EurOpen

News Letter

Number 2

Summer 1991

Volume 12

EurOpen Forum For Open Systems

Volume 11, Number 2, Summer 1991

ISSN 1011-4211

Editorial Team

Frances Brazier
Publication Executive

Department of Computer Science
Vrije Universitei
de Boelelaan 1081
1081 HV Amsterdam
The Netherlands

Telephone +31 20 5485588
Facsimile +31 20 6427705

Email frances@cs.vu.nl

Alain Williams
Editor

Parliament Hill Computers Ltd
7 Prospect Street
Caversham
Berkshire RG4 8JB
United Kingdom

Telephone +44 734 461232
Facsimile +44 734 474194

Email addw@phcomp.co.uk

Gina Baikenycz
Typesetter

The Instruction Set
City House
190 City Road
London EC1V 2QH
United Kingdom

Telephone +44 71 253 5121

Email gina@inset.co.uk

Joe Di Mascio
Printer

Rank Xerox Business Services Ltd
89-91 Hampstead Road
London
NW1 3EL
United Kingdom

Telephone +44 71 388 5607
Facsimile +44 71 388 8841

Publisher

EurOpen
Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255

Email europen@EU.net

Contents

Editorial

Alain D D Williams
Parliament Hill Computers Ltd
Caversham
United Kingdom

Email addw@phcomp.co.uk

Tromsø Conference

By the time that you read this the Tromsø conference will be an exciting event in the memory of those who attended. The standard of papers and tutorials will have raised again expectations.

As I write (just before the conference) it is hoped that some new groups will join EurOpen - more details in the next issue.

Publications

If you did not attend you can obtain a copy of the proceedings from Owles Hall. Don't leave a gap in your collection: see page 28.

You can now order any National Group (NALUUG in EurOpen speak) publication from Owles Hall, as well as those produced by USENIX and Uniforum.

Standards

Henk Hesselink has been representing EurOpen at standards meetings to ensure that the European viewpoint and needs are not forgotten. As you know Dominic Dunlop has long been doing this. He will now co-write the Standards Column with Dominic; see page 42

Do you have a point of view on standards that is not being presented? Do not complain to your friends contact Henk or Domo.

Technology Reviews

There will be a special Technology Review section in the next newsletter. This will be the first of a series of such product review sections. Each newsletter will focus on a particular area.

The purpose of this section is twofold:

- Help real end users make important decisions, and in a manner that they can understand.
- Provide a showcase for manufacturers.

This is in accordance with EurOpen's objective to promote the knowledge, use, and application of Open Systems.

The first such focus will be on backup (archive) products - one of the most important "non application" tasks that we all (I hope *all*) do.

Vendors of such products are invited to write about their offering. The article should contain the following information: -What the product is called.

- Major features.
- How does it improve on what is offered by straight UNIX.
- What are the Unique Selling Points.
- Why it is better than the competition in general.
- How long has the product been available, how well has it sold?
- What support is available?
- What documentation is available?
- Performance, how fast is it?
- Are different (human) languages supported in the user interface, are configurations for European languages available 'off the peg'?
- Availability - ie what systems, what counties? Up to 2 names and addresses may be given.
- Pricing information. If it is modular give module prices (e.g. the network may be extra). For base price comparisons please quote the price for an 8 user 386 box.

Products that are not available in Europe or only available in one European country will be rejected.

Advertising space is available and may be booked in the same issue.

All those interested in participating should contact me by 25th of June, final copy will be expected by 11th July.

The next topic (for ie the Winter issue) will be on Desk Top Publishing.

Security Advice for UNIX Machines on a TCP/IP Network

Version 2.0 (April 91)

Jean-Luc Archimbaud
IMAG
Grenoble
France

Email jla@imag.fr

I live in Grenoble (France - ski). I work at the Network Coordination Team of the CNRS (Centre National de la Recherche Scientifique), mainly about security and TCP/IP. With little (and even no) money, people, time and mind; be pragmatic.

Jean-Luc Archimbaud CNRS/UREC who's mission is "computer security" at CNRS

Without becoming paranoid, users and managers of computer systems must take steps to limit exposure to risk of their machines and data. System security is the responsibility of the system administrator. This document is thus to help UNIX work station administrators in this task. It's aim is to provide basic advice to make a UNIX machine, connected to a TCP/IP network, less vulnerable to pirates. This is a compilation of different publications, but not just a recipe book to the detriment of the analysis and remedies of failings. UNIX is an operating system created by programmers, for programmers, in a research laboratory. Security was thus not a dominant force in it's conception. But, for more than it's origins, it vulnerability comes mainly from:

- It's popularity: it is the best known operating system. Many pirates have made use and will make use of bugs in a system for which the source is easily obtainable.
- vendors' attitudes: they deliver a system which is totally open.

Advice For Administrators

Having responsibility for the security of your machine, you should operate the necessary procedures and put in place the tools provided by the UNIX designers it protect it. This is what the following points review. But this technique is a long way short of protecting the integrity of your system. Alertness and vigilance are your two principal defences. The military have a phrase for it "Security is 20% technique and 80% good sense". Before all else, do not forget this golden rule: When you have a piracy problem, immediately notify your superiors.

Verifications from the moment the machine is used

After your system is installed onto disk (as when delivered or when upgraded), make a backup and keep it safe. This can provide a reference.

/etc/passwd and /etc/group

Of these files:

- Ensure that they are owned by root with access permissions of 444 or 644; and that all users and all groups are set up with a password (the second field of each entry must not be left empty).
- If you don't use "Yellow Pages", remove the line which starts with a "+" (if it exists).
- If you use them, ensure that the lines "+ : :0 : :0 : : :" and "+:" are removed from these files on server machines. They must only be present on client machines.

In /etc/passwd:

- Only root should have a "user id" (UID) of 0.
- Change the passwords that you used to install the system, likewise those that arrived with the system.
- Remove the service users: guest, visitor, tutor, demo,...

Sensitive Files

- Under /dev :

The super user "root" must be the owner of all these files, except those files which relate to terminals actually logged in. The file kmem, mem and the disk partitions (with the names sd*, rxy*, ... depending on the version) must have access mode 0 for "other".

- If your system has the option of a "secure terminal", put it into effect: remove the word "secure" from each line of the file /etc/ttytab (or /etc/ttys). The absence of this attribute will proscribe the direct logging in as root. Access to the super user will thus only be available though the su command. Specify as well which users can perform su though the declaration of a group wheel in /etc/group.
- Remove "." (the current directory) from the search path for the user root (the variable path is initialised in /.cshrc and PATH in /.profile).
- Destroy the file /.rhosts if it exists.
- Remove the file /etc/hosts.equiv if you don't need it. If you do, verify that this file contains only one line only consisting of the single character : "+".
- If you don't use NFS, or of you don't wish to make a part of your file tree accessible (exported), destroy the file /etc/exports. If you do want this, ensure that the name of each directory that you wish to "export" is followed by the names of the work station which will have the right to access it. Otherwise any work station will be able to mount this directory. The syntax depends on the version of UNIX (use : man exports).
- Remove, if they are present, the alias "decode" and"uudecode" in the file aliases in /etc or /usr/lib.
- Remove the access right "w" to "other" on aliases, aliases.dir and aliases.pag in /etc or /usr/lib.
- In the file sendmail.cf (in /etc or /usr/lib), ensure that the variable "W" (wizard password) has the value asterisk "*". To be explicit, in this file the line starting by "OW" (the letter "O" followed by the letter "W") must be in the form : "OW*".
- Remove all access rights to "other" from the file /usr/var/spool/cron/crontabs/root. Ensure that root is the only user with the access right "w" on all procedures started though this file.

inetd.conf

The daemon inetd, always active, provides services such as telnet, rlogin, ftp, ... These services which execute on your machine are available to machines on the network.. Only the services declared in the file inetd.conf (in /etc or /usr/lib) or /etc/servers will respond to remote machines. Do not hesitate to tidy up this file by adding a "#" at the start of the line for the services that you do not use. This could be:

- fingerd (or in.fingerd) : very useful to remote users (to obtain a user's phone number for example); it also allows a pirate, from any machine on the network, to find out who is connected with your machine.
- tftpd (or in.tftpd) : a simplified version of ftp, it is used to boot the system of a diskless work station (an X terminal for instance) through the network. Versions earlier than the end of 1988 have a very dangerous bug. Recent and secured versions can be recognised by the presence of a "-s" in the command line which starts the daemon.
- To be more restrictive, is your work station is never accessed from another machine (in the incoming sense), you can remove:
- telnetd (or in.telnetd) : prevents interactive access to your machine though telnet
- ftpd (or in.ftpd) : prevents file transfer by ftp initiated from a remote machine
- rlogind (or in.rlogind) : prevents rlogin entry
- rshd (or in.rshd) : prevents the execution of commands on your machine initiated from a remote machine
- rexecd (or in.rexecd) : on which rshd depends for functions
- rpc.* : responds to RPC requests (Yellow Pages, NFS, ...) coming from remote machines.

These restrictions will not affect outgoing requests. You will always from your work station be able to use telnet, ftp, ...

/etc/rc*

- The scripts /etc/rc* initiate network daemons such as inetd. The scripts delivered with the system have an annoying tendency to activate useless daemons. It is preferable to not start them (by inserting a "#" before in an ad hoc manner in these scripts) if you don't need them. These can be: rwhod (or in.rwhod) : broadcasts regularly to all machines on the network information concerning those users logged in on your work station.
- sendmail : receives mail from other machines on the network (it is the remote debug option present is some versions of sendmail which is dangerous). Note that this daemon is necessary for inter-machine mail.
- routed (or in.routed) : maintains and distributes IP routing tables in an automatic and uncontrollable manner.
- nfsd : needed to be a NFS server.
- biod : needed to be a NFS client.

Access TCP/IP

Calls on TCP/IP

To access a service offered by a machine on a TCP/IP network (interactive access, file transfer, NFS, rwho, lpd, ...) that service must be open : the daemon must be running and you must have the necessary authorisations (password, ...). But, before this can happen, the machines need to support the TCP/IP protocols. This is what one can call "access TCP/IP". In limiting the opportunities for TCP/IP access to your machine, you likewise minimise the risks of piracy. With TCP/IP access is always symmetrical. If you can access a machine X, a user on this machine X can access your system. Conversely, if you cannot access X, X cannot reach you. This if you cannot reach a network, you can be certain that pirates on machines on this network cannot touch you. However if you have access to machines world wide ...

IP Routing

The IP routers installed on your machine (seen though the command netstat -r) determine TCP/IP access to you machine. Here are a few words of advice:

- Only install the routing that you need for access.
- Unless you understand RIP, remove the routed (or in.routed) daemon started from the /etc/rc* scripts. Use instead the manual routing using the route commands.
- Consider carefully the consequences of permitting route by default (command route add default ...) : every TCP/IP machine in the world could then try to get in to your system.
- In an even more restrictive manner, set routing privileges by individual machine and not by subnet. Thus the command : route add 129.89.32.2 ... will allow you to communicate with this particular machine, without opening access to all machines on subnet 129.89.

External Connections

If your laboratory makes use of an Ethernet TCP/IP network linked onto a wider network (of the campus or of the region) which is itself connected to a national network ... it can be wise to install equipment with two ethernet connectors between your internal network and the wider network. It will be an IP gateway between your laboratory and the outside world. This could be between dedicated equipment or a plain work station. By limiting IP routing (by removing, for example, the routed daemon on the gateway and adding each one with the command route add ...), the internal work stations will never become compromised. It will be sufficient for you to monitor gateway access; and not to install any dangerous daemon or utility, or confidential data. But be careful, a user who is able to access the gateway can move onto internal machines.

Regulate Housekeeping

Backups

It is obvious that a good policy of regular backups is imperative for the safety of you system. If must be possible to return to a safe and sound previous state. Use dump not tar or cpio for these operations.

Passwords

In /etc/passwd :

- Remove users who do not use your machine any more. Also destroy all files belonging to these users. The following command will list all files which belong to no one (in fact those files for which the UID of the owner is no longer in /etc/passwd). It can be useful :

```
find / -nouser -o -nogroup -print
```

- Ensure that everyone has a password and that two users do not share the same UID.
- Ensure that the character "+" has not disappeared from the line "+ : : 0 : : 0: : :" if it exists. Remind your users (via mail or motd) to regularly change their password. If your system supports password aging, switch it on.

Root

Review the history or root logins (and su). You can inspect the audit trails in files such as messages* and sulog in /usr/adm.

Checks on certain files

Ensure that :

- There are no strange files whose name start with a "." in /tmp or /usr/tmp.
- The contents of /etc/hosts.equiv and of /etc/exports are correct.
- The executables of the su, login and telnet programs are the original ones.
- The scripts started by cron for root are safe (in /usr/var/spool/cron/crontabs/root or ...)
- There are not too many files with a "w" access for "other" with the command :

```
find / -type f -perm -2 -exec ls -al {} \;
```

Sushi

A Sushi (Super User Shell Interactive) allows a user to work in the shell with all the privileges of root. It is the shell program with the Set User Id bit (SUID) set. To guard against a Sushi regularly check that the files which belong to root with SUID set are only the utilities. Such files should not be found in user file systems. The command :

```
find / -user root -perm -4000 -exec ls -al
       {} \;
```

will list this type of file.

General Advice

Do not forget that a work station can be restarted in stand alone mode and that in this state the user has all privileges. You must allow for this possibility by, for example, controlling physical access to machines.

Work Habits

- The root password is the key which opens all the doors : choose it with care, change it regularly, do not give it to anyone.
- Logout every time that you leave your work station.
- If you use X-Windows, always keep a "console" window on you terminal (command : xterm -C)
- Only login as root for the purpose of system administration. Use another login when you do not need the privileges.
- Never let someone else work as root, even for a few minutes.
- Type /bin/su instead of su to become root.
- Add the command who to your .login or .profile file. This will allow you to see users who should not be logged in to your machine.
- Use a special account with a minimum of privileges when running demonstrations, tests, ... where others are present.

/etc/hosts.equiv

With hosts.equiv you leave security controls entirely up to the machines mentioned in this file. You are thus totally trusting other administrators. This is very dangerous. So, except for particular needs, eliminate the use of the file hosts.equiv (remove it).

.rhosts

Control the use of .rhosts files by your users. These allow entry to your machine without the need for a local password. To inspect the contents of these files you can use the command :

```
find / -name .rhosts -print -exec cat {} \;
```

Groups

- If users wish to share their files, do not let them open up access rights "rw" to "other" on these files. The good use of groups (cf /etc/group, /etc/passwd, umask 007, chown, chmod, newgrp, ...) will permit this problem of shared access.
- Each account declared on your machine must correspond to one and one only person who is clearly identified. You must know the background of each user, with their scope of activities on your machine.

Miscellaneous

- Make your users aware of security issues. In the first instance widely circulate the chapter "Advice to Administrators".

Advice To Administrators

One official and basic rule : if you discover piracy, an attempt at piracy or a suspect state of affairs : Immediately alert the machine administrator and your management hierarchy.

Passwords

- Choose it carefully

- Change it regularly.
- Do not reuse a password that you have already used.
- Change it before going on holiday.
- Do not write it down or tell it to someone.
- Do not choose:
- Your name, forename, or that of someone close to you.
- Important personal information : telephone number, ...
- A common name or proper name found in a dictionary.
- A variation on the 3 preceding groups (inversion initials, ...)
- A good password should be made up of:
- At least 6 characters
- A mixture of upper and lower case letter, digits, and punctuation characters.
- Example : it's1KO

When trying to find a password, pirates often do not try all combinations of characters. He gets to know the profile of the typical user. He tries, with others, all information pertaining to a user (name, ... with variations) and the words in the dictionary.

umask

The command umask allows the creation of a mask which forces the initial access modes when a new file is created. Often this mask is initialised to 002 or 022 for all users on a system. This corresponds to the access rights 775 or 755. This, by default, all new files are readable by everyone. By adding a umask command in your .login or .profile, you can change the default access. This umask 077 will give you maximum protection on new files.

Work as a Group

If you must share the same environment with several collaborators, ask the administrator to register all the members of your team in one group. To keep the shared files in a location "Common" :

- Create a sub directory which will contain the shared objects : mkdir Common
- Give "rws" access to the group for this location : chmod g+rws Common
- To prevent some access conflicts : chmod +t Common
- Reinsert into your .login or .profile : umask 007

Miscellaneous

In the search path (path or PATH), specify the current directory "." at the end of the list and not at the start. When you access your machine, carefully check the date and time of your last login, this is generally displayed in the welcome banner. Use wisely .rhosts and regularly check its contents.

Documents

Documents which inspired this "recipe book" :

- Improving the security of your UNIX system, *David A. Curry*
- Security features guide, *Sun Microsystems OS 4.0.3*
- The Internet Worm Program : An Analysis, *Eugene H. Spafford*
- Unix System Administration (security chapter), *David Fiedler & Bruce H.Hunter*
- Internet Intruder Warning, *CERT*

European Vendors are Loosing Ground in the UNIX Battle

Per Andersen
IDC Scandinavia A/S
Copenhagen
Denmark

For 7 years, Per Andersen worked as a consultant in the IT industry responsible for the introduction and use of information technology in large organisations. This included being adviser to the Danish government, establish procurement agreements on Cs and departmental multiuser UNIX systems. Per is the author of a widely used textbook on the PC marketplace.

Per Andersen's background includes manaing a consultancy department at UNI-C, a Danish supercomputing centre, responsible of advisory and training projects in the fields of minicomputers, PCs and local area networks.

Per Andersen is the manager of IDC's Pan-European UNIX service programme publishing market research reports to the European IT vendors and users.

Primarily driven by growth in UNIX PCs and workstations, the total European UNIX systems market grew by 25% in 1990, according to research from the market research company IDC.

Even though the use of desktop PCs running UNIX did not meet expectations in 1990, the use of UNIX PCs as small multiuser systems experienced higher growth in Europe than in the States and contributed significantly to the growth in the UNIX PC segment. Workstations also had a healthy growth rate at 38%.

Multiuser UNIX systems did not have high growth rates which was expected considering the overall slow down in the multiuser systems area. The value of small scale systems increased by 21%, while medium scale systems only grew by 10% in value.

While multiuser UNIX systems did not perform well in its own right, the increase looks OK when compared to the growth in the overall multiuser systems marketplace, which declined las year. Accordingly, the UNIX penetration of the medium scale systems still continues to climb.

Winners are Emerging!

Among the winners of the 1990 UNIX marketplace were IBM and Sun increasing their market shares in the workstations and the small scale marketplaces. hewlett-Packard, while loosing market share in workstations, had very good midrange sales.

The now merged company Siemens Nixdorf had a rough year on the European marketplace, when revenues in the small scale as ell as the medium scale dropped. It seems that the Siemens Nixdorf slogan of "Synergy at Work" did not work too well and especially the Nixdorf part of the business did not meet expectations. The result was, that Siemens Nixdorf dropped from being no. 1 UNIX vendors in Europe to a more modest third place.

Siemens Nixdorf, like the NCR's of this world, are going for the Intel-platform at the low end and during 1990 as well as this year they are replacing the National Semiconductor models in the MX 300 and 500 series with Intel-based models. By riding the Intel wave, which rapidly increases its presence in the low end of the UNIX marketplace, Siemens-Nixdorf hopes to gain momentum from a large base of Intel/UNIX software. In the medium range, the Targon series from Nixdorf will be continued for some time. The low end model 31 based on Motorola 68K will gradually be phased out and new 68040 model is expected to be announced this year.

Similarly, Olivetti did not grow as much as the total marketplace in 1990, but the company managed to keep its fourth place in the ranking. Olivetti expects its 1990 total turnover to be broadly unchanged, but profit is expected to drop sharply.

Bull and ICL Did a Better Job

the growth of Bull was slightly above European average, and accordingly the company increased its market share. Some of the growth, however, was achieved by the acquisition of Zenith and their shipments of PCs running UNIX.

In the multiuser arena, the Bull DPX product lines today use 3 different processor platforms: Intel, Motorola and MIPS. This is certainly too many and it is now clear that the Motorola line will go.

By using Intel processors in the low end bull takes advantage of the rapidly growing marketplace for Intel based UNIX systems and at the same time, by using MIPS processors in the high end Bull is able to provide available technology and is well positioned for the growing RISC marketplace.

However, the drop-out of the Motorola based systems, is leaving a big gap between the low end Intel-based systems and high end MIPS based systems which has to be filled. The Intel based models will have to be expanded upwards by using 486 and eventually 586 processors and multi-processing. The high end will have to be expanded downwards buy using MIPS processors.

Of the European vendors, ICL was the company experiencing the highest growth rate in 1990. From being a rather invisible UNIX vendor in 1989, ICL by announcing the DRS 6000 system in January were able to ship around 1,000 of these new UNIX systems and as a consequence they jumped to the 10th ranking of UNIX suppliers in Europe.

The success of ICL, however, were not enough to ensure a combined success of the European vendors in 1990, and the question is whether they will be able to keep up pace with the US and Japanese vendors in the far more open Europe to come.

"It is doubtful that the European vendors will be able to continue their highly dominating position in the home markets. The emerging border-free and open Europe will make any local vendor vulnerable, and especially those vendors having a dominance in a single market will find themselves under fierce competition from worldwide vendors such as IBM and HP", explains Per Andersen, who is heading IDC's European UNIX research.

The new IDC report on the European UNIX marketplace will be available from any of IDC's local offices or from:

IDC European UNIX Expertise Centre Copenhagen Denmark

Telephone +45 31 18 63 44 Facsimile +45 31 18 44 48

Rename("open", "swinging_to_and_fro");

David Tilbrook
Sietec Open Systems Division

Email dt@snitor.uucp

Abstract

The push for Open Systems is on. The objectives of Open Systems are to achieve independence of supplier and/or manufacture, portability of software, and environmental differences that are transparent to the user. Part of this push must involve the evolution of a standard software foundation, and in some ways it appears that UNIX is evolving as that foundation. However, the successful achievement of the objectives of the Open Systems push depends largely on the reliability and consistency of that software foundation.

This paper looks at one subroutine rename(2), that has been deemed to be part of the Posix standard. This paper relates the author's experience with this subroutine across some dozen different platforms and configurations, and draws some conclusions regarding the chances of fulfilling the Open Systems marketing promises.

Introduction

The rename function was not part of the early versions of UNIX. Its basic functionality was relatively easy to implement, as discussed later, using the link(2) and unlink(2) system calls. One could live without a kernel implementation of rename, and, indeed, on some systems, one still does. However, it was implemented as part of the 4.2bsd fast file system extensions and has been adopted as part of the Posix standard, so examining this subroutine to attempt to prognosticate the future of the Open Systems effort is not gratuitous.

The following is an extract of the UPM manual section for rename. Other variations exist; however, barring the copious caveats concerning circular directory graphs, and the various error conditions, the basic essence of rename is as described.

NAME

rename - change the name of a file

SYNOPSIS

```
rename(from, to)
char *from, *to;
```

DESCRIPTION

Rename causes the link named from to be renamed as to. If to exists, then it is first removed. Both from and to must be of the same type (i.e., both directories or both non-directories), and must reside on the same file system.

Rename guarantees that an instance of to will always exist, even if the system should crash in the middle of the operation.

RETURN VALUE

A 0 value is returned if the operation succeeds, otherwise rename returns -1 and the global variable errno indicates the reason for the failure.

Now, how could anyone get that wrong? Actually getting it right is difficult (see Epilogue #1). There are all sorts of potential errors, some of which are covered in the parts of the manual section that I did not include. Certain aspects, such as renaming directories, should be approached with great apprehension. In fact I do. rename is not provided on a variety of systems on which I have to run my software. On those systems I have to provide an alternative mechanism to support the basic functionality of changing the name of a file. Furthermore, there are many situations where using rename arbitrarily can give rise to errors from which one then has to recover (i.e., cross device links or cross directory renames not supported on some file systems). Therefore, I limit my use of rename to files within the current directory. An examination of a variety of rename manual pages seems to indicate this subset of rename's supposed capabilities avoids many of the potential errors. Furthermore, by limiting myself to that subset, the rename functionality can be almost duplicated using the following code:

```
#include <errno.h>

    int rename(from, to)
        char * from;
        char * to;
{
       if (unlink(to) == -1 && errno != ENOENT)
          return -1;
        /* If we crash here, to is gone!
         * If from cannot be linked we lose as
         * well! */
        if (link(from, to) == -1)
          return -1;
        return unlink(from);
}
```

The reader will appreciate that this is obviously less efficient than the true rename in that it uses three system calls vs. one, and has the added disadvantage that to is unlinked before we are sure that from exists and is the same type as to. However, more importantly, the guarantee that to always exists, even in the event of a system crash, cannot be maintained, as noted in the comment. But that's a risk we have to take, because on systems that don't offer rename, we are forced to do so.

For a long time, even though more and more of the systems on which I was running my code supplied rename, I used the above routine (under a different name) to implement the limited semantics. I still avoid using rename for any other uses, and I think that caution is merited, as will be defended in the rest of this document.

Despite this aversion to using rename to move files or directories, I do now use it to rename files within a directory. So what's the big deal?

Section 3 describes seven different situations that I have encountered in using rename, even when I limit myself to the renaming of a file within the current directory. So if you are thinking that the Open Systems push is going to alleviate your porting problems, read on. The problems I have discovered with a routine whose semantics can be almost duplicated in five lines of fifth edition code, might cause you to think again. But before launching into these problems, let me discuss installing a file, something one wants to do occasionally.

How to Install a File

Inherent in the process of building a system is the problem of copying the new version of the file into its destination directory. If you just say, "Use install(1)", you have not considered the problem, nor have you looked at the some 20 different implementations of that command. Furthermore, you are probably dealing with very small software system, because the customary sh implementation is excessively slow when dealing with thousands of files. But the efficiency is a minor consideration - the inconsistency of the interface is not, but we will ignore that problem for now.

The major failing of the standard shell implementation of install is that it is woefully naive and far too trusting!

When one is installing a production file, one must take all the measures one can to ensure that the installation fully succeeds or fails noisily! Furthermore, one must ensure that one never leaves the system in a state in which neither the new copy nor the old copy is available and usable! For those of you who have never experienced it, let me assure you that trying to cope on a system that failed to correctly install a new copy of `/bin/sh`, after it removed the old version, is not a pleasant nor relaxing situation[1].

1. This happened to us once at the University of Toronto in 1975. It was exciting and yet worrying.

It is to provide this guarantee that rename should be used.

But it is not that simple. Often one has to manipulate the file once it has been installed (e.g., apply strip or ranlib). Without further ado, I use a program called instal[2] which is now briefly explained, as is very good at manifesting and detecting problems with rename.

Due to its complexity and importance, instal provides a -X flag, in addition to the ubiquitous -x flag. Its output is presented below and rationalised.

```
% instal -X
instal [flags] targetdir/file newcopy
```

will do the following: if targetdir does not exist

```
    mkdir targetdir [1]
if targetdir/@nfile exists
    unlink targetdir/@nfile [2]
copy newcopy targetdir/@nfile [1,3]
if size(newcopy) != size(targetdir/@nfile)
    abort # check if yet another F.S. failure
chdir targetdir [2]
if -r flag ranlib @nfile
if -s flag strip @nfile
if -o or -g flags chown/chgrp @nfile [4]
if -M flag chmod @nfile [4]
if file exists {
    if @?file exists [5]
        unlink @?file link file @?file [1]
}
# NOTE: @nfile exists and file does not or is
# linked to @?file
rename @nfile file [1,2]
if @?file exists and ! -k
    unlink @?file [6]
```

[1] Check if successful using stat(2) - abort if not

[2] abort if returns -1

[3] this is a built-in copy - cp(1) not used to save time and to facilitate checking the result of every write

[4] aborts if -1 unless -l flag specified

[5] To deal with problem on systems that prohibit removing a ETXTBSY file and other situations when a file might not be removable, instal iterates over '@[0-9]file' until one is found that does not exist or can be successfully unlinked.

[6] Failure ignored if due to busy text file, otherwise aborts.

Note the following:

- instal copies the new file into the target directory, but not to the ultimate name. The copy is done to a temporary name, thereby protecting the old version until the new copy has been fully installed in the directory and any subsequent processing has been completed.
- The success of the copy is checked by both examining the result of every write(2), the close(2) and then (since I am a paranoid) by checking that[3] the size (as retrieved using stat(2)) has not changed.
- We do not rename the new file to the new location before the old version has been safely linked to a new name. This process is complicated by the possible need to try multiple names before the link can be successfully performed.

2. Note spelling change to ensure that the correct version is used, no matter setting of ${PATH}.
3. Yes Viceginia, there are file systems on which copying a file can change its size (even ignoring sparse files such as those created by dbm).

- Unlinking the old version is immediately followed by linking in the new version, with absolutely no intervening processing! These operations are combined using rename if possible. This is essential since once the unlink of the old file has succeeded, the program does not exist under its usual name - think again about a crashed system on which /bin/sh does not exist.

The desirability of using rename in the above processing should be obvious. Its guarantee of preserving the to file is necessary. But before moving on, consider the following two points:

1. instal is used to install instal itself and other programs that might be "busy". This is why care is taken to link the current version of the target file to the temporary @?file so that the ultimate rename that unlinks the original and links in the new file does not destroy the last link to a busy text file.
2. Some versions of make, including our variant mimk, run processes in parallel. Consequently the situation in which multiple instals are manipulating links within the same directory arises often. However, the same file is never involved in parallel renames.

Six different permutations of rename, plus the degenerate case

Let's now return to the examination of the rename function, which is supposedly the subject of this paper. The following sub-sections describe experiences I have had with rename via the instal program. With the exception of the sixth case, each of the cases exist on one or more of the systems attached to our network[4]. The reader should also keep in mind that I am limiting my use of rename to regular files within the current directory, thereby eliminating problems that can arise due to busy directories, cross device or directory links, circular paths, etc.

Behold - a working rename

There are systems that actually supply a rename function that seems to work - well, let's say that I have not as yet detected a failure due to a software bug. I mention it to demonstrate, through an existence proof, that it is possible.

rename undefined

As mentioned, there are relatively new systems that still do not provide rename. However, of the seven situations listed in this paper, this is the second easiest with which to cope. If one cannot have a working version, one is almost better off with no version at all. One is faced with the problem of where to put the unlink/link/unlink implementation and how to adapt imported software that expects it to be in place, but this is a situation with which I am well familiar.

rename defined - as a no-op

There is a version of rename, distributed by one organization that does nothing - absolutely nothing! It does not rename the files, it does not return -1 - whoops I tell a lie - it does something - it returns 0, which is not exactly nothing, but very close to it.

This situation is actually not the most unpleasant, as the problem is discovered the very first time one tries to use rename(2). In this situation one immediately treats the offending system as if rename was undefined.

targons. With the exception of the sun, all constructions are done on a local disk. All installations are to a local disk.

4. Machines on our network currently include targon35s (pyramid) using both att and ucb universes, apollo 3500s (bsd side), mips (bsd side), mx300s (att side), wx200s, m88ks, an old sun3, and various 386s. All constructions are done using source accessed via NFS to one of the

Almost works - just one minor recoverable glitch

Remember how I stated how instal was to install itself? There is yet another variation on today's theme in that there is a rename that works everywhere, except when trying to rename a busy text file (e.g., instal). We have not seen any mention of this situation being illegal. In fact on the same system, the unlink/link/unlink simulation works as the link increments the link count thus the unlink does not reduce it to zero. Some implementations of unlink legitimately object if an attempt to remove the last link to a busy file is attempted, but rename should not raise this condition when the busy file is the from file. So, on such systems instal checks for errno==ETXTBSY after a failed rename call. If the test is positive, the unlink/link/unlink simulation is used.

Discovering if the system's rename exhibits this errant behaviour is simply a matter of compiling and executing the following program:

```
main()
{
    if (rename("a.out", "new.out") == -1) {
        perror("a.out");
        exit(-1);
    }
    puts("might have worked... please check
         new.out exists");
    exit(0);
}
```

Almost works - just one minor unrecoverable glitch

Some un-named manufacturer has distributed in the past a version of rename() that frequently fails to work when trying to rename a file on a remote file system. Unfortunately, trying to detect that the rename failure is due to a file system time out is difficult to do in a consistent and portable way. Furthermore, it goes against my paranoiac nature to try to second-guess hard errors.

The only course of action is to restart the building process, which almost inevitably completes successfully the second time. This success-on-second-try phenomenon may be due to the fact that the second run is usually done after the constructions on all the other parallel construction streams have completed[5], thereby with a much reduced network load.

It is not definite that this occasional failure is due to a rename bug, but because its second time success rate, and the rarity of occurrence on a relatively unimportant system, it has not merited much investigation. It's mildly irritating, but not as much as the next two situations.

Panic - it's multiple renames

Surprisingly, the version of rename that frequently caused a kernel panic is not the most unpleasant, but it's close. This implementation of rename seemed to cause the system to get its knickers in a twist when multiple renames were being run in parallel. The result was a kernel panic and a mildly corrupted file system, usually in the directory in which the renames were attempting to function when so rudely interrupted.

The solution in this case was initially to eliminate the parallel runs, and ultimately to change jobs.

5. Our major system builds are done as three parallel job streams, each stream processing three or four of the platforms or configurations. All the constructions use the same remote source file system, via our LAN; consequently the network is usually fairly busy servicing three job streams, each of which is running multiple parallel compiles.

Panicking would be preferable.

Finally the ultimate rename disaster. The most recent new port of our software gave rise to a situation on UNIX with which I was previously unfamiliar. Way back when - circa 1977 - I saw a directory created in which the normal ordering of.. and. was reversed. It was amusing, yet not unhealthy. However, until I encountered the rename bug described in this section, I have never consciously seen a directory with duplicate entries, that is two entries in the same directory, with the same name, referring to the same inode. Again, this is not as unhealthy as it might seem - provided the inode's link count includes both entries. UNIX is fairly adept at handling multiple references (i.e., links) to the same file. The name, for the most part, is irrelevant, once the directory search has retrieved the inode number. You would find that, if was possible to create two entries in a directory with the same name, the file system primitives and tools (notably rm) would all work properly. This is, for the most part, true, even when the duplicated names had different inodes, although specifying or predicting which inode was used could be difficult.

Unfortunately, if the inode link count does not include the duplicate entry, which is the situation produced by this rename bug, the file system is mildly corrupted, but soon to be dramatically corrupted as soon as one of the duplicate entries is unlinked. This will reduce the inode link count to zero, thereby returning the still-in-use inode and disk- blocks to their respective freelists[6].

The first time that this situation arose, the full system construction had completely, supposedly successfully, at which point the file system integrity checking package aborted on a sequence error. Thus, detecting this problem proved to be yet another defence of one of Dr. Tuna's Software Hygiene Nostrums [Tilbrook 90]:

- Create and frequently use a file system integrity package that checks for spurious files, missing files, obsolete files, and so on.

It was the "and so on" check that revealed the problem. The comm(1)-like process that compares the installed file list against the master list aborts on sequence errors and duplicated lines - something which comm does not do, hence its replacement. Initially I assumed that the sequence error was in the manually maintained master list, so I sorted this list and reran the file system integrity package. However, the same sequence error was reported indicating a sequence error or duplicate line in the following command's output:

```
find . -print | sort
```

My experience with sort is that it works almost without fail[7], so I ran:

```
find . -print | sort | uniq -d
```

There were dozens of duplicate lines, a situation I had never experienced in my previous 16 years of UNIX use. Furthermore, I discovered dozens of files pointing to the wrong contents, probably due to an in-use-inode being reallocated to a new file. Recovery was difficult and time-consuming - I had to remove the entire production tree and start all over again. It did have one positive effect. I was forced to create yet another regression test that actually checked that, not only did the required programs exist in the right directory, with the right modes, but they also contained the right contents[8].

Further experimentation with the strengthened installation regression testing suite definitively fingered yet another rename bug. This bug was the most difficult to detect. It was also the most insidious in that it caused the most damage and required the most effort to repair and rectify. The solution was to eliminate parallel instals by reducing the maximum number of parallel processes executed by mimk (the D-Tree make replacement) to one. Since doing so, the file system has remained relatively healthy.

Conclusion

As stated in the abstract, the successful achievement of the objectives of the Open Systems push depends largely on the reliability and consistency of the software foundation.

My use of a simple and basic function that has been deemed to be part of that foundation, has yielded a dismayingly wide range of compliance and quality. This leads one immediately to the conclusion that the suppliers' approaches to system validation is less than rigorous.

rename is a function that can be tested relatively easily. Checking that a call to it works is easily done using any number of standard tools (e.g., ls) or mechanisms (e.g., stat). Actually, rename is so fundamental that it is rather surprising that its failure is not detected fairly quickly by the supplier through normal system use - assuming suppliers actually use their product before they ship it. I detected the problems in the five buggy versions through normal use.

If suppliers have problems creating and testing a function as simple as rename, am I wrong in thinking that they might have similar problems when it comes to more complicated routines with much broader domains, such as semaphores, sockets, signals, process control, programming languages, or window managers[9]?

Am I unjustified in being publicly critical of some of the suppliers who exhibit what can only be interpreted as gross negligence in the interest of marketing or politics? The cited problems with rename form a very small, but highly representative, subset of the problems I have discovered in trying to maintain our products on 9 different platforms simultaneously.

What can one do to improve this gloomy forecast?

Hit them where it hurts - in their commission belts. When some salesman starts promoting his/her product as a wonderful, leading edge Open Systems approach solution to your problems, say:

"Oh yeah? Tell me how you test your rename(2).

Then we'll get on to things that aren't trivial!"

Epilogue #1

To verify the origin of the first rename I called Kirk McKusick. In our conversation he told me that when he presented the design of the new directory format and the kernel implementations of the mkdir, rmdir, and rename functions to the 4.2bsd review committee, Dennis Ritchie questioned the wisdom of allowing one to rename directories. Dennis thought that it would be very difficult to get right. Kirk then went on to state that Dennis's insight was correct: implementing rename was one of the most difficult coding tasks he has ever done. Furthermore he related

6. Note that, in the event of a system crash during a rename function, the link count for the from file might actually be one higher than it should be. This is because the from link count is incremented before the to entry's is set to that inode. But this is a situation that may be harmlessly rectified by fsck when the system is rebooted.
7. Mind you, its performance is pretty abysmal on some of our 386 systems.
8. The test was to run all the programs with the -x flag, which generates the program's synopsis and description, and compare the collected outputs against a master list.
9. Examining the variations in interpretations of the C language's semantics would be another, much longer, paper. Discussing coping with different releases of a window system is a thesis.

some problems that I have not discussed, but occur even when we limit the analysis to renaming files within the current directory. For example, consider the problems of executing any two of the following calls in parallel:

```
rename("alpha", "beta");
rename("alpha", "gamma");
rename("beta", "alpha");
rename("gamma", "alpha");
```

while ensuring the guarantee that the "to" file existed, even in the event of a system crash.

Granted ... getting it right is difficult, but not impossible.

Epilogue #2

Initially this paper was written quickly, without any intended destination or audience. It was just a case of: "this mess should be recorded". In the absence of a formal outlet, I mailed a copy to another renowned curmudgeon, Barry Shein, of Software Tool & Die.

The next day, I received a reply, which contained in part:

"That rename() grump is a good one, a good sermon. It's worse than you state, at another level. The world is now going nuts.

Under BSD and all versions of Sun/OS previous to 4.1 that I know of, the SYNOPSIS for the manual section for rename (and many similar "from->to" kind of calls) looked like:

```
rename(from,to)
char *from, *to;
```

as you mention. Now, in some sort of rabid attempt to be SVR4/ POSIX/1003 or something compatible, someone decided to change all those SYNOPSIS to:

```
rename(path1,path2)
char *path1, *path2;
```

Now, isn't that precious? What might have motivated a company to call forth their legions of tech writers to change these manual pages in this way?"

Given that new synopsis, how long do you think it will be before I can add an eighth case (the one in which the arguments are reversed) to my list?

Bibliography

Tilbrook 90 David Tilbrook and John McMullen, Washing Behind Your Ears - or - The Principles of Software Hygiene., Keynote address, EurOpen Fall Conference, Nice, 1990.

EurOpen Executive Report

Helen M W Gibbons
European Forum for Open Systems (EurOpen)
Buntingford
Hertfordshire
United Kingdom

Email europen@EU.net

Helen Gibbons is the business manager of EurOpen and is contactable at the EurOpen secretariat.

New House Style

Everyone must by now have noticed EurOpen's new House Style. This had been evolving, along with the new logos for all the National Groups, and soon all EurOpen material will be easily identified by its new distinctive style.

Governing Board

At the time of writing preparations are under way for the next Governing Board meeting which will be held in Tromsø on 18 and 19 May. The meetings will be reported in the next issue of the newsletter. This report, therefore, is of necessity brief. Much more news and information will be available after the Governing Board has met.

Special sub-committees from the Governing Board have met to discuss new by-laws (consisting of Mick Farmer, Georges Schild, Jacques Guidon, Jaana Helsingius and Joy Marino) and a new subscription algorithm (consisting of Kim Biel-Nielsen, Dominique Maisonneuve, Kel Simonsen, Fritz Kofler and Donal Daly) and these matters will be debated at the forthcoming meeting.

Executive Committee

The Executive Committee has met twice already this year on 13-14 January in London and 17-18 March in Amsterdam, to deal with the management work load, and it has held interviews and talked to applicants with a view to appointing in the near future the new EurOpen Managers.

EurOpen Working Groups

After the first EurOpen Working Groups meeting held in Brussels in December, 1990, the Working Group Secretariat has passed from AFUU to the EurOpen central office, and a second meeting was called for the 26 March 1991 in Paris.

Although the initial response from the National Groups had seemed favourable, it transpired that very few were prepared to attend the meeting and therefore it was postponed.

The whole subject of EurOpen Working Groups will now be discussed at the Governing Board meeting in Tromsø and there will be a presentation during the Conference by Jean-Michel Cornu, the Working Groups Co-ordinator. If the Governing Board and the National Groups respond favourably an attempt will be made to recall the second meeting later in the year.

Publications

The Membership Directory had been sent out and has been well received. Plans are in hand for producing a second one if the Governing Board agrees.

The 2nd edition of the European Email Directory is also now available at a cost of £20 (30 ECUs) from the Secretariat.

International CommUNIXations was sent out free to all members with the last newsletter. Requests for further subscriptions can be sent to the EurOpen Secretariat.

Budapest Conference 16-10 September

At the time of writing this report we are finalising the technical programme for Budapest with the help of the Programme Chair, Andrew Findlay. Tutorials planned to date include:

Vaclav Rajlich	'Decomposition/Generalisation methodology for C++ programs'
Matt Bishop	'UNIX Security'

Bill Rieken	'System V Release 4 Internals'
Bill Rieken	'Introduction to X Windows' (half day)
Bill Rieken	'Programming Environments' (half day)
Donald Lewine	'POSIX for Programmers'
Mike Karels & Kirk McKuisick	'New Kernel Facilities in 4.3 BSD – Reno'
Evi Nemeth	'UNIX System Administration'

An Exhibition will accompany the event entitled 'UNIX Goes East'.

EurOpen Hungary, under the capable hands of Maria Toth has helped the Secretariat to organise a fine Conference of a high technical quality in a superb location. Budapest is one of the most attractive, scenic European capitals, standing on both sides of the River Danube, and you are encouraged not only to attend the Conference but to also spend a few days exploring this fascinating city.

Budapest Airport has direct flights connecting with 42 cities in Europe and overseas. The Airport terminals are served by a regular bus service. The public transport in Budapest is extremely cheap compared to most western cities and delegates should take advantage of the trams and buses to see this beautiful city.

The Conference & Education Centre which will host the Tutorials, Technical Sessions and Exhibition stands on the Buda side and is a purpose built centre with good facilities.

The Conference Hotel is the new 4 star Korona Hotel which is situated in the very centre of the Pest side. It is only a 10 minute bus ride from the Conference Centre.

The Conference Dinner will be held at the superb Summer Palace built at the beginning of the century. It has elegant rooms, large balconies and a garden and will provide an excellent opportunity for delegates to meet and enjoy a combination of Hungarian and International cuisine.

EurOpen Hungary has organised a very full touristic programme to accompany the event. Attractions such as visiting Renaissance Castles, The Danube, the historical wine district, the thermal baths, horseback riding and even learning how to cook special Hungarian dishes are just a few.

For a full information and bookings booklet please contact the EurOpen Secretariat.

There was an error in the last issue of the newsletter concerning the availability of Uniforum publications. These publications are now available from the EurOpen Secretariat at Owles Hall - please see the publications information on page 28.

EurOpen Autumn Conference

in conjunction with EurOpen Hungary
Budapest, Hungary, 16th – 20th September 1991

A major International Conference and Exhibition on Unix and allied subjects covering such areas as Distributed Applications, Internationalisation of Standards, Intelligent Systems, Object Management, Security, Design Tools for User Interfaces, and Computer Music.

Special Offer !

As a special offer for delegates attending the first ever EurOpen event in Eastern Europe, a FREE Conference Place will be made available to anyone submitting five additional delegate applications – provided the applicants have not attended an EUUG / EurOpen Conference.

Start recruiting Now !

For furthur details, contact:

The EurOpen Secretariat

Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

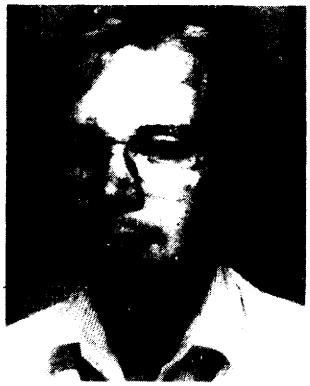
Telephone +44 763 73039
Facsimile +44 763 73255
Email europen@EU.net

European Forum for Open Systems

News from Iceland

Magnús Gíslason
University of Iceland
Computing Services

Email magnus@rhi.hi.is



Magnús is the secretary of the Icelandic UNIX Systems User Group (ICEUUG). He has been involved in UNIX since 1984.

Annual General Meeting

In November ICEUUG held its Annual General Meeting. None of the executive members wanted to be reelected, so a completely new executive was elected. Well, not actually elected, but kindly asked to volunteer. The new ICEUUG executive looks like this:

Chairman	Tryggvi Edwald
Secretary	Magnús Gíslason
Treasurer	Haukur Arnthórsson

New naming and logo were discussed and it was agreed to keep the name, but adopt a logo based on the EurOpen logo. ICEUUG didn't actually have a logo before.

Membership

The ICEUUG membership is increasing slightly and there are currently 37 members in ICEUUG, 11 academic and 26 commercial.

What are we doing?

ICEUUG holds informal meetings 5-8 times during the winter where a speaker talks about and introduces some topic to the members. This winter the topics have been:

date	speaker	topic
Oct	Einar Kjartansson	X Windows
Jan	Heimir Sverrisson	HBX-PAD (an X.25 PAD emulator for IBM RS6000 written by Heimir)
Mar	Ebba Hvannberg	Object Oriented Programming
Apr	Magnús Gíslason	Larry Wall's PERL

Publications

ICEUUG publishes a quarterly newsletter (Fréttabréf). It also has published a draft version of *UNIX Handbókin* (a UNIX Book in Icelandic) in cooperation with *Reiknistofnun Háskóa Íslands* (RHI, The Computing Services of the University of Iceland).

Network

The Icelandic network, *ISNET*, is run by the *Association for research networks in Iceland* (SURIS), which consists of most of the research institutions in Iceland. SURIS is a member of NORDUNET and EUNET and owns the Icelandic backbone machine, *isgate.is*, which is a Sun 3/60 located at and run by RHI. The backbone is also an archive site, where users on ISNET can get PD software. This is the only way ICEUUG distributes PD software to the members. Currently there are 33 sites on the net, thereof 16 with IP connection. 12 sites are receiving news.

Sweden Reports...

Anders Tjäder
FörsvarsData (Defense Data Systems Agency)
S-10787 STOCKHOLM SWEDEN

Email anders@f109a.mil.se

I'm a 23-year-old guy who's first contact with UNIX was in the year of our lord 1987. After finishing my studies in economics I discovered that economics wasn't my game so I did a longjmp to computers and UNIX. That's my story and I'm now working on a security project in the Swedish defense involving an AT&T-product called System V/MLS (Multi Level Security) which pushes a UNIX system up to a B1-security level.

Welcome to EUUG-S...

In march 1983 some folks banged their heads together and realised that 1) it hurts to bang your heads together 2) we needed a local EUUG (as it was called then before EurOpen) association which take care of the Swedish community of UNIX users. As a result they formed EUUG-S.

The number of members is rising steadily and I saw, to my surprise, in the first edition of the newsletter this year that Sweden was on fourth place in the table of members compared to the population. One remarkable thing was that five nordic countries were leading the table. OK, I realise that the Soviet-Union has a slightly different population figure than we have here in Sweden.

Population...

Here's the latest population figures (May 91)

Category	Members
Academic/ Commercial	448
Individuals	76
Students	13
Total	537

Name frustrations and all that...

There's an ongoing debate about changing our name and logo to something that would connect us more closely to EurOpen. The decision will probably be taken at our Annual General Meeting in Stockholm, Saltsjöbaden 30 September - 1 October 1991.

Conferences, Exhibitions etc.

The annual Swedish UNIX Expo is being held in Stockholm 6 - 8 November 1991. I haven't got sufficient information about this event so I'll be back on this subject, be sure!

Swedish Newsletter...

In our 1990 AGM we decided that our local newsletter should get a face-lift and it has! EUUG-S has engaged a consultant who works as an editor. This gets the layout a bit more professional than it has been before. The newsletter evolves and gets better as time goes.

Standards...

A working group has been formed in EUUG-S on the subject of standards. The goal for the WG is to apply standards to reality. One important field which the WG will investigate is a character set standard and the problems which arise around 7- and 8-bit character sets.

Summing up...

This was my first attempt to inform you folks about what's going on in the Swedish community of UNIX users and I'll hopefully continue with it.

If you want to get in contact with any of the people in the EUUG-S committee, send an email to me and I'll forward it to the right person.

We are soon in the middle of summer and vacations are hiding around the corner.

My advice to you is this: Get a sun-tan, not a tube-tan.

See ya'

Press Release from the AFUU

Ann Garnery
AFUU
11 rue Carnot
Paris
France

Email anne@afuu.fr

AFUU - AHRIA - CIGREF

A Forum for Open Operating Systems Users

Paris - Convention UNIX 91

The AFUU (the French Association for UNIX and Open Systems users), the AHRIA (the Association of data processing officials in Administration), and the CIGREF (the Data Processing Club for great French companies) signed an agreement on Open Systems on 27 March 1991, during the Convention UNIX 91.

This agreement is the result of a reflection conducted during the two previous years by Mrs Dominique Maisonneuve for AFUU, Mr Brion for the Ahria, and Mr Chevalier for the Cigref.

The agreement focuses on the promotion of UNIX and Open Systems as well as on the expression of users' needs.

The Promotion of Unix and Open Systems

18 months ago, the AFUU created a working group called Observatoire UNIX et des Systémes Ouverts. This working group works in close relation with the end users. It provides for an analysis on the brakes and obstacles users may meet when adapting open systems. It also provides the promotion of the open systems through a publication, Panoram'X.

According to the agreement, the AFUU allows ARHIA and CIGREF members to take part in the works of this working group with no obligation for them to become AFUU members.

Expression of the Needs

The 3 associations will set up a Forum for the users of Open Systems strictly reserved to the users belonging to the associations.

A high level Direction Committee will synthesise the Forum works and will transmit them to the users interlocutors: hardware and software companies, France and EEC authorities, standardisation bodies, the education and science world.

At first, the Direction Committee will gather representatives of each association (1 for each) with one or several forum members.

J.L. Anthoine, the current AFUU chair, comments: "These past few years, the AFUU has gained an acknowledged expertise in the field of Open Systems thank to its working groups and to its involvement in different national and international bodies. The agreement with the Ahria and the Cigref is an additional proof of the AFUU involvement, and will give the association a wider scope". R. Brion, Ahria chair adds: "Since a few years, the administration has become aware of the problems related to UNIX. The expression of our needs combined with those of the AFUU and the Cigref will make a new dialogue possible with suppliers". G. Chevalier, Deputy General Delegate to the Cigref, concludes: "Since its creation, the struggle against any form of captivity in the demand has been a concern for the Cigref. We consider this agreement as a new opportunity to encourage the development of Open Systems".

The AFUU gathers more than 1000 members belonging to education and research or to public and private companies. Every month, 12 working groups gather and actively contribute to the definition, development and adoption of Open Systems, whether technically or strategically.

The AHRIA gathers about 25 data processing officials in Administration belonging to the different departments of the central administration. The aim is to gather civil servants having the same occupation and responsibilities. It is a meeting place which allows an information and expertise exchange as well as a joint reflection on everyday or strategy problems.

20 years ago, the data processing club for great French companies (CIGREF) was created by 6 companies who wanted to share their users' expertise in the field of information technologies. Today, they are over 70 representing more than a half of the French stock, and about three quarters of the French companies telecommunication consumption.

The IUUG - A Status Report

Annrai O'Toole
Trinity College
Dublin
Ireland

Email annrai@cs.tcd.ie

*Annrai O'Toole is a research assistant with the Distributed Systems Group in the Department of Computer science, Trinity College Dublin. His research life began with an investigation into multiprocessor operating systems. This work was closely related to an ESPRIT project called COMANDOS, the aim of which was to COnstruct and MAnage a Distribuetd Open System. His main work is now concerned with this project and in particular an environment for a distributed and persistent version of C++ called C**. Details can be provided by mailing aotoole@cs.tcd.ie.*

Since the last AGM of this august body was held in November much has happened within the IUUG. Now is a good time for a report on our past activity and a view of things to come.

Much of the activity of the committee has revolved around a reorganization of the internal affairs of the group. In particular the network, or IEunet as it is now called needed much time and attention. Following in the footsteps of many other NalUUGs the IUUG has essentially laid the plans for separating the network administration from the other aspects of the groups activity. We believe that this should lead to a better and more professional (and competitive) networking service for UNIX users in Ireland. In addition plans are afoot to provide Ireland with an Internet connection. The only delay in providing this service now rests with the Irish Telecom company.

On the meeting front the IUUG has been running monthly meetings since the start of the year. These have included presentations from SUN and IBM on their current UNIX offerings, a talk on a persistent distributed environment for C++, Minix (small is beautiful) and TeX. All these meetings have been well attended and have provided a forum for some lively conversation. In particular, IBM had a difficult, but successful time in convincing the meeting of their commitment to Open Systems. There is now a schedule of meeting for the summer and autumn of 91 which we hope will be a successful as our last season. All in all, the committee of the IUUG are pleased with the status of regular monthly meetings.

Looking ahead to the future, the committed are putting together some plans for a IUUG conference for '92. This is obviously a large undertaking but we feel that it is important for the visibility and credibility of the group. The current proposals are for a one day conference with an associated one day tutorial session. Many new members have joined the IUUG since its last conference and we anticipate a good demand.

In short, much effort has been expended on getting the structures in the IUUG sorted out and stable. We hope that the fruits of this work will be evident in an active and vibrant IUUG, which provides for the needs of all its members.

PUUG Report

J Legatheaux Martins
Lisbon
Portugal

Email jalm@puug.pt

PUUG, What is in a Name?

PUUG, the Portuguese UNIX™ Systems Users Group, affiliated to the EurOpen, is a non profit association resembling Open Systems users and vendors. PUUG is the portuguese forum to exchange independent views on Open Systems and has about 100 members (mainly institutional: vendors, end user organisations, and universities).

We have already promoted several activities among which "The First Portuguese UNIX Convention - UNIX in the 90's", held in Lisbon one year ago. This conference had a great success with more than 300 attendees and a great impact in the press.

Our second anniversary

This year, during the commemorations of our second anniversary, PUUG will promote a series of tutorials during the week starting the 3rd of June.

Among the possible tutorials, we have selected tutorials of technical value to our members, not available in the portuguese training market. Here is the list:

T1	3 June 1991	An introduction to Open Systems, *J Legatheaux Martins, PUUG*
T2	3 June 1991	Micro Kernel UNIX Operating Systems (Mach and Chorus) *Marc Rozier and François Armand, Chorus Systémes*
T3	4 June 1991	An Introduction to C++ *Manuel Sequeira, INESC*
T4	4 June 1991	Advanced Network Programming *Henrique João Domingos, Faculty of Sciences of the Univ. of Lisbon*
T5	5 June 1991	The X-Windows System and the main available Toolkits *Nuno Guimarães INESC*
T6	5 june 199	1 The UNIX System V release 4 Operating System *Julian Lomberg, AT&T (USLE)*
T7	6 June 1991	The OSF/1 Operating System *Per Pederson, OSF (Germany)*
T8	7 June 1991	OSF/DCE - Distributed Computing Environment *Andreas Kronberg, OSF (Germany)*

PTUnet/EUnet (Portuguese UNIX Network)

We are in the process of setting up several services to our members. We already have a tape duplication service and we sell several publications (EurOpen and EUnet publications, as well as several books of the collection "UNIX in a Nutshell"). This year also, PUUG got deeply involved in the managing of PTUnet, the portuguese EUnet branch.

EUnet in Portugal is available since several years ago (5 I think). The first site and backbone was INESC, an R&D private institution. Until the last year, only 4 or 5 sites where connected to EUnet in Portugal. VAX/VMS systems where very popular, and a lot of universities where using X.25 and an experimental version of X.400 to communicate. EARN was available for free, and was the last resort for a lot of other sites. Eunet was also available for free since INESC doesn't charge for its services.

Since its beginning, PUUG had, as a clear goal, to help to promote EUnet in Portugal. We think that this promotion is a strategic option of our association and are completely committed to this process. In order to promote EUnet, PUUG:

- bought the first trailblazer that came to Portugal (the one used today by INESC to communicate with mcsun; fortunately, it will be soon replaced by a direct TCP/IP link)
- has delivered a tutorial on uucp and TCP/IP with the help of Yves Devillers
- got a new machine (dec4pt), offered by Digital to help the promotion of EUnet
- asked INESC to share the responsibility with us.

Since last month, PUUG and INESC came to the following agreement:

- INESC will continue to be the Portuguese backbone until PUUG proves that it is technically and financially able to run the network. We can not afford the price of the X25 link used, until recently, by INESC to import news from mcsun.
- PUUG would take in care all the customers outside INESC and would pay INESC all their expenses and pay INESC a share of 80% of the income coming from news. It will be the first time that the backbone gets something for its services.
- PUUG runs some sort of sub-backbone.

Now, EUnet has 10 sites in Portugal. 3 are Inesc sites (INESC is a nation wide organisation with departments in several cities). 7 are PUUG sites: 1 is the sub-backbone, 3 are commercial and 3 others are academic. Everybody, except inesc, is paying PUUG to maintain its connect status to EUnet. We think we will have at least 5 new customers this July.

PUUG is investing a reasonable share of its resources in the network. We bought 4 Trailblazers and a CISCO box that will be soon available (we are now using the CISCO box of the faculty that hosts PUUG). dec4pt is available by dial up lines (we have 3 lines), uucp above X25, and TCP/IP for academic sites. Our link with inesc is a TCP/IP link above the infrastructure of the academic network.

Contacts

The PUUG board can be contacted at:

PUUG
Grupo Português de Utilizadores de Sistemas UNIX
Avenue 24 de Julho, no 134, 7° Lisboa
PORTUGAL

Email sec@puug.pt

Telephone +351 1 395 06 42
Facsimile +351 1 67 18 76

News from Switzerland

Patrik Eschle
Physics Institute University of Zuerich
Switzerland

Email eschle@physik.unizh.ch

Patrik Eschle is the secretary of the CHUUG. When not busy with connecting telephone lines for the backbone, he still tries to bounce laser beams from each other in sodium vapour as part of his Ph.D. thesis.

Membership

The number of members in the CHUUG is still growing slowly. We currently have 58 institutional, 33 individual and 1 sponsor member.

Logo and Name

The CHUUG exec presented to the general assembly (held on 16th April in Olten) a new name 'EurOpen Switzerland' and a new logo, the EurOpen logo in red with a white swiss cross. The members were not pleased with both and decided to stay with the old name CHUUG and let the exec board propose a new logo, that is not based on the EurOpen logo.

One of the English speaking members pointed out, that CHUUG (if pronounced properly as tschugg) is preferable to 'EurOpen Switzerland' because it has the nice sound a of steam engine...

Backbone

The CHUUG has finally started operation of its own backbone machine 'chsun'. After a test period in March and April, the general assembly decided to run the machine as official Swiss EUnet backbone. A working group inside the CHUUG will survey backbone operation and report to the members at the next general assembly.

The main reason for CHUUG to run the backbone by itself is to have full control over budget, tariffs and services by the members of the CHUUG.

The backbone currently offers uucp connections for mail and news, an online archive, a tape distribution and mailboxes.

The link to the world is through mcsun by modem, inside Switzerland we cooperate with the other existing networks.

The offices are situated in Zuerich (where most of the UUCP sites are). Visitors are welcome at any time (please give a phone call or fax first), although most of you have already seen a couple of SUNs and modems:-)

The start of backbone operation wouldn't have bee possible without the excellent job Simon Poole did and is still doing. I'd like to thank him for the time and knowledge he invests to realize the idea of a network owned and controlled by its users.

If you want to connect to the backbone, please contact:

CHUUG
Patrik Eschle
Viktoriastr. 19
CH-8057 Zuerich
Switzerland

Email chuug@chuug.ch

Telephone +41 1 313 13 26
Facsimile +41 1 312 49 77

UKUUG Column

Mick Farmer
Department of Computer Science
Birkbeck College
London, UK

mick@cs.bbk.ac.uk

Mick is the Secretary of the UKUUG. His primary interests are Ornithology (restricted to the Western Paleartic at the moment because of cost) and Oenophilism (especially pre-1962 Bordeaux, pre-1980 Burgundy, and 1945 Port). His secondary interests include Software Consultancy (to pay for the above primary interests) and Distance Learning Methods (especially interactive video and hypertext). When not pursuing these and other interests he can be found at Birkbeck College (London) where he teaches in the Department of Computer Science.

He lives in Lewisham (South East London) with his wife Sue and a TV called Sonya. His neighbours have two children and a dog.

Start Bit

Don't you just love the new EurOpen house style? Lots of thin lines (and thick lines) starting (and stopping) all over the page. Some even change colour. Evidently a designer house style *- you can't tell digit ones (1) from lower-case ells (l). See what I mean!

Quiet Lull

There's very little to report from the UK group since the last Newsletter. We haven't yet decided whether to hold a pan-European conference next year (we work in biennial cycles). The plan is to broadcast the conference using interactive video relayed through the national backbone sites (who provide real-time language translation) to interested parties. No travel costs. No accommodation costs. All for the cost of a telephone call. The national PTTs should provide the necessary infrastructure real soon now!

Dates for your Diary

- 15-17 July, 1991 – Summer Technical Meeting at Liverpool University, Liverpool (England).
- 16-18 December, 1991 – Winter Technical Meeting at Herriott-Watt University, Edinburgh (Scotland).
- .Summer 1992 – Queen's University, Belfast (Northern Ireland).

If you and your organisation, be it academic or commercial, are interested in hosting either a Summer or a Winter meeting please contact Jim Reid (jim@cs.strath.ac.uk) in the first instance. There are many areas of the country that we haven't visited, e.g the Midlands, the West Country, anywhere north of Glasgow, etc.

FaceSaver/FaceServer

Our FaceSaver kit was not in evidence at Tromsø because we're in the process of upgrading the service. Additional storage and networking capabilities are being added. In addition, the FaceServer, based at Imperial College (London) alongside our software archive, should soon allow you to request faces from outside the UK.

Stop Bit

We're still interested in collecting the words of songs, bollards, poems, etc. concerning the early days of UNIX. If you know any such pieces please send them to me at the usual e-mail address together with their provenance if known.

Report from the Netherlands

Emile van Dantzig
Transmediair
The Netherlands

Email <emile@tmu.nl>

Emile has been a general manager at Transmediair for 1 1/2 years where he brings turnkey products to market though a miraculous applications generator called SuperNova.

NLUUG Membership

The NLUUG's membership continues to increase in number. In the table below the increase in academic, industrial and individual membership is shown over the last three years.

April 89	April 90	April 91	Member type/fee
80	90	97	academic members (@ ECU 136)
164	193	237	industrial members (@ ECU 272)
35	35	49	individual members (@ ECU 45)
279	319	383	TOTAL

The board will propose a new fee structure to its members, introducing a new form of individual membership with possibilities to have connection with NLnet and a subsidiary membership. Individual members had no access to the network before. The subsidiary membership will be interesting for individuals employed by academic or general members. They will have no access to the network, but they will receive the Newsletter and will pay the (reduced) membership fee for conferences.

There are many ideas how to offer more facilities to the end-users and industrial members, while maintaining the services to our technical involved members by offering interesting conferences.

NLUUG Board

Two members of the NLUUG board, Marten van Gelderen (chair) and Ted Lindgreen (member), were re-elected at the last AGM for a period of two years.

New Office

The NLUUG has a new office and a new office manager. As of 1st March 1991 Sandra Scharff runs the new NLUUG office.

The new address of the NLUUG office is:

NLUUG office
p/a Transmediair B.V.
P.O.box 297
3720 AG BILTHOVEN
The Netherlands

Email nluug-buro@nluug.nl

Telephone +31 30 281820
Facsimile +31 30 292294

At the last AGM Patricia Otter, our previous office manager, was thanked for the 5 years she had been involved with the organisation. She helped to transform the NLUUG from a purely volunteer based organisation to a more professional one. In Ede, at our recent conference, she was awarded honorary membership of the NLUUG.

New Logo

As agreed in Nice, NLUUG has decided to follow the proposal for standardisation of logos and has incorporated the characteristic

contour of The Netherlands in the new style logo. Also the colour has been changed from red into Lilac.

To change both logo and office required a tremendous investment of time and effort. This is an exercise which should not be repeated to frequently. The preparation of the spring conference together with all these changes was the acid test for Sandra and her assistants. She has done very well.

Help Desk

Our new trial service - a help desk, which started in June 1990, is still in a trial due to the change of office and lack of personnel force.

The attempt, to answer all questions without favouring one vendor or another, will be continued and professionalised.

Conferences

Our last conference about "System Management, Securing Open systems in Open Networks" was held last 8th May at Ede.

Considering the number of attendees (330), this conference was very successful. Also the clear distinction and separation between product sessions and technical sessions was an improvement appreciated by the attendees. This distinction was pointed out in the program booklet and separate tracks in different rooms avoided confusion for the attendees. The tutorials were over crowded because of the very popular topics, the well known speakers and the small room.

All sessions (including tutorials, coffee, tea, lunch and drinks) and the proceedings were included in the registration fee of DFL 125,-- (ECU 57) for members of the NLUUG and DFL 350,-- (ECU 160) for non-members. Abstracts of the sessions will be included in the next EurOpen Newsletter. The proceedings (mostly in Dutch) are available through Owles Hall.

Our next conference will be on *Standardisation, promise is debt.* It will be held on 7 November 1991 and include an exhibition as well.

The topic of this conference is to present information from software developers and large companies who changed from a proprietary system to an Open Systems environment to address such issues as the impact of development- and conversion budgets and portability of software products.

In June 1990, NLUUG provided co-operation to a POSIX symposium of the NGI (The Dutch Informatics Association) at the Jaarbeurs at Utrecht.

Working Groups

The NLUUG is trying hard to start up Working Groups, as part of the EurOpen Working Group (EWG) initiative. It is not easy to find people and organisations prepared to invest time and money in this type of work. NLUUG has been a guest member of the Dutch X/Open group from the VIFKA for a long time. In 1987 the NLUUG initiated a Working Group on market oriented topics. This group has evolved into the VIFKA.

NLUUG is trying to find people from large organisations prepared to participate in working groups addressing topics such as portability, user requirements, standards, etc. We are happy that we have found some people prepared to fulfil this mission but we are still looking for more members to participate.

Backbone

For those who are not aware of the NLUUG/NLnet construction, here comes an explanation. The NLUUG established a foundation called NLnet in 1989, to strictly separate the financial risk and organisational aspects. Conforming to the bylaws three board members from NLnet share their board membership with the NLUUG. This construction maximises communication and minimises conflicts, as is our experience so far. If other NALUUGs are interested in this construction, please contact nluug-buro@nluug.nl.

The contract with SURFnet (the network group from the Dutch University Computing Centre Association) will be ended in May 1991, because of their direct connection with Internet. The end of this contract will result in a substantial decrease of income. However, this will have no effect on the service level. NLnet has diminished the contribution for network traffic for all members on January 1990. The algorithm for the new discount regulation is:

Kb tariff X 10000 X ln (1 + Kb volume/10000)

So the tariff will be dependent on the volume of the mail traffic.

The growth of the number of members of the NLUUG and NLnet, introduced an unexpected problem for our treasurer. Because of the high revenues the tax agent knocked on his door holding out his hand asking for quite a lot of money.

Preliminary Announcement

AUUG '91 Darling Harbour, Sydney, Australia, 24-27 September 1991

The Australian Open Systems Users' Group

Just What is an Open System, Anyway?

Everyone is talking about Open Systems. So what are they?

"An Open System is one which can interwork with any other Open System using internationally agreed OSI standards."

International Computers Limited (ICL)
The OSI Handbook (1989)

"The standard criteria for an Open System are portability, compatability, interoperability, and scalability. An Open System gives customers 'freedom of movement' in every sense of the phrase. Portability gives customers the freedom to move across different computer makes and models. Compatability gives customers the freedom to move from one Open System release to the next as the technology evolves over time. Interoperability gives customers the freedom to operate within a distributed heterogeneous environment of many vendors. Scalability gives customers the freedom to move between systems of different sizes."

UNIX International Inc.
UNIX System V Release 4 vs AIX 3.1 (April 1990)

"An Open System is designed to international or de facto standards and, therefore, can function compatibly with any product designed to the standards."

Unisys

"A computing environment that:

- Is vendor neutral;
- Complies with formal international IT standards;
- Permits system and network interoperability;
- Permits software application portability;
- Offers consistency of data and human access;
- Satisfies one or more of a business' functional requirements."

Digital Equipment Corporation (Australia) Pty. Ltd.
Open Systems - A Guide for Information Decision Makers, March 1990

"A comprehensive and consistent set of international information technology standards that specify interfaces, services, and supporting formats to accomplish interoperability and portability of applications, data, and skills."

ibid

"Software environments consisting of products and technologies that are designed and implemented in accordance with "standards", established and de facto, that are vendor independent, and that are commonly available.

X/Open

"Open Systems means that they want to open my wallet."

Geoff Collyer

AUUG '91 wishes to explore Open Systems from all perspectives, including:

- Standards

  standardisation philosophies, emerging standards, standards deficiencies, experiences with standards implementation

- Operating Systems issues:

  multiprocessor systems, distributed systems, secure systems, fault tolerant systems

- Communications and Networking

  protocols, performance, administration and security

- Programming environment

  user interfaces, windowing, graphics, compilers and language technology, software development and support tools, testing and debugging

- Sophisticated applications

  databases, transaction processing, commercial, educational, scientific, biological, medical, etc.

Events

AUUG '91 will be a four day conference with the first day devoted to tutorial presentations, followed by three days of papers, Work-In-Progress and Birds-Of-a-Feather sessions.

For further information contact:

Andrew Gollan
AUUG '91 Program
C/- Softway Pty. Ltd.
PO Box 305
Strawberry Hills, NSW, 2012
Australia

Email auug91@softway.sw.oz.au

Telephone +61 2 698 2322
Facsimile +61 2 699 9174

Please be sure to include your postal and electronic mail addresses in all correspondence.

EurOpen Publications

EurOpen publications may be ordered from the Secretariat at Owles Hall.

The publications available are listed on the next page, it is planned that EurOpen members will be able to also order National Groups publications from one single point: the secretariat.

The EurOpen publications available are:

EurOpen Newsletter
Your EurOpen National Group membership gives you one free copy of this newsletter, you may order extra copies of the newsletter for distribution within your organisation. Please contact your national group who will arrange this for you.

Proceedings
Proceedings from some past conferences are still available. The list of those that you can order is on the next page.

European E-Mail directory
This is a must if you are a serious e-mail user. It provides you with a way of referencing e-mail sites thoughout Europe by means of several different indexes.

The second edition to this has been prepared and will be available soon.

USENIX Publications

We receive requests for USENIX publications and, as a result of our close cooperation with that organisation, we are pleased to announce a service by which a range of publications can be ordered through EurOpen. The range includes:

;Login:
This is the USENIX newsletter which is published every two months, and carries articles on a variety of topics of interest to UNIX users everywhere and usually runs to around 40 pages.

Computing Systems
This is the journal of the USENIX Association, a quarterly publication which is devoted to the analysis and understanding of advanced computing systems. It is perfect bound with a printed spine for ease of reference and usually runs to around 100 pages.

Proceedings
Some proceedings from past USENIX conferences and workshops are available from stock - the current list is enclosed with the attached order form.

4.3 BSD Manuals
These manuals have proved popular with EurOpen members but our stock is now exhausted. However, USENIX can continue to supply them direct and the only difference from the past ordering procedure is that you may have to wait a little longer to receive them direct from the States.

Ordering procedure. An order form is enclosed with details of how payment can be made by credit card, by direct payment to the bank or by certain types of cheque or banker's draft.

Publications available though EurOpen

			£	ECU
;Login:	The USENIX newsletter	6 issues/year	20.00	30
Computing Systems	The USENIX Journal	4 issues/year	30.00	44
USENIX Proceedings	Anaheim Conference	June '90	25.00	37
	C++ Conference	Apr '90	30.00	44
	Washington DC Conference	Jan '90	27.50	40
	Graphics Workshop V	Nov '89	19.00	28
	Dist & Multiproc Workshop	Oct '89	33.50	49
	Large Inst Sys Admin III Workshop	Sept '89	15.00	22
	Baltimore Conference	June '89	24.00	35
	UNIX Trans Proc Workshop	May '89	13.50	20
	Software Management Workshop	Apr '89	24.00	35
	San Diego Conference	Feb '89	33.50	49
	C++ Conference	Oct '88	33.50	49
	C++ Workshop	Nov '87	33.50	49
	Graphics Workshop IV	Oct '87	17.00	25
	Washington DC Conf	Jan '87	20.00	30
	Graphics Workshop III	Dec '86	17.00	25
4.3 BSD Manuals (EurOpen Members Only)	User's Manual Set Programmer's Set System Manager's Manual		60.00	88

			£	ECU
EurOpen	Dublin	Autumn '83	2.00	3
Proceedings	Nijmegen	Spring '84	5.00	7
	Cambridge	Autumn '84	5.00	7
	Paris	Spring '85	5.00	7
	Copenhagen	Autumn '85	10.00	15
	Finland/Sweden	Spring '87	20.00	30
	Dublin	Autumn '87	20.00	30
	Munich	Spring '90	20.00	30
	Nice	Autumn '90	25.00	37
	Tromsø	Spring '91	Contact EurOpen at Owles Hall for prices	
Directories	European E-Mail directory, 2nd edition		20.00	30

All prices include post and packing.

The price for ;Login: and Computing Systems is for a one year subscription.

AFUU Publications available fromEurOpen

The following publications are written in the French language

		£	ECU
Dossier Benchmarks AFUU 1989 Results - 132 pages	Edition : Mars 1990 Summary send upon request	24.00	35
	Edition: Mars 1991	34.00	50
Réussir avec UNIX	Edition : 1988 - 15 pages	4.00	5
Petit guide destiné aux directeurs informatiques d'organismes ayant fait récemment le choix d'UNIX.			
Il traite de : L'environnement de développement sous UNIX L'environnement d'exploitation sous UNIX La formation La portabilité Les outils d'UNIX Les outils du marché			
Vivre avec UNIX 2 - Administration du Systéme	Edition : 1988 - 16 pages	4.00	5
Petit guide destiné aux techniciens des organismes qui ont fait récemment le choix d'UNIX			
Sommaire : Introduction Systéme de fichiers configuration de l'espace disque Création du systéme de fichiers Gestion des périphériques Gestion des impressions Administration quotidienne Assistance et environnement de l'utilisateur Sécurité Performances			
(Vivre avec UNIX 1 - Utilisation du systéme (Epuisé)	Edition 1988 - 16 pages	34.00	50
CONVENTION UNIX 90 - Conferences proceedings		14.00	20
End-users and solutions conferences proceeding	Edition Mars 1990 - 165 pages	14.00	20
Technical conferences proceeding	Edition Mars 1990 - 198 pages	14.00	20
CONVENTION UNIX 90 - Tutorials			
Are available the following tutorials (the others are no more available)			
Sendmail, Annexe 1 et 2	Edition Mars 1990 - 112 et 205 pages Par Yves Devillers (Inria)	11.00	15
Postcript	Edition Mars 1990 - 119 pages Par Gilles Dauphin (Telecom Paris)	11.00	15
Langage C++	Edition Mars 1990 - 77 pages Par Frédéric Lung (Consultant)	11.00	15
UNIX Systéme V administration	Edition Mars 1990 - 160 pages Par Michel Wurtz (Institut Géographique National)	11.00	15

UKUUG Publications available from EurOpen

	£	ECU
UNIX — The Legend Evolves 1990 Summer Proceedings, Royal Lancaster Hotel, London	30.00	44

UniForum Publications available from EurOpen

		ECU
Your Guide to POSIX	7.00	10
POSIX Explored: System Interface	7.00	10
POSIX Update: Shell and Utilities	7.00	10
Network Substrata	7.00	10
Network Applications	7.00	10
1991 UNIX Products Directory	35.00	50
International Editions of CommUNIXations	Price on application from Owles Hall	

NLUUG Publications available from EurOpen

		£	ECU
UNIX en Standaardisatie	November 1988	17.00	25
UNIX & Mens-Machine Interaktie	May 1989	17.00	25
UNIX & Connectivity	November 1989	17.00	25
Systeem Beheer "Gaan Open Systemen enVeiligheid samen?"	May 1991	17.00	25

ORDER FORM FOR PUBLICATIONS AND SOFTWARE DISTRIBUTIONS

This page may be photocopied for use. Please print!

Name: _____

Company name: _____

Address: _____

_____

_____

E-mail address: _____

I would like to order the following publications: _____

_____

_____

I would like to order the following tapes: _____

_____

_____

Please note that for distributions D1, D2 and D4 a copy of your source licence agreement with AT&T for at least UNIX version 7 should be enclosed. Note also that you have to be an EurOpen member (or a member of a national UUG) to obtain tapes at list prices. Non-members will have to pay DFl 300,- per tape extra as handling fee. Please enclose a copy of your membership or contribution payment form when ordering. Do not send any money or cheques, you will be invoiced.

All 1/2", 9-track, reel tapes come in tar format, 1600 bpi. 800 bpi is possible on request. Cartridge tapes come in tar format, written with dd, with a blocking of 126b. This is a so-called QIC-24 format, written on a Sun. QIC-11 is available on request.

Tape format, either 1/2" 9-track, or 1/4" cartridge: _____

Copy of EurOpen (or national UUG) membership (or payment) form enclosed? Yes / No

Copy of AT&T source licence enclosed? (For D1, D2, D4.) Yes / No

"I declare to indemnify the European Forum for Open Systems for any liability concerning the rights to this software, and I accept that EurOpen takes no responsibilities concerning the contents and proper function of the software."

Date: _____ Signature: _____

Please see payment details on next page.

*I enclose my payment in the sum of £_____ /I have completed my credit card details on next page

---

* delete as applicable

PAYMENTS

EurOpen is a European Federation of National Groups and as such prefers to use the European Currency - ECUs - for payments. To pay in ECUs please note the two methods of payment.

1. By Direct Payment to EurOpen's ECU Bank Account

The Bank of Scotland
International Division
Operations Dept
PO Box 86 120
St. Vincent St.
Glasgow G2 5DZ
Scotland

Account Number: 41791 ECU 01
Bank Sort Code: 80-20-13

Please tell your bank that you will pay all charges so that EurOpen will receive the full amount.

2. By UK Cheque or Bankers' Draft, made payable to EurOpen and drawn on a UK bank. Eurocheques are acceptable, but each cheque must be for 170 ECUs or less.

EurOpen is setting up facilities in Europe for delegates paying by Credit Card in ECUs, but unfortunately this is a vey new facility and not fully functional at the time of printing this newsletter. If you have real difficulties in paying in ECUs you may pay in £ sterling using the following methods of payment:

1. By Direct Payment to EurOpen's bank. Please tell your bank that you will pay all charges so that EurOpen will receive the full ammount due. The Bank of Scotland Account Number: 00613997 61 Grassmarket Bank Sort Code: 80-31-50 Edinburgh Scotland EH1 2JF

2. ☐ By VISA ☐ by ACCESS/EUROCARD/MASTERCARD

Name as it appears on the card (block capitals) _____

Address of card holder _____

_____

Card Account No: _____ Date of Expiry: _____

Signed: _____ Date: _____

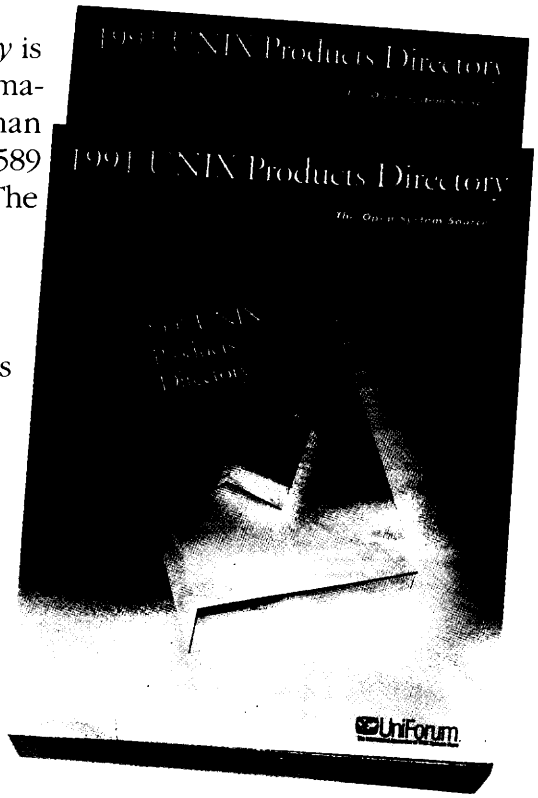
YOUR CONNECTION TO THE UNIX SOURCE

UniForum's *1991 UNIX® Products Directory* is the source for UNIX systems market information. This two-volume set includes more than 6,600 UNIX-specific products and services from 1,589 companies (including 299 international vendors). The 1991 edition directs you to more than:

- 3,500 Vertical and Horizontal Software Packages
- 1,266 System Software and Development Tools
- 647 Hardware Systems from PCs to Supercomputers
- 285 Peripherals from Modems to Printers
- 272 Consultants, including 79 Application Software Designers
- 289 Publications

Order UniForum's *1991 UNIX Products Directory* today and connect with the source.

The Directory is available from EurOpen (see Publications Order Form for details)



FREE Editorial Listings in the 1992 Edition

Vendors interested in having their products and services included in the 1992 edition of the UNIX Products Directory should contact the UniForum Association now for an entry form. To be published January 1992.

Beat the 16 August 1991 Deadline!

Contact:

2901 Tasman Drive
Suite 201
Santa Clara
California 95054
USA

Telephone +1 408 986 8840, ext. 16
Facsimile +1 408 986 1645

Email kathy@uniforum.org.

Calendar of UNIX Events

This is a combined calendar of planned conferences, workshops, or standards meetings related to the UNIX operating system. The information here is collected by those listed below after an idea by John S. Quarterman of Texas Internet Consulting. The information comes from the various conference organizers, ;login:, Communications of the ACM, CommUNIXations, and many others. We encourage others to reuse this information, but we ask for proper acknowledgment, for example by including this statement.

If you have a UNIX related event that you wish to publicise then contact either John S Quarterman at *jsq@tic.com*, Alain Williams at *addw@phcomp.co.uk*, Susanne W Smith at *sws@calvin.wa.com*, or Carolyn Carr at *carolyn@usenix.org* giving brief details in the style that follows.

Abbreviations:

APP	Application Portability Profile
C	Conference or Center
CC	Computer Communication
G, MD	Gaithersburg, Maryland
GM	Generam Meeting
LISA	Large Installation System Administration
MHS	Message Handling Systems & Application Layer CommunicationProtocols
OSE	Open Systems Environment
S	Symposium
SEDMS	Symposium on Experiences with Distributed and MultiprocessorSystems
T	Tradeshow
U	UNIX
UG	User Group
W	Workshop

1991

June	IUUG Motif, *Dublin, Ireland*
June 3	DKUUG C, Communications and Networks, *Denmark*
June 4	DKUUG C, System development and CASE ,*Denmark*
June 10-14	USENIX, *Opryland, Nashville, Tennessee, USA*
June 16-19	Sun User Group,*Atlanta, Georgia, USA*
June 17-20	INET 91', *Copenhagen, Denmark*
June 20	DKUUG C Standardisation,*Denmark*
June 24-26	TC 118N, *UniForum, Toronto, ON*
June 25-26	PortSoft, *(very tentative)*
July	JUS Symposium,*J US, Tokyo, Japan*
July 8-12	IEEE 1003 ,*Santa Clara, California, USA*
July 15-17	UKUUG C, *Liverpool, UK*
August 5-8	Interex C, *SanDiego, CA*
August 29	DKUUG C, The UNIX market - a snapshot,*Denmark*
September 10-12	European Sun User Group CT,*NEC, Birmingham, UK*
September 11	IUUG C, Network Applications Support (NAS),*Dublin, Eire*
September 16-20	EurOpen,*Budapest, Hungary*
September 24-27	AUUG CT, *Darling Harbour, Sydney, Australia*
September 26	DKUUG C, Publishing and image processing,*Denmark*
September 26-28	GUUG C & T,*Wiesbaden, Germany*
October	GUUG W, EUnet,*Dortmund, Germany*
October 16	IUUG C, C++,*Dublin, Eire*
October 10-11	Multi-User C Show, *UniForum Canada, Montreal, Quebec*

October 21-25	IEEE 1003, *Parsippany, New Jersey, USA*
October 30	IEEE CS SCC/SAB,*Nashville, Tennessee, USA*
October 31	DKUUG C, Client-server solutions,*Denmark*
October 31	Sun UG-NL,*Netherlands*
November 6-8	ISO/IEC JTC1 SC22 WG15,*Stockholm, Sweden*
November 7	User Friendliness C, T NLUUG, *Ede, The Netherlands*
November 13	IUUG C, Postscript Extensions,
November 14	APP/OSE Users Forum, *NIST, G, Maryland, USA*
November 27-29	AFUU C, Journees UNIX De Grenoble, *Grenoble, France*
November 28	DKUUG C, "Open" systems - how open? ,*Denmark*
December	JUS UNIX Fair,*Tokyo, Japan*
December 11	IUUG C, Large Systems Administration,*Dublin, Eire*
December 16-18	UKUUG C,*Edinburgh, Scotland*
December 9-11	Sun User Group,*San Jose, California, USA*
December 9-13	DECUS S,*Anaheim, California, USA*

1992

January 13-17	IEEE 1003,*Orlando, Florida, USA (location tentative)*
January 20-24	USENIX,*Hilton Square, San Francisco, California, USA*
January 20-24	UniForum, *Moscone Center, San Francisco, California, USA*
February 6	DKUUG C, Executive information systems, *Denmark*
March 11-18	CeBIT 92, *Hannover, Germany*
May 24-27	AFUU, Convention UNIX 92,*CNIT, Paris-La Defense, France*
April 6-10	IEEE 1003,*Dallas, Texas, USA (location tentative)*
April 13-17	EurOpen,*Jersey, UK*
May 4-8	DECUS S,*Atlanta, Georgia, USA*
May 18-22	ISO/IEC JTC1 SC22 WG15,*New Zealand (tentative)*
Summer	UKUUG C, *Queens University ,Belfast, NorthernIreland*
June 8-12	USENIX,*Marriott, San Antonio, Texas, USA*
June 21-24	Sun Users Group,*Washington DC, USA*
July 13-17	IEEE 1003,*Alaska, USA (location tentative)*
Sept 8-11	AUUG C T ,*World Congress Centre, Melbourne, Australia*
Autumn	ISO/IEC JTC1 SC22 WG15, *Denmark*
November 25-29	EurOpen/UniForum, *Amsterdam, Netherlands*
October 19-23	IEEE 1003,*Scottsdale, Arizona, USA (location tentative)*
October 19-23	IEEE 1003, *Southern Europe (location tentative)*
December	UKUUG/UKnet, *Manchester, UK*

1993

January 25-29	USENIX,*Town & Country, San Diego, California, USA*
March 15-18	UniForum,*Moscone Center, San Francisco, California, USA*
March 24-31	CeBIT 93,*Hannover, Germany*
April 26-30	EurOpen ,*Southern, Spain (tentative)*
June 21-25	USENIX ,*Cincinnati, Ohio, USA*
Autumn	EurOpen/UniForum, *Utrecht, The Netherlands*

1994

January 17-21	USENIX, *Hilton, San Francisco, California, USA*
February 7-10	UniForum, *Dallas Convention Center, Dallas, Texas, USA*
March 16-23	CeBIT 94, *Hannover, Germany*
April 18-22	EurOpen, *Switzerland (tentative)*
June 6-10	USENIX ,*Boston, Massachusetts, USA*
Autumn	EurOpen/UniForum, *Utrecht, The Netherlands*

1995

March 6-9	UniForum ,*Dallas Convention Center, Dallas, Texas, USA*
May 1-5	EurOpen ,*Scotland (tentative)*

1996

March 11-14	UniForum, *Moscone Center, San Francisco, California, US*

1997

March 10-13	UniForum, *Moscone Center, San Francisco, California, USA*

Organising Bodies

NIST/NBS/POSIX
Roger Martin
National Institute of Standards and Technology
Technology Building, Room B266
Gaithersburg, MD 20899, USA

Telephone +1-301-975-3295

Email rmartin@swe.icst.nbs.gov

IEEE Computer Society
P.O. Box 80452
Worldway Postal Center
Los Angeles, CA 90080, USA

Telephone +1-202-371-0101

UniForum (was /usr/group)
2901 Tasman Drive
Suite 201
Santa Clara CA 95054, USA

Telephone +1 408 986 8840
Facsimile +1 408 986 1645

./usr/group/cdn
241 Gamma St.
Etobicoke, Ontario M8W 4G7
Canada

Telephone +1-416-259-8122

Tracy MacIntyre
Exhibition Manager
EMAP International Exhibitions Ltd.
Abbot's Cour-t
34 Farringdon Lane
London EC1R 3AU
United Kingdom

Telephone +44-1-404-4844

AUUG Inc
P.O. Box 366
Kensington
N.S.W.2033
Australia

Email uunet!munnari!auug
auug@munnari.oz.au

Telephone +61 2 361 5994
Facsimile +61 2 332 4066

AMIX, c/o IPA
P.O. Box 919
Ramat-Gan
Israel, 52109

Telephone +972-3-715770
Facsimile +972-3-715772

Email amix@bimacs.bitnet
amix@bimacs.biu.ac.il

Japan UNIX Society (JUS)
#505 Towa-Hanzomon Corp. Bldg.
2-12 Hayabusa-cho
Chiyoda-ku, Tokyo 102
Japan

Email bod%jus.junet@uunet.uu.net

Telephone +81-3-234-5058

UNIX Fair '88 Association
1-1-1 Hirakawa-chu,
Chiyoda-ku, Tokyo 102
Japan

Singapore Unix Association - Sinix
20 Bideford Road #11-05
Wellington Building
Singapore 0922

Telephone +65 734 3256

DECUS U.S. Chapter
219 Boston Post Road, BP02
Marlboro, Massachusetts 01752-1850
USA

Telephone +1-617-480-3418

DECUS Europe
1-3, chemin Anneville, Box 176
CH-1213 Petit-Lancy 1
Switzerland

Telephone +41 - 22 - 709 42 64
Facsimile +41 - 22 - 792 25 03

DECUS Munich (for Germany, Austria, Hungary):
DECUS Muenchen e.V.
Freischuetzstr. 91
D-8000 Muenchen 81
Germany

Telephone +49 - 89 - 95 91 - 44 30

USENIX Association Office
2560 Ninth St., Suite 215
Berkeley, CA 94710
USA

Telephone +1 415 528 8649

Email office@usenix.uucp

National Expositions Co., Inc. (UNIX EXPO)
15 West 39th Street
New York, NY 10018
USA

Telephone +1-212-391-9111
Facsimile +1-212-819-0755

Sun UK User Group
Sue Crozier
Sun Microsystems, UK

Telephone +44 276 20980

Sun User Group the Netherlands (SUG-NL)
Clementine Voest
Sun Microsystems Nederland

Telephone +31 33 501234

Sun User Group, Inc.
Peter H. Salus
PO Box 167
Cambridge, MA 02142
USA

Telephone +1 617 739-0202

Email peter@uunet.uu.net

USING
P.O. Box 1077
Lisle,
Illinois 60532, USA

UniForum NZ Secretariat
PO Box 585
Hamilton
New Zealand

TRUUG
Esref ADALI,
Professor of Control and Computer Engineering,
TRUUG UNIX '90 Chairman,
Istanbul Technical University
Ayazaga
Istanbul
Turkey

Telephone +90 1 176 3586

EurOpen National group addresses can be found at the back of this newsletter.

Here is a list of acronyms that you might find useful:

ACE	Advanced Computing Environments
ACM	Association for Computing Machinery
AFUU	The Association Française des Utilisateurs d'UNIX
AUUG	The Australian UNIX systems Users Group
DECUS	The Digital Equipment Computer Users Society
EurOpen	The European Forum for Open Systems, (was EUUG)
FNUG	Federation of NCR User Groups
GUUG	The German UNIX Systems User Group
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
Interex	The International Association of Hewlett-Packard Computer Users
JUS	Japan UNIX Society
MCNTI	Moscow International Center of Science and Technical Information
NCR UUG	NCR UNIX User Group, Inc.
NIST	The National Institute of Standards and Technology
NLUUG	The Netherlands UNIX Users Group
NSF	National Science Foundation
SAB	Standards Activities Board
SERC	NSF/Purdue/Forida Software Engineering Research Center
SUUG	Soviet UNIX Users' Group
Sinix	The Singapore UNIX Association
UKUUG	The United Kingdom Unix systems Users' Group
USENIX	The Professional and Technical UNIX Association
UniForum	The International Association of UNIX Systems Users

Call Doc Strange

Colston Sanger
GID Ltd
United Kingdom

Email doc.strange@gid.co.uk

Colston Sanger is a senior consultant with GID and a visiting lecturer in the Faculty of Engineering, Science and Mathematics at Middlesex Polytechnic. His book, Open Systems for Europe: towards 1992, which he has been trailing in his biog here for the last year, has finally been published by Chapman & Hall.

NB: He hates the new-look EurOpen Newsletter, and 9-Point Gill Sans in particular.

Software Configuration Management for Beginners

New versions of RCS and `dmake` appeared in comp.sources recently, which got me thinking about the problem of software configuration management.

It's an awfully grand phrase: what exactly does it mean? Software configuration management has been defined as:

" the process of identifying the configuration items in a system, controlling the release and change of those items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items. It is a discipline applying technical and administrative direction and surveillance to (a) identify the functional and physical characteristics of a configuration item, (b) control changes to those characteristics, and (c) record and report change processing and implementation status. Major subtopics included in configuration management are: change control, configuration identification, configuration control, configuration status accounting and configuration audit[1]." (Phewh!)

1. Axel Mahler and Andreas Lampen, 'A toolkit for software configuration management' EUUG Spring Conference, London, 13-15 April 1988

The problem

Let's unpack some of that to see what it means in everyday terms. Wayne Babich in his book Software Configuration Management: Coordination for Team Productivity[2] provides lots of 'war stories':

- Marge is trying to find an obscure bug. She has narrowed down the problem to a particular section of code and has inserted all the necessary traces to find out what is happening. She is mystified because the source listing looks correct, but the values produced by the traces are all wrong. Finally, when she examines the load image with an octal debugger, she discovers that the program listing she's been using doesn't match the code that she's debugging. When she finds the correct listing, the bug is obvious.

Babich calls this the 'What Program is This?' problem, otherwise known as the 'This Worked Yesterday... What Happened?' problem.

Then there's the 'Double Maintenance' problem:

- Monopolistic Industries has automated its payroll system and is now beginning on its accounts payable. In 1983, Eric Promotable, chief programmer of the accounts payable project, discovers a utility package in the payroll system that formats, dates, and records cheques, drives the special forms printer, and prepares for reconciliation with the bank statement. The software is fairly complicated and not trivial to build again, so Eric copies it for use in his own project. The accounts payable software now includes exactly the same code that is used in the payroll system, and Eric has saved himself four weeks' work.

  Next year, the accounts payable system is successfully delivered and in use. On 29 February, as the company prepares its end-of-month payment of bills, the computer operator is horrified to discover that all the cheques coming off the printer are dated 0 March 1984. The cheque-formatting utility doesn't understand leap years. It takes more than a day to find and fix

2. Addison-Wesley, 1986. ISBN: 0-201-10161-0.

the error. The bug is in the code that Eric copied from the payroll system.

Some questions on this:

Boing!

Who will remember to fix the bug in the payroll systems's copy of the cheque-formatter?

Boing!

How likely is it that the fix that is (eventually) made to the payroll system will be exactly the same fix that is made to the accounts payable system?

The 'Shared Data' problem arises when several programmers are simultaneously accessing and modifying the same code. At least there is now only one copy of the code, but the problem is that changes made by one programmer can interfere with the progress of others. The obvious case is when a modification is plain wrong or contains a typo that crashes the whole system, but even a correct modification can give rise to unpleasant surprises elsewhere.

A common solution to the shared data problem is to divide the source code into a number of modules. A programmer who wants to modify a module makes a private copy of it and works on the copy. Common sense isn't it? However, now we come to the 'Simultaneous Update' problem:

- Dan and Joan are both assigned to the enhancement of some database management software. They are aware of the 'Shared Data' problem, so to avoid disrupting the other members of the project team, they decide that they should both make private copies of the modules they are working on...

  Joan is making a change to the representation size of integers, which requires a number of small changes to a variety of subroutines, so she copies the twenty subroutines she needs from the shared project software. She will be working on them for more than a week because of all the testing involved. At the end of the week, she will copy the new versions of the subroutines back into the project's shared copy.

  Meanwhile, Dan is assigned to fix a number of little bugs. He makes a private copy of a subroutine, fixes it, tests it, and copies it back - all in the space of a couple of hours. He fixes two or three bugs a day.

  The problem arises when Dan fixes a bug that coincidentally happens to be in one of Joan's twenty subroutines. Dan fixes the bug on Wednesday; on Friday, Joan copies back her twenty fully-tested subroutines and, in so doing, inadvertently overwrites Dan's bug-fix. The bug that Dan fixed on Wednesday is now unfixed again.

Any of this sound familiar?

UNIX to the rescue

The UNIX system, as befits its history as a 'programmer's workbench', has had tools for software configuration management for a long time. The grandpa of them all is probably *make*, These days, *make* is available in at least three versions (the original, SV 'Augmented make', and the BSD version), has been widely-ported to other operating systems (including MS-DOS, if that counts) and has spawned a whole host of make-oids: *mk*, *cake*, *dmake*, *Imake*, *shape*, *Pmak* and *qef*, to name but a few.

Loosely speaking, *make* takes care of the software build process, whereas version control of pre- and post-release code is covered by *SCCS* (in SV) and *RCS* (in BSD). Since this column is aimed pretty much at beginners (including myself) who usually have available only the 'SV - Consider it Standard' tools, I propose to introduce only the SV *make* here and, since space is limited, to tackle *SCCS* in a later column.

Using *make*

Compiling a simple, self-contained C program under UNIX is straightforward. It can be as easy as typing a command line like:

```
$ cc -O -o simple simple.c
```

It's not quite so easy, though, when you have to build a program that's split into several source modules. Of course, it's possible to write a shell script to do the job, but it's far better to use make. With *make*, you need only specify what has to be done to build a program; *make* then takes care of the details.

For example, imagine you are writing yet another editor. (All real programmers write at least one editor!) Yours is called - I don't know, how about *svi* for 'son of vi' - and its source modules live in the directory $HOME/src/svi. Here they are:

```
$ cd $HOME/src/svi ; ls -C
Makefile buffers.c end.c svi.c begin.c
commands.c insert.c svi.h
```

Notice the file called Makefile, which contains the specification for building the program. The contents of Makefile are:

```
#
# makefile for svi
#
svi: svi.o begin.o commands.o insert.o \
       buffers.o end.o
       cc -o svi svi.o begin.o commands.o \
       insert.o buffers.o end.o -lcurses

commands.o: svi.h insert.c
```

All you need do now is type:

```
$ make
cc -O -c svi.c
cc -O -c begin.c
cc -O -c insert.c
cc -O -c commands.c
cc -O -c buffers.c
cc -O -c end.c
cc -o svi svi.o begin.o commands.o insert.c b
uffers.o end.o -lcurses $
```

As you can see, make compiles each source module, then links the resulting objects to build *svi*. But there's more. If you now repeat the command, make doesn't execute the same lengthy series of compilations. Instead, it simply prints a message:

```
$ make
'svi' is up to date.
```

The specification contained in Makefile guides *make* to build a tree of inter-file dependencies. Then, by examining the last-modified time of each file, *make* finds out which files are older than one or more of their dependents. Any out- of-date files are rebuilt according to the specification in *Makefile*.

What happens if you change one of the source modules? For example, suppose you edit commands.c and then run make again:

```
$ make
cc -O -c commands.c
cc -o svi svi.o begin.o commands.o insert.o b
uffers.o end.o -lcurses
```

In this case, make does the minimum amount of work necessary to build a new version of *svi*.

Using make speeds up the process of building a new version of a program. If you change one of the source modules, make does only a fraction of the work that would be required for a complete

rebuild. Also, since the interdependence of source modules is taken into account, it is not possible for configuration errors to occur - unless, of course, the makefile itself is at fault.

Makefiles

By default, make looks for a file called makefile or Makefile in the current directory. A makefile contains a series of rules that consist of a dependency line and one or more action lines.

The first type of line (the dependency line, which describes what depends on what) consists of the target name, followed by a colon (:), optionally followed by a tab, and then a blank-separated list of files that the target depends on. For example, the dependency line

```
svi: svi.o begin.o commands.o insert.o \
        buffers.o end.o
```

says that *svi* (the target) depends on `svi.o`, `begin.o`, `commands.o`, `insert.o`, `buffers.o` and `end.o`. Dependents can be the targets of other rules or the names of files to be examined by make to see that their last-modified times tally.

The second type of line (the action line) tells make what commands must be executed to build the target once all the target's dependents are up-to-date. Action lines consist of an initial tab followed by any shell command. (Action lines must begin with a tab, not spaces. make will get confused if there are spaces at the beginning of an action line.) You can continue an action line over more than one line by escaping the end of line with a slash (\ ).

The # sign introduces a comment. All characters from the # to the end of the current line are ignored.

Implicit rules

Take another look at the second rule in Makefile:

```
commands.o: svi.h insert.c
```

There is no action line to tell make how to build the target object file `commands.o`. And yet make seems able to build it correctly. How is this possible?

It's possible because make has implicit rules. For each set of suffixes, .c, .f, etc., make has a special internal rule defining the commands that must be executed to transform files of these types. For example, make has an implicit rule that tells it how to transform a .c file into the corresponding `.o` file.

If you have a self-contained C program, such as `love.c`, lying around, you can test make's implicit rules for yourself. Try typing

```
$ make love
Make: Don't know how to make love. Stop.[3]
```

Oops! Sorry, that should be - if you have a self-contained C program, such as `hello.c`:

```
$ make hello
```

make will compile and link hello from `hello.c`. You don't even need a makefile.

Macros

If you want to, you can define macros within a makefile. make macros are similar, but not identical to, shell variables. For example, you could rewrite the makefile for *svi* as:

```
#
# makefile for svi
#
TARGET = svi
OBJS = svi.o begin.o commands.o insert.o \
        buffers.o end.o

HDRS = svi.h
LDFLAGS = -lcurses

$(TARGET): $(OBJS)
    $(CC) -o $(TARGET) $(OBJS) $(LDFLAGS)

commands.o: $(HDRS) insert.c
```

To define a macro, type:

```
MACRO_NAME = any string of characters
```

You can then refer to the macro as:

```
$(MACRO_NAME)
```

Certain macros - such as CC and CFLAGS - are built-in and have default values. For example, the default value of CC is cc (the C compiler), and the default value of CFLAGS is -O (Optimise).

Some other internally-defined macros are:

$*	the target name with the suffix deleted
$@	the full target name
$<	the complete list of targets upon which **$@** depends
$?	the list of dependents that are out of date.

For instance, the internal rule that enables make to build hello from hello.c is:

```
.c:
    $(CC) $(CFLAGS) $(LDFLAGS) $< -o $@
```

Pseudo targets

As well as macros, you can define pseudo targets in makefiles. One that's commonly used is install (as in install the compiled program in the correct directory, with the correct permissions set). Two other traditional targets are clean and clobber, used as follows:

```
$ make clean
```

or

```
$ make clobber
```

Here is the makefile for *svi* with these and other pseudo targets added:

```
#
#
makefile for svi
#
SHELL = /bin/sh
CC = /bin/cc
CFLAGS = -O
TARGET = svi
SRCS = svi.c begin.c commands.c insert.c
buffers.c end.c
OBJS = svi.o begin.o commands.o insert.o
buffers.o end.o
HDRS = svi.h
LDFLAGS = -lcurses
LINT = /usr/bin/lint -p
DESTDIR = /usr/local
TEXT = $(SRCS) $(HDRS)
LP = lp -dpr1
LASER = lp -otroff -dpr2

$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -Ksd -s -o $(TARGET) \
          $(OBJS) $(LDFLAGS)

commands.o: $(HDRS) insert.c

lint:
```

3. Which is why UNIX is called UNIX ...

```
        $(LINT) $(SRCS)

prof:   $(SRCS) $(HDRS)
        $(CC) $(CFLAGS) -p -c $(SRCS)
        $(CC) $(CFLAGS) -p -o $(TARGET) \
        $(OBJS) $(LDFLAGS)

print:
        pr $(TEXT) | $(LP)

install:
        $(TARGET)
        cp $(TARGET) $(DESTDIR)
        chgrp bin $(DESTDIR)/$(TARGET)
        chown bin $(DESTDIR)/$(TARGET)

doc:
        troff -man $(TARGET).1 | $(LASER)

clean:
        rm -f *.o core a.out mon.out

clobber: clean
        rm -f $(TARGET)
```

Make_options

Like many UNIX commands, *make* has a wealth of options, most of which are used only occasionally. However, three of them are worth mentioning. First, the -n option, which tells make to scan the makefile looking for dependencies, but then only to print the commands to be executed (instead of actually executing them). This is particularly useful for testing new makefiles or for estimating how long a rebuild is likely to take.

Second, the -k option. Usually, make terminates as soon as a command returns a non-zero exit status. With the -k option, however, make carries on for as long as possible before giving up. This option is very useful when you are building a program for the first time. For example, suppose that the file insert.c contains a syntax error. Typing just make results in the following:

```
$ make
        cc -O -c svi.c
        cc -O -c begin.c
        cc -O -c insert.c
"insert.c", line 2: warning: ambiguous
assignment:
simple assign, unary op assumed
"insert.c", line 2: declared argument ch is
missing
"insert.c", line 2: syntax error
*** Error code 1
Stop.
```

With the -k option, make gets a lot further:

```
$ make -k
        cc -O -c svi.c
        cc -O -c begin.c
        cc -O -c insert.c
"insert.c", line 2: warning: ambiguous
assignment:
simple assign, unary op assumed
"insert.c", line 2: declared argument ch is
missing
"insert.c", line 2: syntax error
*** Error code 1
        cc -O -c buffers.c
        cc -O -c end.c
'svi' not remade because of errors
```

Third, the -s (silent) option - which is particularly useful if you are doing large 'makes' in the background.

Making the most of make

There is, of course, much more to *make*: you can use it for many other things as well as building software. For example, you could use make for document preparation. More generally, make is useful for packaging any set of operations that have time dependencies.

Make is wonderful, but...

make is wonderful, and it can do much more than the simple stuff I've described so far. However, it does have some shortcomings[4].

The first is the lack of 'standards' in using make. Despite being in use for nearly fifteen years now, no dominant style of using make has evolved. There are a number of traditional pseudo targets (such as clean or clobber), but it would be foolish indeed to blindly make clobber without looking closely at the makefile to see what clobber actually does.

Second is the 'overloading' of makefiles with all sorts of rules to beautify the source, run it through lint, print listings, troff the documentation, etc. as well as building and installing the software. make is convenient for doing all that - and there is no other suitable tool - but it does mean that makefiles can get very complicated. To be fair, it is possible to have a hierarchy of makefiles, as in this extract from the top-level makefile for the Elm mailer:

```
# Targets that are simply executed in each
# subordinate makefile as is.
#
all install uninstall lint clean:
        cd src; $(MAKE) -$(MAKEFLAGS) $@
        cd utils; $(MAKE) -$(MAKEFLAGS) $@
        cd filter; $(MAKE) -$(MAKEFLAGS) $@
        cd doc; $(MAKE) -$(MAKEFLAGS) $@
```

This is a sort of 'super-make': in which *make* itself runs make in each subordinate directory in turn.

Third is the difficulty of maintaining *makefiles*. Because makefiles tend to get overloaded with pseudo targets, maintaining them can be more difficult than it should be. Moreover, makefiles are, more often than not, hacked rather than designed: the starting point for the makefile for a new piece of software is likely to be a copy of the makefile for a previous piece of software. No harm in that, except that, over time and many generations of copies, the makefile is equally likely to become unstructured, and its correctness for the current software taken as an article of faith.

Fourth is the way in which make is only half a language: it has macros (variables), but it doesn't have control-of-flow constructs. For example, it would be useful to be able to select what to do next dynamically, depending on the result of some previous rule. (The new version of dmake seems to have this capability.)

Finally, is the fact that make is only useful for packaging a set of operations that can be expressed in terms of time dependency - but this is perhaps a weakness (Heaven forbid!) of the underlying UNIX filesystem and the limited set of file attributes that it chooses to store.

---

4. For a much fuller discussion, see D.M.Tilbrook and P.R.H.Place, 'Tools for the Maintenance and Installation of a Large Software Distribution', EUUG Spring Conference, Florence, 21-24 April 1986.

Standards Column

Henk Hesselink
ACE Associated Computer Experts bv
van Eeghenstraat 100
1071 GL Amsterdam
The Netherlands

Email henk@ace.nl

Henk Hesselink is one of the members of the EurOpen standards group and he is also on the board of the NLUUG. His interests include CASE environments, user interfaces and networking. His cats are named Alcatraz and Beelzebub.

At the last Governing Board meeting it was decided that EurOpen should take a more active role in standards work. To this end the Standards Activities Management Group was set up, and yes that's an incredible mouthful and you can't pronounce the acronym either.

The group is to coordinate EurOpen standards activities, such as Dominic Dunlop's ISO Monitor Project and participation in EWOS. It will also function as clearing house and contact point for information and for members' standardisation activities.

To give some idea of how EurOpen can participate in the various POSIX activities, first a little background: Within ISO/IEC, POSIX work takes place in Working Group 15, referred to as JTC1/SC22/WG15, or just WG15 when you run out of breath. What this means is that WG15 reports to SC22 (Programming Languages), which reports to JTC1, the technical committee in ISO responsible for IT standards. What it also seems to mean is that POSIX is a programming language. The interesting (for EurOpen) work is done at the level of the working groups and their internal rapporteur groups. Other interesting working groups include WG11 (language bindings) and WG14 (C).

Input to the working groups is from the National Member Bodies such as AFNOR, ANSI, BSI, DIN and the NNI. Only official delegates from the national member bodies can vote, though working group meetings can be attended with "observer" (non-voting) status. For POSIX, SC22 has asked a specific national member body (ANSI) to produce the standards documents. ANSI has delegated this to the IEEE, which has set up its Technical Committee on Operating Systems (TCOS). Note that this does not mean that IEEE TCOS defines the final OSI standard(s): it is producing documents based on its 1003.x activities, but WG15 can and does request modifications and withhold approval before these documents are accepted as ISO standards.

Another source of input to ISO is via CEN/CENELEC. Residing under CEN is the European Workshop on Open Systems (EWOS) which concerns itself with OSI networking. Some time around 1989 CEN proposed adopting X/Open's XPG3 as a European standard. This proposal was rejected for a number of reasons, some political (e.g. who gets to manage the document), some technical (a standard cannot contain another standard and XPG3 for instance contains Cobol, for which there is already a standard). CEN then asked EWOS to expand its work to include Open Systems which resulted in the installation of the Expert Group on Common Application Environment (EG-CAE).

EurOpen now has Institutional Representative status (this means we're allowed to speak) on IEEE TCOS, and participates in EWOS EG-CAE meetings (where you don't need any particular status to speak). It is also possible that, at a later stage, EurOpen may be able directly to participate in WG15 work.

Dominic Dunlop, who attended Chicago's April meeting of the IEEE POSIX working group as EurOpen's Institutional Representative (IR), adds:

At the April meeting, the IEEE Computer Society's Technical Committee on Operating Systems (TCOS), which develops the POSIX standards, demonstrated that, in a world where standards are political as much as technical tools, it can play politics as well as the next organization. The Open Software Foundation had put in a Project Authorization Request (PAR) for work on a standard

corresponding to its X Window-based Motif user interface; UNIX International had responded by bringing forward a PAR for its answer to the same problem, Open Look.

It is up to TCOS' Sponsor Executive Committee (SEC), composed mainly of chairs of existing standards working groups, to accept (sponsor) or to reject PARs. Clearly, whatever stance the SEC took, it was going to upset somebody:

- If it accepted both PARs, the result would be competing standards in similar subject areas -- something that TCOS has so far managed to avoid. What is more, both would overlap the work on windowing systems taking place in the 1201 working groups.
- If one proposal were accepted, but not the other, the IEEE would be open to potentially highly damaging accusations that it was favouring one technology above another in the marketplace[1].
- If neither were accepted, TCOS could be accused of being unresponsive to the needs of the marketplace for formal standards in the area of windowing interfaces.

After three hours of inconclusive debate at a committee meeting which showed signs of continuing past midnight, procedural manoeuvres by the EurOpen and USENIX institutional representatives resulted in two quickly-taken votes[2]: first, the Motif PAR was not accepted, and then the SEC voted by a wide margin not to sponsor either PAR "at this time".

This leaves open the possibility that one or both of the proposals may come back before the SEC in the future. But in the mean time, the 1201.1 and 1201.2 working groups respectively will continue working on a "Layered Application Programming Interface" and a "Driveability Guide", which aim to address the issues of a common programming interface to windowing systems, and of a consistent user interface. Hopefully, this work will be successful, and will result in standards which are both acceptable and widely accepted. But if it does not, and if Motif, Open Look, or some other "look and feel" becomes accepted as predominant, yet is ignored by TCOS, and opportunity will have been missed, giving ammunition to those who charge that the standards process, cannot deliver fast enough or efficiently enough, and so should be replaced by something else. Just as soon as they've worked out what it is.

---

1. There is a move afoot to rescind IR's voting rights. Needless to say, IRs are fighting to block it. If we hadn't used our privilege on this occasion, TCOS might still be locked in debate...

2. Inevitably, both O.S.F. and UI can quote figures which suggest that their respective technologies are market leaders in some way or another.

PCs – A Window onto UNIX

Stephen Purdham
Sales Director
JSB Computer Systems Ltd
United Kingdom

Steve Purdham, Sales Director, joined JSB in 1983. He has over fourteen years experience in the computer industry which has given him a valuable edge in ensuring that JSB's sales objectives are always achieved.

Steve has been responsible for setting up a European distribution netowrk for JSB and developing JSB's successful marketing strategy.

He started his computing career at ICL in its Finance Sector where he was acknowledged as one of the leading pre-sales support executives.

With the battle for which devices should become the desktop workstation, Stephen Purdham, Sales Director of JSB Computer Systems Ltd, looks at DOS based PCs and reports how DOS and UNIX integration is an ideal opportunity to get the best from both environments.

The fight to get workstations with the latest GUI environments onto peoples desks is turning out to be one of the hotest battles in computing history. With vendors forming unexpected alliances (or standards bodies) to ensure their particular technologies has the greatest chance of general acceptance. Unfortunately, most of these battles are being fought within the confines of the UNIX theatre of war and all sides have seemed to have missed two major details which may cause all sides to be in a no-win situation.

Firstly, what are end users actually trying to do?

and secondly

What do end users already have sitting on their desks in vasts numbers?

The answers to both these questions are quite simple. The end user basically is trying to do a job and with a forecast of 12 million DOS PCs to be sold this year, it is not difficult to see what end users are putting on their desks. If we assume that the PC is winning the hardware battle for the desktop then it is justifiable to predict that the windows environment that will be used must be MS Windows 3. this prediction will win no awards since sales of 3 million copies of MS Windows 3 were announced in the first six months from launch.

So if PC's DOS and MS Windows are a fact then let's look at how it is possible to capitalise on the available technologies so that the DOS and UNIX worlds can be integrated into a seamless environment.

We have already stated that the end user basically needs to do a job, and the concept behind DOS/UNIX integration is to deliver the "right tool for the right job" in a manner which reflects the way a user works and not restrict it. DOS/UNIX connectivity, within an MS Windows environment, is simply the means by which we provide seamless integration so that the strengths of each environment are embraced under one umbrella, which means the user is then given a CHOICE.

To address this area in more detail we need to cover three areas:

- The User Interface
- Connectivity
- Integration

The User Interface

The choice of MS Windows 3 as the User Interface for the PC should not necessarily be taken at face value. When considering the user interface there are several basic elements with which the interface should comply:

- Warm and friendly
- Intuitive
- Common User Interface
- Graphical

Taking each in turn, the user interface should be *warm and friendly* - the user should not be apprehensive about using it. Going hand in hand with this the interface needs to be *intuitive* - the user shouldn't need a Degree in Astrophysics to be able to use it!

Applications should ideally conform to a *common user interface*. By providing a common user interface, fundamental learning capabilities can be migrated from application to application without the need for retraining. To do this the application would require modifying to adhere to a standard way of working, such as IBM's Common User Access from their SAA approach. Since most applications do not conform to a standard CUI then any offered interface should provide mechanisms to at least start and finish jobs in the same way and gain help in the same way, this would at least provide a minimum level of consistency.

Finally, the user interface should be *graphical*. The 90's has been hailed as the decade of the Graphical User Interface (GUI). this is not because GUI's look "nice" but studies in the States have proven that there is a significant increase in productivity when graphical applications are utilised. There is also a subsequent decrease in support and training overheads.

Due to its penetration and wide spread acceptance the graphical user interface, which delivers the required facilities for DOS PCs, is undoubtedly Microsoft Windows 3.

However, to cater for other environments in addition to DOS we need to use Microsoft Windows 3 to extend the boundaries of the DOS PC so that the user interface principles can be applied to UNIX (and possibly other environments such as VMS).

Products like JSB MultiView DeskTop provide this by allowing all applications to be represented by icons, so that invoking a remote UNIS task is as simple as starting a local DOS task. The PC looks like a window onto the rest of the mixed computer environment and in this way users can simply get on and run applications totally unaware of the underlying technology. In effect the user sees what he needs to see - that is the application and not the technology.

Connectivity

Once we open the boundaries of the PC we need to explore mechanisms for connectivity. To do this we must identify from the outset what we are trying to achieve and also from what point we are starting. The following questions are just some which need answering before a decision can be made on connectivity strategy for DOS and UNIX integration:

- Do I already have a PC LAN (e.g. Novell) and want to embrace the power of UNIX?
- Do I already have a UNIX network and want to embrace the power of DOS?
- Do I need to share resources such as laser printers and disk between DOS and UNIX environments?

the answers to these and many other questions are not always easy or immediately apparent.

I you want to gain access to the UNIX system to simply run the applications that the UNIX system provides then that can be done by using a simple RS232 connection or any "off the shelf" TCP/IP for PC packages. However, if you also want to embrace the UNIX system to share resources then you need to choose a much more specific network strategy such as SUN PC/NFS or Locus PC/I which allow the PCs on the network to access the UNIX printer and disk resources as if they were DOS devices.

If PC LANS already exist and need to be maintained then it is necessary to work out a strategy which allows for mixed network protocols to co-exist simultaneously. For example, in the case of Novell and UNIX networks how can IPX and TCP/IP protocols coexist? To satisfy this need then the choice of network hardware and software is vitally important. Multiprotocol cards will allow network software to coexist and additionally the emergence of Packet Drivers for network cards also allow some network cards to service multiple stacks.

Once the correct selection is made then it is possible to have a mixed network which allows the user to gain access to local PC services, PC LAN services and UNIX services. This area is becoming further complicated by the addition of PC LAN services to UNIX systems for example LM-X and Ported Netware.

Integration

The third areas looks at integration in terms of shared information between applications.

Simple cut and paste mechanisms between DOS and UNIX applications can provide a powerful tool to share information. For example, the user can mark text in a UNIX application, paste this to the Windows clipboard and then copy it to a DOS application.

Cut and Paste is satisfactory unless the information required is constantly changing. Within MS Windows integrating changing information can be catered for by providing Dynamic Data Exchange Links. by creating DDE links information can be constantly updated from application to application. A high level of integration is possible by creating fully integrated applications by adopting one of the various client/server approaches in the market.

It is a fact that most UNIX applications are character based. Taking integration a step further, the widespread acceptance of MS Windows 3 has lead to users demanding that all their applications should be graphical. Unfortunately the cost and timescales involved in retraining personnel to redevelop current character based UNIX applications into graphical applications is astronomical.

A simple but effective solution is to use Deskterm protocols developed by IXI Limited of Cambridge. It is possible to encapsulate a current application with the Deskterm protocol to provide a graphical "look and feel" with support for radio buttons, dialogue boxes, scroll bars, mouse motion, colours and fonts. Once your application supports the Deskterm protocols, you can visualise it on standard terminals, graphically under MultiView DeskTerm on MS Windows 3 or on an X Window terminal or workstation under IXI's X. deskterm. This type of mechanism provides for a fast and cost effective way of migrating current applications into the graphical world.

Providing cosmetic surgery in an MS Windows environment is an interim step which saves development time and money. However, an alternative to using Deskterm is to develop fully integrated applications using Client/Server technologies. This is where MS Windows based applications can be used or developed which have a communication link of some sort to a remote application. In this way the local windows based Client application is receiving data from the remote server based UNIX application. A good example of the type of technology is the SQL Client/Server mechanisms provided by Gnosis.

The idea of "Open Systems" is to provide a computing system which is flexible, scaleable, interoperable and portable and most importantly it must reflect the users working environment. The last step towards total integration is to provide a path for future technological developments - in the UNIX market today that means X.

In the DOS/UNIX integration arena the need to service the X applications has resulted in the development of the PC based X servers which are basically sophisticated terminal emulators. The X servers fall into two categories namely of X servers for DOS, such as Locus X-Sight, Hummingbird PC EXCEED and MS Windows based X servers such as Integrated Inference Machine's X11/AT, JSB MultiView X11/AT, Hummingbird PC EXCEED/W and Visionware's XVision. It is expected that MS Windows based X servers will be predominant in the market place within the next year.

With PCs and MS Windows 3 being predominant in the world market place, the raging battle within the UNIX camps for the prize of the most accepted GUI becomes almost insignificant and can really be looked upon as a minor local skirmish.

To conclude: MS Windows 3 and the ability to extend the boundaries of the PC, with products like JSB MultiView Desktop, means that the PC can become a window onto the rest of the computer world. This means that systems integrators can choose from a wide spectrum of applications, regardless of their location and environment. Which means the end user can be given the right tool for the right job.

JSB Computer Systems Ltd, *Telephone +44 625 433618*

IXI Ltd, *Telephone +44 223 462131*

Locus Corporation, *Telephone +1 213 670 6500*

GNOSIS, *Telephone +44 61 474 1963*

Hummingbird, *Telephone +1 416 470 1203*

Sun, *Telephone +44 203 692255*

Integrated Inference Machines, *Telephone +1 714 978 6776*

EWG News

Jean-Michel Cornu
Consultant
69 rue de Seine 91130 Ris Orangis
France

Email Jean-Michel.Cornu@afuu.fr
Email Jean-Michel.Cornu@enst.fr

Jean-Michel Cornu is a French consultant involved each time there is a group or a committee on the planet. That means that he participates to several users organisations and standard bodies, but has also created research Working Groups on fundamental physics and a worldwide amateur TV network. In EurOpen he proposed to coordinate the support to the creation of Working Groups.

This is a new column in the newsletter. You will have to remember a new acronym "EWG" which stand for EurOpen Working Groups. I will give you regular news on what is going on and what support we can provide to help you create WGs at national and/or EurOpen level. This is also an opportunity to present an individual Working Group and its results or ones which need a bit of extra help. I hope that this will give you the opportunity to get some direct contacts.

Few things that you can get from me:

- A presentation on how WGs may be organised in your country.
- A Guide on how to create a WG is nearly ready.
- A list of people interested in creating WGs on specific topics is maintained.

If you want some (or all) of them, please feel free to contact me or the EurOpen secretariat. I can also put you on the list of people who receive direct information on EWG (postal mails and ewg@dkuug.dk)

Jean-Michel Cornu EWG coordinator
69 rue de Seine 91130
Ris Orangis
France

Telephone +33 1 69 43 48 47
Facsimile +33 1 60 78 00 98

Email Jean-Michel.Cornu@afuu.fr or Jean-Michel.Cornu@enst.fr

Report on i2u Working Groups: Factory Automation

Fabio Coda Zabetta
i2u/SOI
Viale Monza 347
20126 Milano
ITALY

Telephone +39 2 2520 2530

Most of the people attending the meetings were management-level staff, with a small number of technical personnel. We are carrying on investigations within our own WG members, with special attention to:

- Information technology, within currently established sub-committees; these groups address highly technical problems for technical staff.
- which CIM Applications have already been developed in UNIX environments; when this information will be collected, a subsequent analysis will evidence:
- current state of the art for CIM applications in UNIX; (strategy analysis) we would like to make clear in which CIR areas/levels UNIX may already be used successfully today, and in which areas/levels the use of UNIX can be considered critical;
- (technical analysis) a report will be written where we will identify the technical issues that have driven the strategy analysis.
- Experience and/or technology transfer between the WG members, achieved by hosting sessions where CIM application based on UNIX are presented by WG members.

USLE Column

Morris Schwartz
Sales and Marketing Director
UNIX System Laboratories Europe
London
United Kingdom

Morris Schwartz joined AT&T's UNIX System Laboratories Europe (USLE) in January 1990 and has overall responsibility for commercial sales and marketing activities throughout Europe.

He was previously with UNIX International where, as Marketing Director, he launched UNIX International in Europe, quadrupled the membership during the year and planned the European launch of UNIX System V Release 4.

For further information on this column, please contact Gill Mogg on gill@uel.co.uk or +44 81 567 7711. Gill is Marketing Manager at USLE.

Multiprocessing – its importance in the '90s

Demands for processing power by end users are increasing at an unprecedented rate. In many respects, microprocessor technology is having an effect on almost all aspects of our everyday lives. A visit to the bank, a trip to the doctor, going shopping in a high street store – all these activities can involve a transaction being recorded, filed, transmitted or processed in some form.

The overall performance of computer systems related to these activities is determined by a number of factors:

- The complexity of the transaction in relation to the associated application software;
- the efficiency of the communications network
- the workload on the network, and
- the interface between the end user and the application (e.g. point of sale terminal, cash dispenser, graphical user interface, etc.)

All of these items create demands for superior performance.

Additionally, the development and increased use of Multi-user relational database environments can consume vast amounts of computing power. Client/Server-based architectures also require their own power to manage and control the network, even before they perform any useful function for the end-user. Increasingly sophisticated word processors, desk top publishing systems and large multi-dimensional spreadsheets, all contribute to an insatiable thirst for more processing power.

In this article I will discuss multiprocessing as a business issue in today's demanding commercial environment and provide a closer look at evolving UNIX® System V Release 4 (SVR4) multiprocessing applications and their benefits for the marketplace.

Price and Performance

The method by which hardware vendors have traditionally met this need for more computing power has been the sub-division of the marketplace. Over the last decade we have seen the emergence of three general categories of computer – the PC, the mini and the mainframe – which act as an indication of the 'Architectural thinking' over this period with each category of computer providing its own range of performance – but all mutually incompatible.

The IT industry has, however, learned that the biggest improvements in the price/performance relationship could best be achieved through the technological 'leapfrogs' in microprocessor technology. An IDC study places the cost of mainframe computing at $100K per MIP, mini-computer computing at $40K-$50K per MIP, while the cost of multiprocessing computing requires only $1K-$5K per MIP.

Perhaps the most significant new element introduced by multiprocessor architectures is a wholly different approach to cpu power. Previously, users had to exchange their existing computer hardware for new hardware if they needed to gain a 50 percent increase in computer power. In many cases this has also entailed migrating the software, with all its attendant headaches and enormous expense.

The multiprocessor approach, on the other hand, allows the user to 'grow' the system in increments of one cpu, simply by slotting in additional cpu boards. This provides a quantitative change in the way in which increased performance can be achieved.

There are a number of ways in which applications performance on multiprocessor-based systems can be improved. In a system running several applications concurrently, one of the most efficient methods is to allocate specific cpus to specific applications. More powerful systems may also offer a 'multi-threading' capability which allows individual applications to be executed across several different cpus. Both of these features will be progressively offered by USL. (Figure 1)

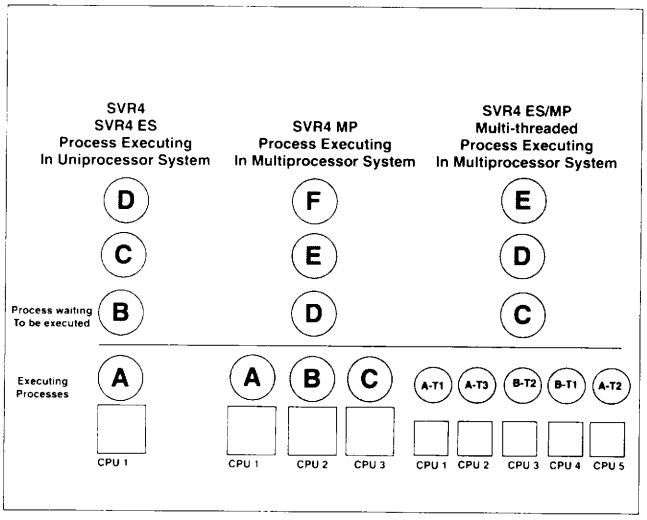


Figure 1:

Types of multiprocessing

At this stage, an important distinction should be drawn between symmetrical and asymmetrical multiprocessing. In a symmetric multiprocessing system, all processors have the same capabilities. It is therefore scalable to tens of processors with every single processor able to run the operating system and the applications. All applications have access to the same memory and I/O devices and can handle interrupts.

In contrast, in an asymmetric multiprocessing system the operating system runs on a designated master cpu making it very vulnerable: Asymmetric multiprocessing systems can easily hit a bottleneck when the operating system use increases, and if the master cpu fails, the entire system fails.

A symmetrical multiprocessing system therefore provides the user with significantly increased reliability. End-users are facing a commercial reality in which multi-microprocessors represent the next performance step forward.

USL Multiprocessing – delivery goals

USL is currently developing a symmetrical multiprocessing release with multi-threaded capabilities of UNIX System V Release 4. The plan is for a two-phased introduction as laid down in the UNIX International 1991 Roadmap. The first set of features is scheduled for general availability in mid-1991 and the second part of the strategy will be fulfilled in the latter half of 1992.

The goal is to provide a Multiprocessing release of SVR4 which has upward compatibility with today?s SVR4, is compliant with X/Open's® XPG3 standard and capable of supporting tens of processors. It will also comprise a secure base and a full multiprocessing application programming interface.

USL will supply programming algorithms targeted to improve performance of such business-critical applications as multi-user relational database applications and on-line transaction processing, involving high activity database servers.

1. Multiprocessing – Phase 1

To achieve this complex undertaking, while providing an early response to the clear demands for multiprocessing SVR4, USL is working with Intel, Olivetti, Motorola, NCR, OKI and Unisys to provide a release, designated UNIX System V Release 4 Multiprocessing (SVR4 MP) – available in mid-1991.

SVR4 MP will adhere to all relevant standards at the source code level including X/Open's Portability Guide, Issue 3 (XPG3), the System V Interface Definition (SVID) and the Device Driver Interface/Device Kernel Interface (DDI/DDK). It will also be ABI conformant at the binary level, thus ensuring binary applications portability. SVR4 MP will be available on the Intel 386/486(TM, Intel i860™ and in conjunction with Unisys and Motorola 88000 microprocessor architectures.

SVR4 MP will not only provide symmetrical multiprocessing but also additional capabilities:

- It will provide end users with a simple standard way to incorporate MP hardware into their information systems and migrate existing applications to multiprocessor systems.
- All System V compliant applications will be able to run unchanged on SVR4 MP.
- SVR4 MP will afford the user greater freedom of choice when making multiprocessor system purchases.
- The system will incorporate multi-threading of the base OS kernel, I/O and STREAMS systems, NFS, RFS and TCP/IP to ensure performance without bottlenecks.
- SVR4 MP will adhere to standards and will be ABI conformant at the binary level.

In addition, several vendors have implemented their own multiprocessing versions of the UNIX System for different architectures but in compliance with all SVR4 interfaces. They include ICL, Sun, Pyramid and AT&T Computer Systems, who have agreed to make their implementations compatible with the device driver interface, device kernel interface (DDI/DDK) and the STREAMS interface in SVR4 MP. As a result, independent software developers and end users can be sure that applications and device drivers written to standard interfaces will run without difficulty on different multiprocessing architectures.

2. Multiprocessing – Phase 2

In the second half of 1992, USL will introduce an advanced architecture multiprocessing release, designated UNIX System V Release 4 ES/MP (SVR4 ES/MP), which will be built upon an enhanced security base. This release will support full symmetrical multiprocessing on small, medium and large scale systems. SVR4 ES/MP will be architectured to scale over 30 parallel processors.

USL is developing the multiprocessing features of SVR4 ES/MP in conjunction with Sequent Computer Systems and other partners including AT&T Computer Systems, ICL, Intel, Motorola, NCR, Unisys, Fujitsu and Pyramid. Each company is contributing technology and/or development support for its creation.

Key features of SVR4 ES/MP include:

- a fully parallel, multiprocessing implementation of UNIX System V with a high level API,
- fine grained fully multi-threaded OS, scalable from one to tens of processors,
- compatibility with earlier releases of UNIX System V at both source and binary level,
- near linear performance improvement as processors are added,
- User-level Threads and User-level Synchronisation with the API which will allow developers to take advantage of

algorithmic parallelism from within applications,

- the highest level of security available for UNIX System V today,
- support for the latest XPG3, IEEE™ 1003.1a and 1003.2 standards, and enhanced international support,
- simultaneous execution of multiple threads in a single address space,
- a flexible debugger architecture with enhancements to support lightweight processes, and
- Operations and Administration Enhancements.

Summary

The addition of multiprocessing technology to UNIX System V will provide the next major advance in the price/performance ratio in the Open Systems market. USL has made compatibility between the evolving multiprocessing releases its top priority. Applications and device drivers written for SVR4 will be upwardly compatible through both SVR4 MP and SVR4 ES/MP.

UNIX International members will have early access to both SVR4 MP and SVR4 ES/MP commencing in the first and third quarters of 1991 respectively.

Customers who buy existing products today can be completely confident that they can migrate their applications to take advantage of emerging technological advances. This will enable them to meet the needs for greater processing power in fully distributed environments.

USENIX Association News for EurOpen Members

Donnalyn Frey
Frey Communications
Fairfax
Virginia
USA

Email donnalyn@frey.com

Donnalyn is the USENIX Association Press Liaison. She provides members of the press, USENIX Association members, and EurOpen members with information on the activities of the USENIX Association.

Summer 1991 Conference and Exhibition

The Summer 1991 Conference and Exhibition was held at the Opryland Hotel in Nashville, Tennessee on 10 - 14 June. The focus of the conference was Multimedia -- for Now and the Future. A report on this Conference and Exhibition will appear in the next issue.

C++ Conference

The Third USENIX C++ Conference was held in Washington, D.C., 22-25 April 1991. The first two days were reserved for a total of five tutorials, the remaining two days for the technical program. The conference was attended by 274 people. One attendee described the USENIX event as "the conference of C++ experts."

The technical program was quite strong and well received. The keynote address, by David DeWitt of the University of Wisconsin, titled "C++ + Persistence != An Object-Oriented DBMS," highlighted the difference in perspective between the typical programmer and the database community. The functionality required to provide a database system, including such things as atomic transactions and serialisability, are often more than the programmers want. Of course, there was much speculation before the talk as to whether the "+" and "!=" operators in the title had been overloaded.

Other topics covered by the technical program included proposed extensions to the language (all of which were interesting, but none received anything approaching a consensus of support), critiques of the interactions between different parts of the language, class library implementors experiences, and a panel debating the value of Multiple Inheritance. The panel did not reach any particular conclusion, being divided between proponents of MI, who argued that there are large systems in which it seems to be a useful abstraction mechanism, and the unconvinced, arguing that there have never been any published examples that make a compelling case. All did seem to agree that MI is a large addition to an already large language.

SEDM Symposium

The Second Symposium on Experiences with Distributed and Multiprocessor Systems was held on 20 - 21 March in Atlanta, Georgia. It was attended by 123 people from around the world.

Overall, it was a well received workshop. Attendees and speakers arrived from several different countries, including Japan and Australia. The papers were very forthcoming about things done wrong as well as those done right, concentrating on real experience, not conjecture.

Many knowledgeable and experienced people attended and it brought about many interesting discussions. The symposium was so successful that SEDMS-III is already planned for 1992.

USENIX Standards Liaison

At its January meeting, the USENIX Board of Directors authorised funds for this year for an Institutional Representative to the IEEE Computer Society Technical Committee on Operating Systems, and the U.S. Technical Advisory Group to WG15. The position was offered to Peter Collinson, of Hillside Systems, Kent, UK, and he accepted.

Peter has been active in UNIX since 1976, and has a strong technical background. As many of you know, his previous

employment was as an academic, and he has been active in EurOpen and USENIX for more than 10 years. He is currently a consultant, writer, and lecturer. His knowledge of USENIX and EurOpen, as well as his technical expertise should serve him well as the USENIX IR.

Besides his duties at IEEE/CS TCOS meetings, he will coordinate USENIX Standards BOFs at its conferences , discuss standards issues with USENIX membership, recruit and instruct snitches, and work with the snitch editor in publishing the reports. Peter can be reached via email: pc@hillside.co.uk or Facsimile +44 227 762554.

USENIX Standards Watchdog Editor"

The end of an era in snitch reporting is at hand. Jeff Haemer is laying aside his quill as USENIX Standards Watchdog report editor and is taking on a more paternal role for a time. Jeff has edited the Standards Watchdog reports which are published in the USENIX Association's newsletter, and on comp.std.unix for the past two years. He was responsible for building the USENIX Standards Watchdog Committee into influential, bustling, nationally-recognized organisation.

Stephe Walli will take his place. Stephe has just attended his sixth IEEE POSIX meeting; his first as snitch editor.

Who are the USENIX Members?

The answer is that members are a diverse group, with no single identity. The USENIX membership survey shows a membership of individuals, corporations, and educational institutions, with wide-ranging interests and responsibilities.

Over 2,200 members replied to the survey, sent out in the Autumn of 1990. 64% of the members who replied were individual members, with 3% corporate members, 5% educational members, and 3% student members. 24% of the respondents did not state their affiliation. Most have been members of USENIX for 2 - 5 years, with a third being members for less than 2 years. However, 17% of the respondents have been with USENIX for at least 6 years.

The largest block of USENIX members are from the Pacific coast of the U.S., with the second largest block in the mid-Atlantic states. The top five states were California, Massachusetts, New York, Texas, and New Jersey. Countries with USENIX members include Australia, Brazil, Canada, Denmark, Finland, France, Hong Kong, Iceland, Italy, Japan, Korea, Malaysia, the Netherlands, New Zealand, Norway, Sweden, Switzerland, and the United Kingdom.

Approximately half of the members are programmers, with many other members being project leaders, senior managers, self-employed, researchers, and other positions. The members work for computer related manufacturing organisations, educational organisations, software companies, engineering/science/research and development groups, the government or military, system integrators, and more. Most of the members have a Masters degree or Bachelors degree from a college or university. Almost 10% have a Doctorate as well. Members span age groups, with 49% being 25 - 34 years of age, 35% being 35 - 44, 11% 45 or older, and 5% less than 25 years old. Experience in UNIX is varied, with half the members working with UNIX for 6 - 10 years. Other members have 2 - 5 years experience, and 18% having 11 - 15 years in the field. 3% of the members have 16 - 20 years experience.

The majority of the members attend one USENIX conference per year, with the largest group attending the summer conference. The attendees find the invited talks and panels add interest to the conferences and prefer having 5 or 6 invited talks per conference.

Members thought that the Association should continue its activity in POSIX balloting, POSIX institutional representation, reporting on International standards group WG15, provide standards snitch reports, provide the moderated newsgroup on standards, and the standards BoF at the conferences.

Software patent issues were important to many members, with 65% requesting the Association to help contest software patent applications and 53% asking for patent snitch reports.

40% of the members have access to the Internet and 40% have access to USENET. Other networks mentioned include BITNET, Applelink, ACSNET, CSNet, JUnet, NORDUNET, NSFNET, SURAnet, UUNET, VNET, and more. Members have the ability to both communicate and assist one another.

Overall, the membership survey shows a highly technical professional group with wide areas of interest and responsibility in their organisations. They find the USENIX Association helps keep them up to date in their field and provides a forum for discussion of advanced computing.

Call for Tutorial Proposals

In an effort to continue to provide the best possible tutorials to its conference attendees, the USENIX Association is soliciting proposals for future new tutorials. The proposals can cover any subject, ranging from introductory to advanced materials, although one should avoid overly introductory materials. Previous conferences have included tutorials on such diverse topics as UNIX Network Programming, X Toolkit Intrinsics, Topics in System Administration, Multimedia and Hypermedia, System V and Berkeley Kernel Internals, and Software Contracts and Intellectual Property, among many others.

All proposals will be considered for the Winter 1992 in San Francisco and all conferences after that. The tutorial schedule for San Francisco is almost finalised, so proposals should arrive as soon as possible. The deadline for proposals for the Summer 1992 conference is 17 December 1991. For a more detailed description of the USENIX tutorial program and guidelines in preparing a tutorial proposal, please contact:

Daniel V. Klein, USENIX Tutorial Coordinator

Email dvk@usenix.org, or by physical mail to:

5606 Northumberland
Pittsburgh
Pennsylvania 15217-1238
USA

Please include your physical and e-mail address, along with your telephone number.

USENIX Mach Symposium, Monterey, California

The Mach Symposium will be held in Mid-November 1991. Please see the preannouncement and call for papers in this issue.

1991 USENIX Symposia, Workshops, and Conferences

Upcoming conferences and symposia include:

- Large Installation System Administration V Conference, on 30 September - 3 October 1991 in San Diego, California.
- MACH Symposium, Mid-November, 1991 in Monterey, California. USENIX Winter Conference, 20 - 24 January 1992 in San Francisco, California.
- Symposium on Experience with Distributed and Multiprocessor Systems - III, 26-27 March 1992.

Further Information on Conferences and Workshops

If you need further information regarding USENIX conferences or workshops, contact the USENIX Conference Office at

22672 Lambert Street
Suite 613
El Toro
California 92630
USA

Email to judy@usenix.org or {uunet,ucbvax}!usenix!judy

Telephone +1 714 588 8649
Facsimile +1 714 588 9706

Further Information about the USENIX Association

USENIX, the UNIX and Advanced Computing Systems professional and technical organisation, is a not-for-profit association dedicated to

- fostering innovation and communicating research and technological developments,
- sharing ideas and experience, relevant to UNIX, UNIX-related and advanced computing systems,
- providing a forum for the exercise of critical thought and airing of technical issues.

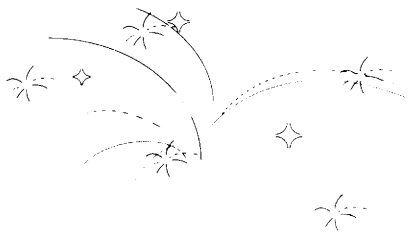
Founded in 1975, the Association sponsors two annual technical conferences, and frequent symposia and workshops addressing special interest topics, such as C++, Mach, systems administration, and distributed/multi- processor systems. USENIX publishes proceedings of its meetings, a bi-monthly newsletter ;login:, a refereed technical quarterly, Computing Systems, and is expanding its publishing role with a book series on advanced computing systems. The Association also actively participates in and reports on the activities of various ANSI, IEEE and ISO standards efforts.

If you would like information on membership, or would like information on ordering USENIX publications (proceedings, manuals, Computing Systems, or the Association's newsletter, ;login:), please contact the USENIX Association Executive Office at:

2560 Ninth Street
Suite 215
Berkeley
California 94710
USA

Email office@usenix.org

Telephone +1 415 528 8649
Facsimile +1 415 548 5738



Donnalyn Frey and Rick Adams

In case you haven't heard yet, Matthew Kevin Adams was born on March 20.

He is absolutely adorable and is doing fine!

Preliminary Announcement and Call for Papers

Usenix Mach Symposium

Monterey, California, USA

Mid-November, 1991

Background

Mach has become a dynamic addition to the operating systems marketplace. DARPA originally sponsored Mach development, and continues to emphasize the use and growth of Mach. In the larger research community, Mach is ever more widely used at many university sites and industrial research labs. Versions of Mach have been released commercially by Encore, NeXT, BBN and mt Xinu. The Open Software Foundation chose Mach as the basis for its operating system offering; now, OSF/1 is finding increasing acceptance as computer vendors ready products derived from it.

Recent developments have demonstrated the feasibility of Mach 3.0, the combination of a pure Mach kernel with single or multiple servers emulating the features of traditional operating systems. Performance of Mach 3.0 has begun to approach or exceed that of Mach 2.5. Workers outside of the CMU community have begun to use Mach 3.0 as the basis for their projects. In short, acceptance of Mach has come about in an astonishingly brief time.

Activity in this field has been sufficiently wide-spread that, little more than a year after the first Usenix Mach workshop, the Usenix Association is pleased to sponsor an expanded Mach symposium to bring together researchers, engineers, vendors and users of Mach systems. We will encourage discussion of all past and present Mach-related research, development, production and applications activities.

Symposium Overview

The symposium will be spread over three days. The first day will be devoted to two half-day tutorials on advanced programming for Mach 3.0. The following two days will concentrate on presentation of refereed papers on current and historical Mach-related work. Long breaks between presentations provide ample opportunity for informal discussion. Some time will be available for descriptions of work in progress.

Tutorials

Richard Draves	Writing a Multi-Threaded Mach 3.0 Server
David Black	Writing an External Memory Manager

Richard Draves will lead a tutorial analysing the process of writing a multi-threaded server, with particular attention paid to the complexities of using Mach IPC. During the course of his doctoral studies at Carnegie-Mellon University, Rich rewrote Mach 3.0 IPC to solve problems that became apparent with Mach 2.5 servers.

David Black will demonstrate how to create an external memory manager; discussion will centre on the intricacies of developing an efficient (and well-behaved!) external manager. David, currently of the Open Software Foundation, received his doctorate from Carnegie-Mellon for his contributions to Mach.

These tutorials are being developed precisely for this Usenix Mach symposium. They will explore concepts and rationale as well as real examples. They are oriented towards programmers who already have some familiarity with using Mach IPC and VM. Each tutorial is a half-day, so conference attendees may take part in both. The tutorials will be priced separately from the conference registration fee.

Submissions

Extended abstracts of 1500-2500 words (9000-15000 bytes or 3-5 pages) should be sent to Alan Langerman at the address below (those submitting hardcopy abstracts must send six copies). Shorter abstracts run a significant risk of rejection as there will be little on which we can base an opinion. Preference will be given to those submissions that include an outline of the entire paper in addition to the extended abstract. Authors must also supply an estimate of the length of the full paper.

A good extended abstract will contain the following information in one form or another:

Abstract	100-300 words (half a page) included verbatim in the final paper
Introduction	The problem; its importance; previous work
Solution	Issues, decisions, tradeoffs, rationale.

	Implementation details.
Evaluation	Performance results; effort required; lessons learned. Conclusion

The extended abstract will allow us to analyse the content of your proposed paper. This layout is not cast in concrete; just submit enough material to convince the committee that they want to accept the paper!

An outline lists the headings, major points and many minor points for each section of the actual paper. The outline gives us an idea of the form and style of your paper.

The submission package should include:

- Your extended abstract
- Outline of rest of paper, if at all possible
- Cover letter, detailing
  - Title of paper
  - Authors
  - Estimate of paper length
  - Contact author (liaison to program committee)
  - E-mail address and daytime phone number for contact author
  - Optional home phone number
  - Optional FAX number
  - Surface mail address (required)
- If you are submitting hardcopy, six copies are needed.

The submission should be sent electronically to me, alan@encore.com, or by surface mail to me at the address listed below. I will not copy and re-distribute FAXes, so don't bother to send them.

All submissions will be acknowledged. Authors of approved abstracts will be required to submit full-length papers (8-15 pages) approximately five weeks after notification of acceptance.

Areas of interest include, but certainly are not limited to:

- Applications
- Mach 2.5 and earlier development
- Mach 3.0 monolithic server
- Mach 3.0 multi-server
- Problems with Mach 2.5 / Mach 3.0 features
- Multiprocessor or parallelisation experiences
- Security
- Performance
- Productisation
- Experiences with OSF/1
- Use of Mach subsystems in other operating system kernels
- Comparisons of Mach with other operating systems; e.g., Chorus, Sprite, Amoeba, V, and of course Unix
- Porting Mach to off-beat architectures
- Future work

Important dates:

Extended abstracts:	19 July 1991
Notification:	23 August 1991
Camera-ready, full papers:	4 October 1991

For further information about the symposium, contact the program chair:

Alan Langerman
Encore Computer Corporation
257 Cedar Hill Street
Marlborough, Massachusetts 01752
USA

Email alan@encore.com

Voice +1 508 460 0500
Facsimile +1 508 485 0709

Program Committee

Larry Allen	Open Software Foundation
Nawaf Bitar	Hewlett-Packard Company
Susan LoVerso	Encore Computer Corporation
Melinda Shore	Cornell University
Michael Young, Ph.D.	Transarc

Call For Papers & Preannouncement

USENIX Winter 1992 Technical Conference

20-24 January 1992

San Francisco Hilton San Francisco, California, USA

Schedule Of Events

Tutorial Program

Monday and Tuesday, 20-21 January

Refereed Papers and Invited Talks

Wednesday through Friday, 22-24 January

USENIX Reception

Wednesday evening, 22 January

UniForum, scheduled concurrently with the USENIX Winter Conference, takes place 22-24 January at San Francisco's Moscone Convention Centre.

The USENIX Association's two annual technical conferences are well recognized as the leading forums for the communication of new research and investigation of important developments in UNIX, UNIX-related and advanced computing systems as well as for the tutorial program offerings.

At the USENIX Winter 1992 Technical Conference, software professionals and technical managers tackle questions of immediate importance to advance computing systems development and management.

Tutorial Program

Tutorial Coordinator: *Daniel V. Klein* Telephone +1 412 421-2332, Email dvk@usenix.org

Introductory as well as advanced, intensive yet practical, example-filled tutorials by leading experts will focus on topics essential to successful technical management of UNIX, UNIX-like advanced computing systems, X windows, Mach, the C and C++ programming languages and related areas of interest. The tutorial program at San Francisco will include topics such as:

- Programming in Perl Advanced System Administration
- Introduction to the TCP/IP Suite Network Programming
- Introduction to C++ Programming with OSF/Motif
- System VR4 Internals Programming X Windows
- Network and System Security OSF/1 Internals
- UNIX Programming Tools Introduction to X Toolkit Intrinsics
- Mach Overview New Kernel Facilities in 4.3BSD-Reno

Refereed Papers

Papers formally reviewed and accepted by the Winter 1992 Conference Program Committee are presented during the conference's three days of technical sessions and published in the conference proceedings. Presentations are supported with appropriate audio/visual and include a scheduled question-and-answer period.

Invited Talks

Invited Talks Coordinators:

Sharon Murrel, AT&T Bell Laboratories

Andrew Hume, AT&T Bell Laboratories

A full series of invited talks will teach the tricks of using standard UNIX tools and tackle the difficulties of system administration and integration. We welcome both suggestions for new and interesting topics as well as sub- missions proposing a particular session. Proposals should include a brief outline. Be sure to emphasize why your topic is of general interest to our community and what your main focus would be. Send these submissions to:

Email ITusenix@Usenix.org

or

Email uunet!usenix!ITusenix paper

or

Andrew Hume
Bell Labs 2C-515
Murray Hill
NJ 07974
USA

Facsimile +1 (908) 582-5857

Birds-of-a-Feather Sessions

Tuesday-Thursday evenings., 21-23 January

Schedule a BOF in advance by contacting Judy DesHarnais in the USENIX Conference Office at +1 714 588-8648 or judy@usenix.org.

Conference Headquarters: San Francisco Hilton

A luxury hotel in the heart of downtown San Francisco, two blocks south of Union Square. Discount air fares and special, affordable room rates at the San Francisco Hilton on Hilton Square and nearby hotels will be available to USENIX Conference attendees.

The SF Bay Area abounds with adventures for every taste. Information on many activities will be included in your registration packet, such as:

- "The City's" wonderful restaurants
- Golden Gate Park
- The Exploratorium
- The Garage (Technology Museum), San Jose
- Muir WoodsQA Natural "Redwood Cathedral" just across SF Bay
- Touring the Wine Country

and much more!

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline reservation information will be mailed in October 1991. If you did not receive a printed copy of this announcement directly and wish to receive the pre-registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
El Toro, California 92630
USA

Telephone +1 714 588-8649
Facsimile +1 714 588-9706

Call For Papers

Some believe that UNIX standardization efforts have killed innovation. And yet, we need innovation, and opportunity for it abounds.

Large write-once disks make the current filesystem untenable. Even the 2 gigabyte file limit built in all through the system breaks. Gigabit networking clogs an I/O model designed to push hundreds of kilobytes per second, not hundreds of megabytes. System administration for thousands of machines? Programming tools for distributed workgroups? Object-oriented and visual programming? Microkernels with client/server architectures? RAID disk arrays? Transcontinental file servers? What's a programmer to do?

The USENIX Winter 1992 Conference solicits new work on all topics related to UNIX or UNIX-inspired systems programming and technology. But as always, we care most about innovation and how it coexists with (and sometimes thrives on) stasis.

Please target a sophisticated technical audience, particularly knowledgeable of operating system issues and keenly interested in new and exciting projects in many areas.

Vendors are encouraged to submit technical presentations on products. However, we will reject obvious product announcements. Previously published papers will also be rejected, although "retrospective" papers may describe work done years ago.

Submissions must be in the form of extended abstracts, 1500-2500 words in length (9000-15000 bytes or 3-5 pages). Shorter abstracts will not give the program committee enough information to judge your work fairly and, in most cases, this means your paper will be rejected. Longer abstracts and full papers simply cannot be read by the committee in the time available. However, you may append a full paper to an extended abstract; this is sometimes useful during evaluation.

The extended abstract should represent your paper in "short form." The committee will want to see that you have a real project, that you are familiar with other work in your area (i.e., include references), and that you can clearly explain yourself. Please, this is not a mystery to be solved: you should have results and they should be summarized in your abstract.

A good submission will contain:

- Abstract
  - The abstract should be included verbatim in the final paper.
- Introduction
  - Introduce the problem: why is it important?
  - Reference previous work.
- How We Solved the Problem
  - More details on the problem and its issues.
  - Design decisions and tradeoffs, and why they were made.
  - Implementation details.
- Evaluation
  - Data on performance and effort required.
  - How well does it work?
  - What would you do differently?
  - If it failed, why?
  - What did you learn from it?
- Conclusion
  - Summarize the paper, emphasizing why it is important and what was learned.

In addition to the extended abstract, every submission should include:

- A clearly designated contact author who will be your link to the program committee.
- A daytime phone number (essential!).
- A surface mail address (required).
- An email address, if available; email is by far our best path of communication.
- A home phone number (optional, although questions often arise on evenings and weekends and it will avoid delays).
- A FAX number (optional).
- Any special audio/visual equipment you may require. A microphone, overhead projector, and 35mm projector will be provided as standard equipment. We are happy to provide additional assistance and equipment to make your presentation as audio and visually appealing as possible.
- Indication of student status.

Presentations are usually scheduled for 25 minutes.

The final date for submissions is August 19. Authors of accepted submissions will be notified by October 1. They will immediately receive instructions for the preparation of camera ready final papers to be published in the conference proceedings. Camera-ready papers of 8-12 typeset pages will be due by 22 November.

Submissions can be sent (in order of committee preference):

via email to:

SFusenix@usenix.org or uunet!usenix!SFusenix

via paper to:

Eric Allman
Computer Science Division, EECS
University of California
Berkeley, California 94720, USA

Facsimile +1 415 843-9461

Award for Best Student Paper: A cash prize for the best paper by a full-time student will be awarded by the conference program committee. With your submission, please indicate if you are a full-time student.

Award for Best Paper at the conference is also made by the committee.

Technical Program Committee

Chair: Eric Allman	University of California, Berkeley
Rick Adams	UUNET Technologies, Inc.
Andrew Birrell	Digital Equipment Corporation, Systems Research Center
Tom Ferrin	University of California, San Francisco
Bob Gray	US West Advanced Technologies
Teus Hagen	OCE
Steve Johnson	Athenix
Pat Parseghian	AT&T Bell Laboratories
Dennis Ritchie	AT&T Bell Laboratories
Greg Rose	IBM Thomas J. Watson Research Center
David Rosenthal	Sun Microsystems
Brent Welch	Xerox PARC

Relevant Dates

- Abstracts Due Monday, 19 August
- Notification to Authors Tuesday, 1 October
- Camera-ready Papers Due Friday, 22 November

Mick Farmer
Department of Computer Science
Birkbeck College
London, UK

mick@cs.bbk.ac.uk

Mick is the Secretary of the UKUUG. His primary interests are Ornithology (restricted to the Western Paleartic at the moment because of cost) and Oenophilism (especially pre-1962 Bordeaux, pre-1980 Burgundy, and 1945 Port). His secondary interests include Software Consultancy (to pay for the above primary interests) and Distance Learning Methods (especially interactive video and hypertext). When not pursuing these and other interests he can be found at Birkbeck College (London) where he teaches in the Department of Computer Science.

He lives in Lewisham (South East London) with his wife Sue and a TV called Sonya. His neighbours have two children and a dog.

Hello peeps,

Solution to Puzzle Number 17

Male gardener and female cook — As 100 is not an exact multiple of three, the gardener at his thirty-fourth bound will go two feet beyond the mark. The gardener will, therefore, run to a point 102 feet straight away and return (204 feet in all), and so lose by four feet.

Female gardener and male cook — Now "she (the gardener) made three bounds to his (the cook's) two." While the gardener is running her 204 feet in 68 bounds of three feet, the cook can only make 45.333... of his two foot bounds, which equals 90 feet 8 inches. The result is that the lady gardener wins the race by 109 feet 4 inches at a moment when the cook is in the air, one-third through his 46th bound.

The moral of this puzzle should be obvious to everyone. It reminds me of a riddle sent to me by Bill Barrett:

A young man was injured in an accident and required surgery. The surgeon, on seeing him on the operating table, exclaimed "My son!" Yet the surgeon was not that young man's father. Can you explain that?

Solution to Puzzle Number 18

46	55	44	19	58	9	22	7
43	18	47	56	21	6	59	10
54	45	20	41	12	57	8	23
17	42	53	48	5	24	11	60
52	3	32	13	40	61	34	25
31	16	49	4	33	28	37	62
2	51	14	29	64	39	26	35
15	30	1	50	27	36	63	38

All the columns and rows add up to 260. Unfortunately, it is not a perfect magic square because the diagonals are incorrect, one being 264, the other 256. So near, yet so far!

Puzzle Number 19

Write down an expression using only the figure 7 four times, which will equal 100. Any mathematical symbols whatever can be used, but no other figure may be used except 7, and this is used exactly four times.

Puzzle Number 20

Write down an expression using only the figure 4 four times, which will equal 71. The rules are the same as for the previous puzzle.

If you have any favourite puzzles that deserve a wider audience then please send them (together with solution :-) to me. They might appear in this column!

Loads-a-puzzles,

Mick

Book Review

UNIX Survival Guide
Tim Parker, Addison-Wesley, 1990
ISBN 0-201-57078-5
(US) Price $22.95, Soft Back, 259 pp
Size 23cm x 19cm

Reviewed by Brian O'Donovan of Digital, Galway, Rep. of Ireland

This book is aimed at UNIX beginners. It makes no assumptions about prior knowledge of UNIX or related systems. It introduces the main UNIX commands in a simple tutorial style.

The title "UNIX Survival Guide", is presumably intended to be reminiscent of the well known "SAS Survival Guide". The author's idea is that you should use this book to help you survive in a UNIX jungle in much the same way that you would use the "SAS Survival Guide" to help you survive in a South American jungle.

The book starts off by describing some very basic operations such as logging on to the system. As you get deeper into the book the material is a little more advanced. Topics covered include:

- Basic Commands.
- Sending and Reading Mail.
- Editing Files.
- Use of the Shell.
- System Administration.

At all stages of the book, the material is well presented and written in clear english. If you have never used a UNIX system before, this book will get you started. Because of its clear style, you should be able to easily follow all of the discussion. Hence, it could be said that the book meets its main goal. However, I have two major complaints about this book.

My first complaint relates to the section about system administration. Since the author does not specify which UNIX system this material relates to, a novice user could assume that this material is valid for all UNIX systems. As any UNIX system administrator knows, UNIX systems can be very different from each other with regard to how they are administered. As a result of this the information provided is incorrect for many systems. On the basis that "a little information is a dangerous thing", I would not recommend that anyone uses this book as a guide to UNIX system administration.

My second complaint relates to the exaggerated claims made on the front cover of the book. The front cover of the book claims that the book "includes coverage of XENIX, OPEN LOOK, and ULTRIX." However, ULTRIX and XENIX are only mentioned in one sentence on page 215, "Digital Equipment sells ULTRIX, IBM has AIX, Santa Cruz Operation has XENIX." Open LOOK is described, but this description is very brief, (only a page and a half of text). If I had bought this book to find out about XENIX, OPEN LOOK, or ULTRIX, I would be very disappointed.

In conclusion, this book will enable you to survive on a UNIX system. However, if you have hopes of achieving anything more than survival, you will need to look elsewhere.

Book Review

Programming Perl

Larry Wall & Randal L Schwartz,
O'Reilly & Associates, 1991,
ISBN 0-9377175-64-1.
(US) Price $24.95, SoftBack, 456 pp plus 16 pp facts card,

Reviewed by John Collins of Xi Software Ltd

If you haven't met Perl you should. It does everything that Awk and Sed do, most of what various shells do, and has goodies that you would normally have to write a C program for thrown in, plus some oddball extras like report formatting.

You can even poke around binary files if you want to (and I use it to produce a meta-RCS system which will identify which version of which program consisting of several source files and directories a binary file is). It's got a built-in debugger too.

The inventor of Perl, Larry Wall, is one of the authors, and he has covered it very comprehensively in this book. It is certainly much better than the man page that comes with Perl, which I found hard to use when I was learning the language as constructs are there mentioned in odd places.

The book is written in a fairly humorous style, and is liberally sprinkled with useful example programs. The facts card is very handy as there are funny constructs like '$_' and '$|' you have to remember the meaning of when you are using Perl.

I think that the book explains Perl very well, but I suspect that some ill-humoured people will object to the style. The layout of the book is good, and things are led up to well. I don't think that anything is left out, even a discussion of the natural history of the camel on the cover, not to mention Larry Wall himself.

My only grumble is that it is describes a more recent version than generally around in the UK. The book gives phone numbers in Ohio to uucp from to get the current version, and flushed with my recent purchase of a Trailblazer clone I tried them but there was no answer or garbage on the numbers I tried. Fortunately I talk to people in Ohio who purloined a copy.

If you do perform text mangling and write monster shell and awk scripts for juggling around and editing files and find them rather tedious, this book, and Perl itself, are well worth getting.

Book Review

Crafting a Compiler with C

Charles N.Fischer & Richard J.LeBlanc,
The Benjamin/Cummings Publishing Company Inc, 1991, Distributed by Addison Wesley.
ISBN 0-8053-2166-7.
(UK) Price £19.95, Hard Back, 812 pp

Reviewed by Rob Henley of Siemens-Nixdorf, Bracknell, UK.
(rob@siesoft.co.uk)

First, let me add my apologies to those of the editor for incorrectly attributing "The UNIX Programmming Environment" to Kernighan and Plauger! This well known book is of course by Kernighan and Pike. At least it shows that someone reads these reviews!

Crafting a Compiler was first published about 4 years ago - the current book is an update of this publication, with algorithms and pseudo-code examples in C syntax. I think many people will immediately compare this book with the famous "dragon book" by Aho and Ullman (Principles of Compiler Design, 1977), and indeed the two are similar in scope and completeness.

Crafting a Compiler with C provides a thorough, but readable introduction to the subject of compiler design. Each subject is dealt with in some detail, and anyone wishing to implement a compiler has a good selection of references to supplement the information given in the book (even 800 pages can't cover every detail of such a large field!). The text is succint but not obtuse. Examples are based on ADA language features (being one of the "biggest" languages), but special attention is also given to well known features of other languages (e.g. Pascal, Modula-2 and C itself).

Chapter 1 gives a brief but clear introduction to the subject and is followed in Chapter 2 by the definition of a small language called *micro* and an introduction to recursive descent compilers. Practical "hands-on" examples like this are a great strength of this book.

Chapters 3 and 4 go on to discuss scanning and parsing respectively: the theory is described clearly and is tempered with real-world examples. Compiler tools such as scanner generators (Lex and ScanGen) and parser generators (Yacc and LLGen) are introduced at an early stage. Every chapter has a comprehensive set of exercises covering both the theory and practical examples.

As you get further into the book, the subject matter becomes more specific. First LL(1) and then LR parsing is introduced, followed by chapters on semantic processing, symbol table handling and run-time management. Particular semantic issues are then covered in detail - translation of declarations, expressions, control structures and functions.

Peephole and global optimisation techniques are covered in separate chapters, although appropriate optimisations are also indicated "in-line" where individual constructs are discussed. Finally, a chapter on "Parsing in the Real World" ties together a lot of the ideas to give a practical strategy for compiler writing, making full use of available tools.

The appendices give details of some of the tools described in the text (ScanGen, LLGen and LALRGen), but the reader is referred elsewhere for details of Lex and Yacc (although the introductions to these tools in the early chapters is very good). A subset of ADA is also defined (ADA/CS), both as the basis of language features discussed in the text and as an example suitable for implementation.

The text contains extensive C-style code which can be used as the basis of the reader's own implementations. The flavour of the book is very much "hands-on" and there are plenty of pointers for interesting projects to try. I felt that the subject of error handling was glossed over a little, although this may have been necessary to keep the book to a manageable size. One or two topics - for instance "threaded code" - I thought deserved a more thorough treatment too, but again this is only a personal preference and the book is already 800 pages long!

In conclusion I think this book is a very good buy for anyone interested in compiling techniques. The treatment is very complete and bang up to date, drawing on a lot of developments over the last 5 years. The book also draws on the authors' experience in education and would be a good basis for an undergraduate course. By using a C based syntax I think the book should be particularly accessible to the UNIX community - unfamiliar pseudo-code can make it harder to concentrate on algorithms! (If anyone has a real aversion to C, they could check out the first version of the book which uses a Pascal- like pseudo code). It is well indexed, and useful as a reference.

Abstracts

Here are the abstracts of the papers that were delivered at the EurOpen conference held in Tromsø in May. Copies of the proceedings may be obtained from EurOpen at Owles Hall.

Thanks are due to Stuart McRobert <sm@ic.doc.ac.uk> who typeset the proceedings and provided the abstracts.

Experiences with Amoeba

Sape J. Mullender
University of Twente, Netherlands

Email mullender@cs.utwente.nl

The Amoeba distributed operating system has been in use now for a few years. It has been used in experiments with parallel algorithms, as a distributed UNIX-like system, in real-time applications, and in event-processing for proton-scattering high-energy physics experiments.

We have discovered many of the strong and the weak points of Amoeba. On the positive side, we are very pleased with our RPC-based communication, with capability-based protection, with a free-standing naming service, the bootstrap service and with the process-management facilities.

On the negative side, we are not happy with our incomplete, non-binary-compatible UNIX functionality, with the flat port name space which limits the ability to scale, with an immutable-file service, and with a kernel implementation of user threads.

In this paper, I will present a brief overview of Amoeba, discuss our experiences with Amoeba, and present some of my current work on the design of a new distributed system.

A New Look at Microkernel-Based UNIX Operating Systems: Lessons in Performance and Compatibility

Allan Bricker Michel Gien Marc Guillemont
Jim Lipkis Douglas Orr Marc Rozier
Chorus systèmes, France

Email {allan | mgu | mg | lipkis | doug | mr}@chorus.fr

An important trend in operating system development is the restructuring of the traditional monolithic operating system kernel into independent servers running on top of a minimal nucleus or 'microkernel'. This approach arises out of the need for modularity and flexibility in managing the ever-growing complexity caused by the introduction of new functions and new architectures. In particular, it provides a solid architectural basis for distribution, fault tolerance, and security. Microkernel-based operating systems have been a focus of research for a number of years, and are now beginning to play a role in commercial UNIX systems.

The ultimate feasibility of this attractive approach is not yet widely recognised, however. A primary concern is efficiency: can a microkernel-based modular operating system provide performance comparable to that of a monolithic kernel when running on comparable architectures? The elegance and flexibility of the client-server model may exact a cost in message-handling and context-switching overhead. If this penalty is too great, commercial acceptance will be limited. Another pragmatic concern is compatibility: in an industry relying increasingly on portability and standardisation, compatible interfaces are needed not only at the level of application programs, but also for device drivers, streams modules, and other components. In many cases, binary as well as source compatibility is required. These concerns affect the structure and organisation of the operating system.

The Chorus team has spent the past six years studying and experimenting with UNIX 'kernelisation' as an aspect of its work in modular distributed and real-time systems. In this paper we examine aspects of the current CHORUS system in terms of its evolution from the previous version. Our focus is on pragmatic issues such as performance and compatibility, as well as considerations of modularity and software engineering.

The OSF/1 Operating System

Open Software Foundation

The OSF/1 operating system fulfills the requirements of open systems for both compatibility and innovation. It is compatible with the most widely used systems and supports all relevant industry standards. A wide array of innovative features, such as fully kernel supported symmetric multiprocessing, enhanced security, and dynamic system configuration, complete the picture.

The design of OSF/1 represents a return to the roots of the UNIX operating system. This system is based on Mach technology and in order to increase maintainability, extensibility and flexibility, is evolving into a micro-kernel architecture, where services are moved into user space. This modular approach provides a open

strategy for future developments in operating system technology, particularly well adapted to distributed systems.

Plan 9, A Distributed System

Dave Presotto Rob Pike
Ken Thompson Howard Trickey
Bell Labs, New Jersey, USA

Email {presotto | rob | ken}@research.att.com

Plan 9 is a computing environment physically distributed across many machines. The distribution itself is transparent to most programs giving both users and administrators wide latitude in configuring the topology of the environment. Two properties make this possible: a per process group name space and uniform access to all resources by representing them as files.

UNIX Internationals' Enterprise Computing Architecture for UNIX System V

Andrew Schuelke
UNIX International - Europe

Email andy@uieu.ui.org

With UNIX System V Release 4 now firmly established in the marketplace, UNIX International recently announced its new extended charter which expands UIs focus to encompass' functionality beyond that of the base operating system. With this expanded focus UNIX International will address the requirements of the industry in three environments critical to the success of open systems in the coming decades: Distributed Computing, Corporate Hub Computing, and Desktop Computing. These areas are tied together in the Enterprise Computing Architecture (ECA). In this talk I will present the architectural framework of the distributing computing components of the ECA. These components are the Basic Services (the core operating system) at the lowest level, moving up through the Network Communication Services (e.g. STREAMS), the System Services (e.g. object management services), the Application Services (e.g. transaction services), and at the highest level the Application Tools. Spanning these hierarchical services are the Security and Interoperability Services. In discussing Interoperability Services I will cover interoperability with IBM SAA, PCs, and existing distributed computing tools.

Evaluation of Distributed Operating Systems in Open Networks

Holger Herzog
Markus Kolland Juergen Schmitz
Siemens AG, Germany

Email makol%venedig%ztivax@unido

This report presents an evaluation method for distributed operating systems, which is based on necessary operating system functionality for specific application areas. Application areas in this context are distributed, real time, multiprocessor, security and UNIX applications. For this purpose it is necessary to investigate the functionality offered by a given system to support features like Object Orientation, Concurrency, Multiprocessor or Real Time capabilities. The evaluation method is applied to Mach and Chorus and gives hints to estimate the suitability of these systems for different application areas.

A Distributed Computing Environment Framework: An OSF Perspective

Brad Curtis Johnson
Open Software Foundation

Email bradcj@osf.org

This paper articulates an architectural framework, and the fundamental mechanisms that are required to support that framework, for a distributed computing environment. The emphasis of this framework is on open distributed systems. That is, it serves as a model that supports many of the requirements of a distributed system: such as the need for interoperability, the need to support the client/server distributed application model, and the need to account for the characteristics and challenges that are unique to a distributed environment.

This paper will describe a number of commonalities and differences between the stand-alone environment and the distributed environment. In doing so, it will raise a number of distributed system issues that must be resolved in order to satisfy the needs of an open distributed system.

Finally, this paper will also articulate the need and process for building an open distributed system so that it behaves like a system rather than a set of disparate components.

Integration Mechanisms and Communication Architecture in AxIS

Dario Avallone
Roberto Dottarelli Felice Napolitano
Ingegneria Informatica, Rome, Italy

Email dario@engrom.uucp

This work presents the communication architecture of the configurable Software Engineering Environment (SEE) AxIS. The architecture goal is to allow the integration and the support of user-defined production tools in a distributed environment.

Incorporating Multimedia into Distributed Open Systems

Gordon S. Blair Geoff Coulson
Nigel Davies Neil Williams
Lancaster University, UK

Email mpg@comp.lancs.ac.uk

In the near future, system software must adapt to the needs of open systems and also to the desire to incorporate multimedia capabilities. This paper describes an approach to reconciling the apparently conflicting demands of heterogeneity and multimedia. Our solution is based on a configurable add-in module with multimedia peripherals together with the necessary processing power and network interfacing. Application and host workstation access to this module is via an ODP compatible base services interface. Higher level software is also described which will complement the base services to provide a comprehensive platform for open multimedia applications.

Scalable Mainframe Power at Workstation Cost

J-P. Baud J. Bunn F. Cane F. Hemmer
E. Jagel G. Lee L. Robertson
B. Segal A. Trannoy I. Zacharov
CERN, Geneva, Switzerland.

Email ben@cernvax.cern.ch

Until very recently, big mainframes were considered essential for the provision of large-scale scientific batch computing services, requiring intensive tape and file space management, high throughput and maximum reliability.

However, RISC-based workstations have outstripped mainframes in CPU price/performance by an order of magnitude for some years, and workstation-class disk systems have finally become cost-competitive with mainframe disk storage. The last missing elements preventing mainframe replacement by cheaper distributed systems have been the software to enhance their (UNIX-based) operating systems, and a powerful enough networking technology.

Over the last year at CERN, two services have been established to prove the feasibility of physics batch processing using off-the-shelf high-end workstations. The initial service, called 'HOPE', uses a single 4-CPU HP-Apollo DN10000 workstation, and achieves well over 80% utilization for periods of many months. The second service addresses the scalability issue, using a modest number of heterogeneous workstations on a high-speed network, connected and managed by a portable distributed set of user-level software. The CPU, disk, and tape services are functionally separated, and individually subject to price-performance optimization. This system is already comparable in CPU capacity to the full CERN computer center (including a Cray X-MP/48 and an IBM|3090-600E), and is scalable to several times that size.

The system is called the 'Scalable Heterogeneous Integrated FaciliTy' or 'SHIFT'.

Design and Implementation of an Experimenta Load Balancing Environment

Wouter Joosen Bruno Vandenborre
Pierre Verbaeten
K. U. Leuven, Belgium

Email wouter@cs.kuleuven.ac.be

Many load balancing strategies have been proposed and simulated, but a significant amount of design issues have only been dealt with in an intuitive way. We have created a distributed load balancing system on top of UNIX. Our system is currently used as a base for evaluating design issues in load balancing strategies. .P This paper describes the design and implementation of the load balancing environment. It is a prototype to experiment with, in order to produce a flexible and adaptive system, with user transparency.

A Decentralized and Efficient Algorithm for Load Sharing in Networks of Workstations

Guy Bernard
Institut National des To"(aae"lo"(aae"communications
Evry, France

Email guy@bdblues.altair.fr

Michel Simatic
Alcatel TITN Answare
Massy, France

This paper presents the design and evaluation of a decentralized load sharing algorithm for networks of workstations, RADIO. With respect to general distributed computing environments, networks of workstations have some peculiarities. First, the global computing power is most of the time much underutilized. Second, users of workstations occasionally need a peak of computing power. Third, workstations are often diskless, so that running a process on one workstation or another does not add file migration overhead. Fourth, network interfaces often provide a broadcast capability, which may be used to reach several destinations in a single message. Last, workstations are often dedicated to an 'owner', so that a workstation may only be used for running foreign processes only when the workstationsownerdoesnot'use it (or at least when running foreign processes would not increase the ownersprogramsresponsetimebyasignificantamount).'The first three points make load sharing very attractive for a network of workstations. The fourth point may be used for simplifying the design of load sharing algorithms, but broadcasting is expensive. The goal of RADIO is to provide the benefits of a decentralized load sharing algorithm while preserving the personal character of workstations and providing good performance results, in particular with respect to extensibility.

The key feature of the RADIO load sharing algorithm is that it is decentralized but involves expensive broadcast messages only occasionally. The design choices for information policy, location policy and transfer policy are described and motivated. RADIO has been implemented on a network of Sun workstations, and runs entirely outside of the kernel. Experimental results show that the extensibility of RADIO is better than that of previous decentralized algorithms, based on broadcast messages.

Distributed Applications in Heterogeneous Environments

Bertil Folliot
Laboratoire MASI, Paris, France

Email folliot@masi.ibp.fr

In a system where many hosts are connected by communication networks, the choice of programs placement allows to benefit from the processing power of all hosts (idle or unused) and thus to reduce the program response time. In this paper we present a solution to execute and to control distributed applications in heterogeneous environment. We consider an application as a set of programs linked by a precedence graph. Each program may be allocated on different heterogeneous hosts and may specify different allocation criteria.

We have implemented the GATOS system to automatically distribute parallel applications among heterogeneous hosts and to provide a software layer in order to easily write new distributed applications in a heterogeneous environment. GATOS has been developed to be portable and to work in a network containing a large number of hosts.

Process Sleep and Wakeup on a Shared-memory Multiprocessor

Rob Pike Dave Presotto
Ken Thompson Gerard Holzmann
Bell Labs, New Jersey, USA

Email {rob|presotto|ken|gerard}@research.att.com

The problem of enabling a 'sleeping' process on a shared-memory multiprocessor is a difficult one, especially if the process is to be awakened by an interrupt-time event. We present here the code for sleep and wakeup primitives that we use in our multiprocessor system. The code has been exercised by months of active use and by a verification system.

Capturing the Behaviour of Distributed Systems

Terje Fallmyr
Nordland College, Norway

Email terje@ioa.hsn.no

David Holden
Harwell Laboratory, England

Email holden@harwell.uucp

Otto J. Anshus
University of Troms(o/, Norway

Email otto@cs.uit.no

The ability to manage distributed computing systems depend on knowledge of its behaviour. Monitoring can be used to obtain information about whole systems or parts of systems. Such pieces of information relate to different *abstraction levels* of the system. A piece of information about a part of a system we term a behaviour indicatorfP. This report describes behaviour indicators in relation to the monitoring model developed in the MANDIS project. In a worked example the mapping from high level behaviour indicators through *object set states* to *basic states and events* is shown. The mapping is illustrated by examples of a language, **SESL**, for specification of events and state changes.

Tools for Monitoring and Controlling Distributed Applications

Keith Marzullo Mark D. Wood
Cornell University, Ithaca, New York, USA

Email meta@cs.cornell.edu

The Meta system is a UNIX-based toolkit that assists in the construction of *reliable reactive systems*, such as distributed monitoring and debugging systems, tool integration systems and reliable distributed applications. Meta provides mechanisms for instrumenting a distributed application and the environment in which it executes, and Meta supplies a service that can be used to monitor and control such an instrumented application. The Meta toolkit is built on top of the ISIS toolkit; they can be used together in order to build fault-tolerant and adaptive distributed applications.

Distributing Objects

Andrew Herbert
Architecture Projects Mangement Limited
Cambridge, England

Email ajh@ansa.co.uk

Similar concepts, called 'objects' have appeared in several areas of computing, from object-oriented databases, object-oriented programming languages, application environments and graphical user interfaces. These concepts have been reviewed by Alan Snyder of HP in a technical report called 'The Essence of Objects'. This paper builds upon Snydersanalysisandpresentsthe'requirements for adding distribution to the object concept. It is written for an audience who understand object orientation, accept Snydersprinciples,'and want to know how distribution might modify them.

A Comparative Study of Five Parallel Programming Languages

Henri E. Bal
Vrije Universiteit, Amsterdam

Email bal@cs.vu.nl

Many different paradigms for parallel programming exist, nearly each of which is employed in dozens of languages. Several researchers have tried to compare these languages and paradigms by examining the expressivity and flexibility of their constructs. Few attempts have been made, however, at .l practical studies based on actual programming experience with multiple languages. Such a study is the topic of this paper.

We will look at five parallel languages, all based on different paradigms. The languages are: SR (based on message passing), Emerald (concurrent objects), Parlog (parallel Horn clause logic), Linda (Tuple Space), and Orca (logically shared data). We have implemented the same parallel programs in each language, using real parallel machines. The paper reports on our experiences in implementing three frequently occurring communication patterns: message passing through a mailbox, one-to-many communication, and access to replicated shared data.

Linking a Stub Generator (AIL) to a Prototyping Language (Python)

Guido van Rossum
CWI, The Netherlands

Email guido@cwi.nl

Jelke de Boer
HIO Enschede, The Netherlands

This paper describes how two tools that were developed quite independently gained in power by a well-designed connection between them. The tools are Python, an interpreted prototyping language, and AIL, a Remote Procedure Call stub generator. The context is Amoeba, a well-known distributed operating system developed jointly by the Free University and CWI in Amsterdam.

As a consequence of their integration, both tools have profited: Python gained usability when used with Amoeba (en for which it was not specifically developed (en and AIL users now have a powerful interactive tool to test and experiment with new client/server interfaces.

Providing Application Interoperability using Functional Programming Concepts

Frank Eliassen Randi Karlsen
University of Tromsø, Norway

Email {frank | randi}@cs.uit.no

Applications that are interoperable (like federated databases) may be manipulated together by the user without global integration. This can be achieved by providing a uniform language format for the definition and manipulation of multiple autonomous applications.

In this paper is given an overview of a proposed uniform language format FRIL based on a computing model and interface paradigm combining functional and object-oriented approaches. The framework for this approach is given as a distributed generic infrastructure (or platform) supporting interoperability of separate and autonomous applications.

FRIL provides remote programmable services interfaces to heterogeneous autonomous applications and includes constructs for formulating global actions combining operations from these interfaces. Global actions are supported by a distributed execution system executing FRIL programs as distributed transactions. Using a functional computing model for the area of distributed transaction processing is an original approach which is both interesting and challenging.

We also briefly present the transaction model for the generic infrastructure and the exception handling facilities of FRIL. The transaction model allows for applications with highly different requirements to local autonomy and local transaction management to participate in the same distributed transaction, e.g. they may exhibit different degrees of control over subtransactions they execute. Fault tolerance is supported by allowing for alternative transactions that are executed instead of transactions that fail.

Merits of Language-Oriented Approaches for Constructing Distributed Systems

Uwe Baumgarten
University of Oldenburg, Germany

Email Uwe.Baumgarten@arbi.informatik.uni-oldenburg.de

The need for high quality distributed systems is a great challenge the computer community is faced with. The quality attributes are, for instance security, reliability, and maintainability. In this paper a few questions will be answered in order to .I (i) gain a deeper understanding of the problems of distributed systems and to .I (ii) propose a methodology to overcome some of these problems. The questions which we are concerned with are: .QP .I What can we learn from current and recent approaches in .B "Distributed Ada" for the specification, design, construction, and implementation of .B "distributed systems" ? .QP How can we implement distributed systems in an elegant and impressive way using a methodology called .B "conceptsstepwiserefinement"?'

Some answers to these two questions will be given in this paper. Two ideas predominate in our approach for constructing distributed systems. The former idea states, that abstractions should be defined by means of languages. The latter idea says, that the implementation strategy should be guided by conceptsstepwise'refinement. .ig

.B Key Words: Distributed systems, levels of abstraction, language-orientation, conceptsstepwiserefinement,'distributed Ada. ..

Domain-based Support for Service Administration and Server Selection

V. Tschammer
GMD FOKUSs0, Berlin, Germany

Email Tschammer@fokus.berlin.gmd.dbp.de

Concepts and related support services for co-operation in an open services environment are described. The environment will develop as a consequence of the installation of wide-spread fiber optical networks and future communication systems and the development of open distributed processing standards. A domain concept is introduced for structuring the environment and applications according to organisational and operational aspects. A service concept provides means for specification and administration of service types, services, and servers. Server identification and selection is a basic functionality considered for delegation to a supporting environment of co-operating open systems.

We describe a domain-oriented concept and an information system for the support of service administration and server selection and its implementation on the basis of standard directory services.

The Evolution of the Kerberos Authentication Service

John T. Kohl
Project Athena, MIT, USA

Email jtkohl@mit.edu

The *(kR Authentication Service, developed at MIT, has been widely adopted by other organizations to eliminate the trusted-host problem in open networks. While a step up from traditional security in networked systems, *(kR version 4 is not sufficiently flexible for some environments. These inflexibilities and the remedies introduced with the *(kR version 5 are described.

Architechture and Implementation of a User-Space NFS

Benoy DeSouza
Nawaf Bitar
Apollo Systems Division, Hewlett-Packard Company

Traditionally, filesystems under UNIX have been built into the operating system kernel. The Sun .I Network File System (NFS) has not been an exception. The NFS client is implemented under the .I vnode layer, a filesystem implementation-independent layer that allows access to multiple filesystems through a common interface. The NFS server is merely a kernel daemon that services request from NFS clients.

In addition to consuming physical memory resources, a kernel implementation has the disadvantage of reducing extensibility and portability. These problems can be overcome by implementing NFS in user-space. This, however, may result in a severe performance degradation unless the underlying operating system provides specialized support for an extensible I/O mechanism.

This paper explores the architechtural and implementation issues involved in constructing a user-space NFS over such a mechanism, following which current performance data and possible enhancements for further performance improvements are presented.

XEUS Director: A Distributed Shell for an Intelligent Terminal System

Laszlo Biczok Kalman Szeker
Central Research Institute for Physics
Budapest, Hungary

Email h1096bic@ella.UUCP h1097sze@ella.UUCP

XEUS is a UNIX based intelligent terminal system that makes it possible for PCs0 (DOS) users to work under a more powerful operating system without having to change from their familiar user interface and hardware environment. Additionally, the system provides methods for a UNIX program running on the host machine to download routines to a terminal. By offering synchronization and parameter passing facilities, XEUS can support distributed processing.

XEUS Director is a distributed shell that works on the XEUS intelligent terminal system. It is designed to provide a common user interface for both the UNIX and the DOS mode of the terminal while fully utilizing the features of the XEUS system.

Austria - UUGA
Friedrich Kofler
Schottenring 33/Hof
A-1010 Wien
AUSTRIA
Telephone +43 222 34 61 84
Facsimile +43 222 310 44 62
Email uuga@tuvie.at

France - AFUU
Miss Ann Garnery
AFUU
11 rue Carnot
94270 Le Kremlin Bicetre
Paris
FRANCE
Telephone +33 1 46 70 95 90
Facsimile +33 1 46 58 94 20
Email anne@afuu.fr

Belgium - BUUG
Edgard Nyssen
BUUG Secretariat
VUB-FAC. T.W. ETRO/IRIS
Pleinlann, 2
B-1050 Brussels
BELGIUM
Telephone +32 2 641 2930
Facsimile +32 2 641 2883
Email buug@etro.vub.ac.be

Germany - GUUG
Dr Anton Gerold
GUUG-Vorstand
Elsenheimerstr 43
D-8000 MÜNCHEN 21
WEST GERMANY
Telephone +49 89 570 7697
Facsimile +49 89 570 7607
Email gerold@lan.informatik.tu-muenchem.dbp.de

Czechoslovakia - CSUUG
Sekretariat CSUUG
Výpočetni Centrum VSE
Nám. A. Zápotockého 4
130 67 PRAHA 3
CZECHOSLOVAKIA
Email csuug@Czechoslovakia.EU.net

Hungary - HUUG
Dr Elôd Knuth
Computer and Automation Institute
Hungarian Academy of Sciences
H-1502 Budapest 112, P.O. Box 63
HUNGARY
Telephone +361 1 665 435
Facsimile +361 1 354 317

Denmark - DKUUG
Inge & Mogens Buhelt
Kabbelejevej 27 B
DK-2700 Brønshøj
DENMARK
Telephone +45 31 60 66 80
Facsimile +45 31 60 66 80
Email sek@dkuug.dk

Iceland - ICEUUG
Magnús Gíslason
University Computer Center
Dunhaga 5
IS-107 Reykjavik
ICELAND
Telephone +354 1 694750
Facsimile +354 1 28801
Email iceuug-s@rhi.hi.is

Finland - FUUG
Outi Nyman
Finnish UNIX Users' Group
Perustie 23 A 11
00330 Helsinki
FINLAND
Telephone +358 0 4002043

Ireland - IUUG
Dr T G Murphy
School of Mathematics
Trinity college
Dublin 2
IRELAND
Telephone +353 1 772941
Facsimile +353 1 772694

Italy - i2u
Ing Carlo Mortarino
i2u
Viale Monza 347
20126 Milano
ITALY
Telephone +39 2 2520 2530

Sweden - EUUG-S
Hans E. Johansson
NCR Svenska AB
Box 1206
S-164 28 KISTA
SWEDEN
Telephone +46 8 750 26 03
Email hans@ncr.se

Netherlands - NLUUG
Sandra Scharff
p/a Transmediair BV
PO Box 297
3720 AG Bilthoven
THE NETHERLANDS
Telephone +31 30 291718/281820
Facsimilie +31 30 292294
Email nluug-buro@nluug.nl

Switzerland - CHUUG
Patrik Eschle
Physik University Zürich
Winterthurer str. 190
CH 8051 ZÜRICH
SWITZERLAND
Telephone +41 1 257 45 88
Email eschle@forty2.uucp

Norway - NUUG
c/o Balder programvare AS
Sentrumsbygget
PO Box 1344
N-1401 SKI
NORWAY
Telephone +47 9 870550
Facsimile +47 9 876766

United Kingdom - UKUUG
Bill Barrett
Owles Hall
Buntingford
Hertfordshire SG9 9PL
UNITED KINGDOM
Telephone +44 763 73475
Facsimile +44 763 73255
Email ukuug@ukc.ac.uk

Portugal - PUUG
Legatheaux Martins
Avenue 24 de Julho, n° 134, 7°
Lisboa
PORTUGAL
Telephone +351 1 395 06 42
Facsimile +351 1 67 18 76
Email sec@puug.pt

USSR - SUUG
Dr. Kuznetsov
Kosinskaja str, 16-3-65
Moscow 111538
USSR
Telephone +7 95 374 7049

Spain - UUES
Jose A Manas
Usarios de UNIX de España (UUES)
Telefonica I&D
C/Emilio Vargas 6
28043 Madrid
SPAIN
Telephone +34 1 337 42 22
Email jmanas@dit.upm.es

Yugoslavia - YUUG
Milan Palian
Parex Institute
Kardeljeva No 8
YU-61000 Ljubljana
YUGOSLAVIA
Telephone +38 61 214223
Facsimile +38 61 214223
Email milan@parex.yu

EurOpen Secretariat

Owles Hall
Buntingford
Hertfordshire SG9 9PL
United Kingdom

Telephone +44 763 73039
Facsimile +44 763 73255
Email europen@EU.net