also a lot of complaints about the lunches, being too tiny and of too bad quality. Several had comments about the form of the conference and/or came with new suggestions. A sample:

- More industrial sessions, less academic topics.
- Workshops
- "Soapbox Sessions" 5 minutes to say anything.
- Conferences are becoming large + expensive. Difficult for academic sites to afford 2 per year.
- Think about special interest groups.
- Don't see what industrial sessions and exhibitions matters.
- Missing the Beer and Inode session.
- Less parallel sessions. I hate missing something.
- NO MORE EXHIBITIONS
- BOF sessions need to be formalized.
- Proceeding before conference.
- Conference + all hotels together.
- Free bus pass a great idea.
- Fresh fruit juice at the tea breaks

The Beer & Inode session was actually planned but nobody would sponsor it.

| | Copenhagen | | | Cambridge | | |
|---|---|---|---|---|---|---|
| **Question** | **num** | **yes*** | **av** | **num** | **yes** | **av** |
| **special tutorials** | | | | | | |
| **attendance** | **134** | **52** | **1.1** | | | |
| **importance** | **90** | | **2.1** | | | |
| **quality** | **55** | | **2.3** | | | |
| **free tutorials** | | | | | | |
| **attendance** | **135** | **70** | **1.6** | **148** | **25** | |
| **importance** | **104** | | **2.5** | | | |
| **quality** | **77** | | **2.7** | | | |
| **technical sessions** | | | | | | |
| **attendance** | **155** | **152** | **3.4** | | | |
| **importance** | **153** | | **3.4** | **148** | | **3.7** |
| **quality** | **144** | | **2.7** | **147** | | **3.0** |
| **industrial sessions** | | | | | | |
| **attendance** | **140** | **80** | **1.1** | | | |
| **importance** | **111** | | **2.2** | **142** | | **2.0** |
| **quality** | **78** | | **1.9** | | | |
| **exhibition** | | | | | | |
| **attendance** | **155** | **152** | **2.2** | **147** | **144** | |
| **importance** | **155** | | **2.1** | **147** | | **2.3** |
| **quality** | **147** | | **1.6** | | | |

Again answers were points 0 to 4. It is clear that the technical sessions are the most important part of the conference, but all parts have quite high figures on importance. Generally the quality was good, with the exception being the exhibition which was too small in most people's opinion. Some of the comments from the questionnaires:

- What exhibition??
- Instruction Set great.
- Special tutorials very good.
- Nothing new and exiting.

* This is number of answers with point greater than zero. I have included them to allow comparison with the figures from Cambridge.

- Great programme.
- General: GOOD.
- Pre-check quality of talks!
- Conference quality good.
- Technical programme very very good.
- Most of technical talks assumed no knowledge and were far to elementary.

My impression is that comments on questionnaires collected on Wednesday are very positive, and comments from Thursday and Friday are not quite so positive. My own personal opinion is in line with this. I think the programme was exciting, but several of the talks were disappointing.

To finish with, here are the suggestions for future speakers and topics. The list is fairly complete, I have only removed a few repetitions and some I could not read.

- graphics for UNIX
- text processing, languages, graphic workstations
- real working productions in UNIX
- new user interfaces
- Campus-Networking
- Gehani from Bell Labs: Concurrent C
- distributed UNIX
- fundamental issues of why UNIX is offered to the rest and what are going to happen in the future
- tutorials on networking protocols + kernel
- standard libraries
- C++
- windows, graphics
- more interaction between delegates
- internationalization of UNIX
- more talks on applications based on UNIX
- parallel processing - concurrent C
- Bob Brown: future of operation systems
- Distributed UNIX systems (invite Bruce Walker)
- Apollo & SUN
- Mr. Jose Furtny Amat of Kuwait Petrol International
- Interface issues between informix and standard packages
- Robert Lucky - AT&T Bell Labs
- A session on future directions
- some end users outside computer science area
- Dave Ressotto, John Candis of Bell Research
- Anyone with a sense of humor
- Real time applications under UNIX
- UNIX in banking - how are security problems tackled

I would like to thanks everybody who returned a questionnaire and specially the ones who took the trouble to write some comments. I have not been able to quote all comments here, but they will all be read and appreciated.

## News from the Netherlands

*Frances Brazier*

National UNIX Systems Users Group The Netherlands

In the past NLUUG Organized two conferences a year: one rather technically orientinted, non-commercial spring conference and one more elaborate conference with parallel sessions (technical and commercial) and an exhibition in the fall. The diversity in the topics and the level of expertise required to follow the sessions practically guarenteed that everyone could follow something. This, however, also implied a greater chance of (at least occasional) boredom. To try and avoid as much frustration and disappointment as possible, especially for new members, we've decided to change our policy. From now on, each conference will be based on a general theme and both technical and commercial sessions will be proposed for both conferences.

Our first theme-based conference seemingly has been met with the approval of our members. The theme of the last autumn conference was 'USER INTERFACES', a topic which we felt was broad enough to ensure a certain amount of variety in the presentations. There were indeed sessions on more general aspects of user interfaces and sessions on specific interfaces. An impression of the days content can be formed by reading the following brief descriptions of the technical presentations.

Gerrit van der Vaer (Vrije University) began the day with a talk on the cognitive and ergonomic aspects of UNIX in electronic offices (his paper on the subject will appear in a subsequent newsletter). Marten van Gelderen (**Xirion) went into a number of important principles involved in the design of user interfaces for office automation; principles which are not found in UNIX, nor in most UNIX applications now available (e.g. WYSIWYG -- What you see is what you get). The philosophy behind a specific environment, namely the B programming environment, and the potential of this integrated user environment were discussed by Steve Pemberton (CWI). Wim Heuber's (NIRHEF**) contribution, based on PISA (the Personal Interface to the Shared Accelerator control system), described a number of design considerations required for the visualization of models of this type. PCE PROLOG was designed to simplify the implementation of graphical user interfaces for research in Artificial Intelligence. Anjo Anjewierden (Universiteit van Amsterdam) described how he achieved this by combining an object oriented virtual machine and a standard version of PROLOG. Ron Tolido (Olivetti/AT&T) assessed the value of the Korn Shell (An interface for experienced UNIX programmers) as compared to the Bourne Shell and the C-shell, whereupon the discussion on the definition of user friendliness, was briefly pursued. The facilities in the Human Interface NIXDORF's implemented in its UNIX TARGON system was described, in detail by Hermann Johannes (NIXDORF). The user aspects and performance of Hyper Search, an information retrieval system for very large databases on dedicated hardware was discussed by Mike Forsyth (Gould/Memex** Information Engines).

The reactions we've had on this part of the last conference were mostly quite positive (not meaning of course that *every* session met all expectations.)

For the sake of completeness the titles of the commercial sessions now follow:

- UNIX in a mainframe environment, Amdahl
- DOMAIN/IX: System V and Berkeley 4.2 implementation in a distributed Apollo workstation environment, Jan-William de Vries, Apollo Computer.
- Application Program Generators on UNIX/XENIX, Holger A. Bok, CBI
- Why Gould is different from the rest of UNIX Hardware Manufacturers, Philip Martin, Gould S.E.L. Computer Systems
- High Performance UNIX Systems, A.W. van Schadwijk, Harris Corp.
- The Open Architecture of the SUN-workstation, Wiel van de Berg, Konig and Hartman

- UNIX SV and RISC Architecture, Bert de R**uiter, Nixdorf Corporation
- The UNIX-PC, Harry van Kneschot, Olivetti/AT&T
- UNIX from PC to Mainframe, H. Kramer, Sperry.

The commercial sessions were well attended and appreciated by the audience.

Our next conference will be on ..... UNIX and networks.

# EUUG Distribution List

*Frank Kuiper*

Centrum voor Wiskunde en Informatica
Kruislaan 413
NL-1098 SJ Amsterdam
tel: +31 20 5929333
telex: 12571 mactr nl
Internet: mcvax!frankk

*ABSTRACT*

This is a list of all the current (February 1986) EUUG distributions. It is a general description of the available tapes. Any changes of the contents of the tapes, as well as announcements of new tapes will be placed in eunet.general and the EUUG Newsletter.

**Ordering tapes.**

If you want to order any tape, please write to:

EUUG Distributions
c/o Frank Kuiper
Centrum voor Wiskunde en Informatica
Kruislaan 413
NL-1098 SJ Amsterdam
The Netherlands

Please note that you have to be an EUUG member (or a member of a local UUG). Please enclose a copy of your membership or contribution payment when ordering. Note also that for distributions D1, D2, D3 and D4 (and in some cases also for D8) a copy of your source license agreement with AT&T for at least UNIX version 7 should be enclosed with the request for that distribution.

Unless stated otherwise, all tapes come in tar format, 1600 bpi. 800 bpi is possible on request. Tapes and bill will be sent separately. Prices of the tapes are in Dutch guilders (HFl), and do not include VAT-taxes or postage.

**Distribution list**

EUUGD1 R6: UNIX V7 system, specially made for small DEC PDPs (11/23, 11/34, etc). The Kernel supports the UK terminal driver.
V7 source license minimum.
Price: HFl 120,-

EUUGD2: Early Pascal compiler of the Free University of Amsterdam.
V7 source license minimum.
Price: HFl 120,-

EUUGD3 R3: UNIX Networking software, 'news', and some auxiliary programs. If a copy of the source license agreement for at least UNIX version 7 is included, a fairly debugged version of UUCP with some enhancements and X.25 support will be added to the distribution tape.
Price: HFl 60,-
If requested, a second and third tape containing the major news-groups received on the Continent for the last several months is available.
Price: HFl 240,-

EUUGD4: UNIX software tools. The tools are sampled by the STUG (Software Tools Users Group). Most of the software is written in Ratfor, for which a Fortran support tool is included. If you want to have UNIX tools available on your non-UNIX system, this tape is available in different formats: DEC RSX, DEC VMS, UNIVAC, IBM MVS, UNIX tar, MIT (Machine Independant Tape) line feed format, and MIT card format (80 columns).
Price: HFl 150,-
The STUG group receives a $10 profit per tape from the EUUG automatically.

EUUGD5: A collection of benchmark programs made up by EUUG.
Price: HFl 60,-

EUUGD6: Usenix tape, containing contributions from various UNIX System Group Members. This is a license dependent distribution: V7, V32, SIII, V6 or no license disclosure available.
Price: HFl 140,-

EUUGD7: UNIX|STAT 5.0. A collection of about 25 data manipulation and analysis programs written in C by Gery Perlman.
Price: HFl 60,-

# EUROPE'S ONLY UNIX WEEKLY

# UNIGRAM/X

**THE WEEKLY NEWSPAPER FOR PEOPLE WHO NEED TO KNOW ALL THE LATEST DEVELOPMENTS IN THE WORLD OF UNIX.**

"Unigram/X has proved to be a valuable source of information covering all aspects of the Unix world. Unigram provides the most up to date information in a fast moving market.

**Tony Lazzerini – Product Marketing Manager, NCR.**

"With up to date and reliable information Unigram/X is very much at the leading edge of the Unix market place – very thorough especially in US coverage."

**Andrew Davis – Business Manager Software Products, Data Logic.**

"We use Unigram/X as a cue for the hot topics slot at our group meetings. We like our information right up to date with no sales pitch."

**John Carolan – Chairperson, Irish Unix Systems User Group.**

"Unigram/X provides a concise report on current activities in the Unix industry week by week."

**Barry Forrest – Marketing Manager, Altos Computer Systems.**

"Unigram/X ar enligt utgivaren världens nyhetsblad for Unix som kommer ut varje vecka."

**Peter Tegelberg – Computer Sweden**

"Having now read Unigram/X since it was first launched I have found it useful reading for anyone involved in the world of Unix; it is concise and keeps one informed of all the latest developments on a weekly basis."

**David Thornley – General Manager, Pyramid Technology.**

"Professionally put together, easy to read, Unigram/X gives in-depth product overviews. It is up to date in a fast moving environment. This is no mean feat."

**David J. Bethel – European Sales / Marketing Manager, Zilog**

**Take advantage of our free trial offer for one month – Call Richard Brighouse on 01-439 1632 or for further information write to**

APT DATA SERVICES
4TH FLOOR 12 SUTTON ROW LONDON W1V 5FH.

UNIX is a Trade Mark of AT&T Bell Laboratories

# USENIX SUMMER CONFERENCE & EXHIBITION

Atlanta, Georgia, June 9-13 1986

# Pre-Announcement & Call for Papers

The USENIX Association will hold its Summer 1986 USENIX Conference and Vendor Exhibition in Atlanta, Georgia, June 9–13, 1986 at the Atlanta Hilton and Towers, Courtland & Harris Streets N.E., Atlanta, GA 30043, USA. The Host for the event is Medical Systems Development Corporation. This announcement provides early information about the date of events as well as persons to contact for further information. A pre-registration packet containing detailed Conference information and hotel reservation forms will be mailed in early April.

## Technical Sessions

Wednesday to Friday — June 11–13, 1986

Emphasis will be on highly technical sessions of an interactive nature. Panels and semi-formal workshops will be presented, as well as formal papers.

## Tutorials

Monday and Tuesday — June 9 & 10, 1986

All tutorials at the USENIX 1986 Winter Technical Conference were completely sold out. In order to address the constantly increasing demand for these high quality tutorials, the USENIX 1986 Summer Tutorials will be offered on Monday and Tuesday, June 9 and 10. The most popular tutorials (4.X and System V Internals, Networking, Device Drivers, etc.) will be offered on BOTH days. In addition, tutorials on System Administration, Language Construction Tools, Generating Portable Code, Managing Networks and other topics will be offered.

These well respected tutorials are taught by leading UNIX experts. They focus on essential areas of UNIX technology and provide in depth coverage on the subjects.

## Vendor Exhibition

Tuesday to Thursday — June 10–12, 1986

The primary intent of the vendor exhibition is to provide vendors the opportunity to display advanced technology relevant to the UNIX technical community. Exhibit space is limited. If you wish to receive details on exhibiting at the Atlanta Vendor Exhibition, please contact: **John Donnelly, USENIX Association Exhibit Office, Oak Bay Bldg., 4750 Table Mesa Drive, Boulder, CO 80303, Telephone 303-499-2600.**

The USENIX Association is an organization of AT&T licensees, sub-licensees, and other persons formed for the purpose of exchanging information and ideas about UNIX and UNIX-like operating systems and the C programming language. It is a non-profit corporation incorporated under the laws of the State of Deleware.

If you did not receive this announcement directly and wish to be on the mailing list for receipt of the pre-registration packet please contact:

**USENIX Conference Office**
**P.O. Box 385**
**Sunset Beach, CA 90742**
**Telephone 213-592-3243/592-1381**

## Call for Papers

The Summer 1986 USENIX Conference urgently solicits papers and panel sessions on any technical aspect of the UNIX Operating System including related areas such as compilers and languages, networks, distributed computation, and software technology. Further, the Committee especially solicits contributions in the following special areas:

**The Audio-Visual Unix; Computer Typography; User Interface Technology; Electronic Mail Systems; Operating Systems Design; Remote Filesystems; The Philosophy and Theology of UNIX.**

Abstracts should be at least 1000 words or 1–2 pages and convey the sense of the final paper. The author should include his name, address, telephone numbers, and network addresses attached to the submission. The submissions will be acknowledged as received. Network addresses are particularly useful in the case of paper submissions.

The Program Committee requests that abstracts be sent via electronic mail directly to:

**mo@seismo.arpa or seismo!mo**

Paper submissions may be sent to the following U.S. mail address:

**Mike O'Dell**
**13110 Memory Lane**
**Fairfax, VA 22033**
**Attn: USENIX Program Committee**

In case of delay in acknowledging electronic mail, or for additional information, contact Mike O'Dell, Program Chairman, at (703) 471-0030 or (703) 378-8574.

EUUGD8: A collection of usefull software, based on the so called Copenhagen tape (EUUG Unix conference Autumn 1985). It consists of the following:

cph85dist: the Copenhagen 85 distribution (Including all the Langston binaries).

| | |
|---|---|
| astro | - Misc. programs about astronomical events. |
| compress | - A program to compress large files. |
| hack | - A game. (Version 1.0.3) |
| kermit | - A file transfer program. (Version 4.2) |
| langston | - A large number of games from Peter Langston. |
| macintosh | - Misc. programs for Unix to Macintosh comm. |
| magtape | - Manipulation of ANSI tapes. |
| mandelbrot | - Image generation program. |
| mh.5 | - A message handling system. |
| new_curses | - Bug fixes to curses for making wm run. |
| patch | - A program to automaticly install bugfixes. |
| ptxnews | - A program to generate an index of news subject. |
| rand | - An editor. |
| rman | - A remote manual server system. |
| rn | - A newsreader program. (Version 4.3) |
| search | - Another game. (4.2BSD only) |
| shar | - Shell archiver. |
| stage2 | - A compiler. |
| stars | - A database on bright stars in our galaxy. |
| stat | - A statistical package from Gary Pearlman. |
| strings | - A portable version of the libc strings rutines. |
| vttest | - Test of VT-100 emulations. |
| webster2 | - Websters 2nd dictionary. |
| wirewrap | - A component generator for wirewrap constructions. |
| wm | - A window manager. (4.2BSD only) |
| xlisp | - A lisp interpreter. |

Emacs: latest public domain version of this editor (currently 16.60.10).

Netnews: the public domain part of EUUGD3.

(Yacc-pcc: By: J A Dain, Dept of Computer Science, University of Warwick
For Yacc-pcc you need at least AT&T V7 source licence. This part is only included on request)

Price: HFl 120,-

## EUUG Tape Distributions Order Form

**If you want to order any tape, please write to:**

EUUG Distributions
c/o Frank Kuiper
Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

**For information only:**

Tel: +31 20 5924056 (or: +31 20 5929333)
Telex: 12571 mactr nl
Internet: euug-tapes@mcvax (or: frankk@mcvax)

Please note that you have to be an EUUG member (or a member of a national UUG). Please enclose a copy of your membership form when ordering. Note also that for distributions D1, D2, D3 and D4 a copy of your source license agreement with AT&T for at least UNIX version 7 should be enclosed.

All tapes come in tar format, 1600 bpi. 800 bpi is possible on request. Tapes and bill will be sent separately.

Name: .....

Address: .....

.....

.....

.....

I would like to order the following:

.....

.....

.....

.....

EUUG (or national UUG) membership form enclosed? Yes / No
Copy of AT&T source license enclosed? Yes / No

The Secretary
**European UNIX† systems User Group**
Owles Hall
Buntingford
Herts. SG9 9PL.
Tel: Royston (0763) 73039