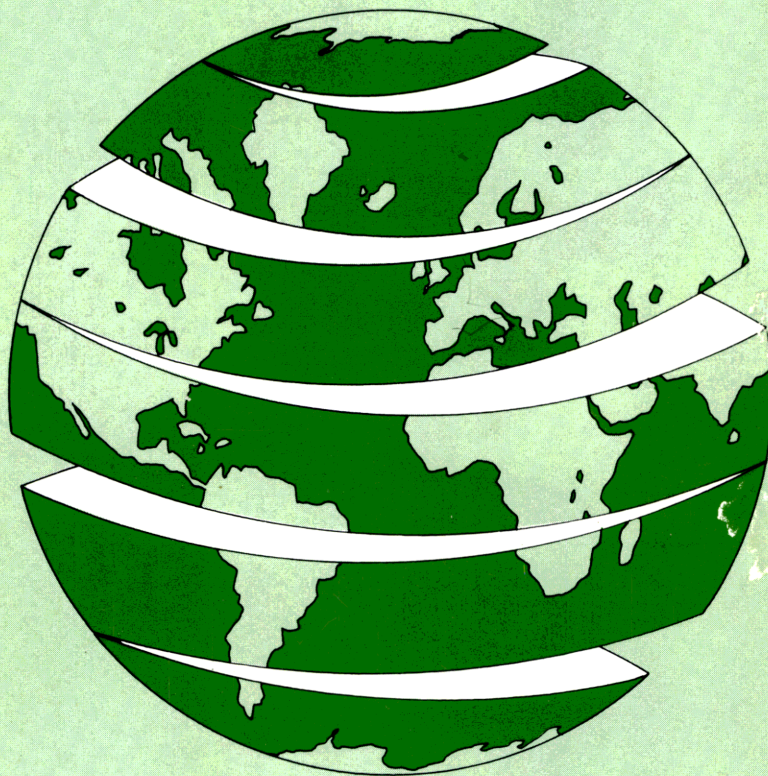


# EUUG

European UNIX<sup>®</sup> systems User Group



Volume 7, No. 1  
1987

## CONTENTS

- Birth of UNIX Cult
- CV Macros
- GKS in C++
- Conference Reports
- NRS Processor in C

# EUROPEAN UNIX SYSTEMS USER GROUP NEWSLETTER

*Volume 7  
Number 1*

The EUUG Newsletter: A Wind of Change .....	1
UNIX Conference Reports	
Atlanta Usenix, June 1986 .....	3
The Manchester Competition .....	19
Uniforum, January 1987.....	25
Notes on the Birth of the UNIX Cult.....	29
The X/OPEN Show in Luxembourg .....	37
The CV Macros.....	39
News from the National Groups .....	41
AFUU Diary.....	43
News from the Danish Group.....	45
News from the Netherlands.....	47
Letter from Germany .....	49
Forthcoming UKUUG Events.....	51
GKS in C++.....	53
An NRS Processor in C and the Future.....	65



## The EUUG Newsletter: A Wind of Change

*Alain Williams*  
*addw@phcomp.uucp*

*Parliament Hill Computers*

### *What ?*

There are changes afoot in the newsletter.

We are trying make it something that you will want to read, enjoy, and argue over. In a short while we hope that you will look forward to that unmarked plain brown envelope landing on your desk. Your company will have to order more than one copy because everybody will want to read it.

The newsletter is part of the glue that holds the EUUG together. For the many of you that do not go to EUUG conferences, it is probably your only view on what the EUUG is and does.

### *How ?*

To date the newsletter has largely contained reports and announcements for conferences, reports from individual national user groups, and a few weighty articles on topics that, though interesting, are often only of immediate interest to a minority of the membership.

We wish to expand on the above and include items that will interest everyone. Some of the articles will reflect the current trend for commercial applications of UNIX. Others will attempt to discuss what is happening to UNIX, and where it is (or should be) going. Others will try to educate people at all levels. Others will contain product announcements. Others will be book reviews. Others will simply amuse.

### *When ?*

Starting with the next newsletter which will appear on the 1st of June. Thereafter strict copy dates will be set with the newsletter appearing one month later. The dates are:

1 February	for	1 March
1 May	for	1 June
1 August	for	1 September
1 November	for	1 December

### *What is the Consequence?*

Much of this depends on you. The newsletter is for you, but can only provide what you want if you take an interest in it. This means that we want people who will contribute articles, as well as suggest ideas on what you want to see it contain.

For starters: Are there any crossword compilers out there ? Are you doing something interesting that others may be interested in ? Are you organising conferences/meetings ? Have you read any good (or bad) UNIX books recently ?

Would you like to write an article for the newsletter?

### *Submitting Articles*

Articles can be submitted by e-mail to:



*euug-n@mcvax*

The following information should be included at the top of the article:

- The title of the article
- The author's full name and title
- The author's e-mail address
- The author's affiliation — employer or user group, as appropriate.

*Advertisements*

Camera-ready advertising copy is accepted for inclusion in this newsletter: please contact the European UNIX systems User Group secretariat at

Owles Hall  
Buntingford  
Herts  
SG9 9PL  
U.K.

for details of availability and prices.

*Joining the EUUG*

If you are not already a member of the EUUG, either directly or through your national group, you can obtain a membership application form from the secretariat at the above address.

## Trip Report: Atlanta USENIX June 1986

*Nick Stoughton*  
*n.stoughton@inset.co.uk*

### *The Instruction Set*

*An amusing but content free diary of events during the 1986 Atlanta USENIX meeting. It is suggested that the reader refers to the conference proceedings for details of the papers actually given!*

#### *1. Friday 6th June, 12.00*

Summoned to Nigel Martin's office: "Boyd is ill, he will be off sick for at least two weeks. Therefore he will be unable to go to Atlanta on Monday. Would you like to go?"

What a stupid question! Enter panic mode to get things tidied up, flights changed, hotel reservations changed, etc etc. Ticket delivered.

#### *2. Monday 9th June, 11.00*

Notice that itinerary has the return flight taking -5 hours, after adjusting for local time differences! Not even Concorde goes that fast! They got the time of landing out by 12 hours. Oh well. Taxi arrived for the airport. Last minute rush to get tapes made for customers. Wave goodbye. Flight is from Gatwick at 13.40. Call in at home (Redhill, just 4 miles from Gatwick) to pick up suitcase. Leave a message for the wife so she doesn't sit at the airport for twelve hours on Saturday.

The plane is crowded. Sitting right at the front of the economy class area, with the bulkhead right in front of me. Too close to the screen to watch the movie. They charge for it anyway, and it will probably be grotty.

World Airways food seems even worse than the average airline rubbish.

#### *3. Baltimore Washington International*

Coming in to land, and the condensation forms on the overhead lockers, and pours on the people sitting in the front rows. That's me. Raining inside the aircraft. Brilliant.

300 people pour off the aircraft and wait at the immigration desk. Two IO's, giving all non-nationals a really hard time, especially if they are non-caucasian. Three quarters of an hour later, my turn. Through in about ten seconds. What happened?? Now I have to run for the connection to Atlanta. Collect baggage, more questions from customs. Check baggage in. Get on plane. Relax again. Flight delayed anyway. Local time is 8.00 pm, body trying to tell me that it is 1.00 am.

#### *4. Atlanta, June 9th*

As you step off the plane onto the jetway, it feels like they have a warm air heater switched on. Cool again inside the terminal. Collect baggage, step outside.

Brain refuses to accept that it is both dark and over 90 degrees. Over 90% humidity too. Hot and sticky. Horrid. Catch airport shuttle bus at 9.30 (2.30 am). Get to hotel at 10.00 pm.

"I have a room reserved, the name is Stoughton, Nicholas Stoughton."

"Mr Anthony Stoughton?", pronounced "anthhhh owny".

"No, Nicholas. From London. I work for a company called the Instruction Set. My reservation was originally made in the name of Boyd Roberts."

"Let me check that. How do you spell Boyd?"

"B-O-Y-D, surname Roberts."

"Just a moment..."

Ten minutes go by. I am falling asleep on the counter.

"I have no record of a Mr Robert Boyd. Just a Mr Anthhhowny Stoughton, from a company called the Instruction Set in London England."

"That's me...you got the name wrong, but there aren't too many Stoughtons working for the Instruction Set right now." Notice I am getting into the americanisms..."right now". Yuck.

"It says here on the reservation form that you will be arriving late."

They're clever these Americans, they can read. I guess you could say 10.00pm is fairly late to check in.

"We have a small problem Anthhhhhowny."

"The name's Nick, but what's your problem?" Not my problem, note.

"Well, when you hadn't arrived by 6.00pm, we let the room to someone else. There is this Southern Baptist Convention on right now, and the city is very busy. We do have another room that you could have though."

My body assures me that it is 3.30 in the morning. Travelling always makes me tired anyway. I really fancy just going to sleep. So I ask

"What's the catch?"

"It really is a very nice room, Sir, very spacious. One of our parlour (parlor??) rooms. But it doesn't have a bed."

Oh no.....

"Of course it doesn't cost as much."

Since I am not paying the bill at the end of it, this is a small consolation. And with the town full of the Suuuuuthern Baaaaaptists, I'm not going to find it easy to get a room elsewhere at this time of night.

"I am intending to be here until Friday. I take it that you will be able to get me a proper room tomorrow?"

"Oh yes, Sir, of course."

"I'll take it, but only under extreme protest, and because if I don't find somewhere to sleep pretty soon, I would sleep in the corridor if necessary. Where is this room without a bed?"

"Well, it does have a folding bed thing, and it is on the 67th floor."

I discover that this is the tallest hotel in the world. And I'm practically at the top. The room actually is huge, and the folding bed is not too uncomfortable. Sleep at last.

#### 5. Tuesday, 9.00

This hotel is full of Southern Baptists. They try to convert you in the elevator (getting used to the jargon...those things that whisk you up to the 67th floor are not called lifts here).

There are about 50 SB's queuing for breakfast. Give this up and go to the Hilton.

Better do the registration bit before eating, just in case there's problems.

Go to counter marked "Pre Registration", because I was pre-registered. Or at least I thought I was.

"I'm sorry, we have no record of a Mr Stoughton."

Oh boy, here we go again.

"Could you try Boyd Roberts, The Instruction Set."

"I'm sorry, nothing for him either. You'll have to go to the On Site registration desk."

Well, not much I can do besides go to the on site registration. Another hundred bucks on the expense claim.

"What tutorials do you want to go on."

"System V internals, please."

"Can I see your source licence?"

I'd left so fast on Monday that I'd clean forgotten to bring a copy. But one should have been sent with the pre-registration, so I wasn't bothered till now.

"I am sorry, I have forgotten to bring it. Anyway, you were sent one a copy with my pre-registration that seems to have got lost. Couldn't you look it up?"

"Our lists are very out of date, I'm afraid. But I'll take a look." He looks under "I" for "Instruction Set". No record.

"But I have got a source license. Would you like me to quote some lines of source from memory or what?"

"I'll take one more look. The Instruction Set, I suppose it could be under "T"...ahh...yes...The Instruction Set, Technical Contact Andy Rutter."

"That's the one. Now can I go to the tutorial?"

"I guess so."

Great. We are finally getting somewhere. Now I can go and eat some breakfast.

#### 6. Tuesday, Tutorial Session

Well, avoided the eggs because I couldn't face the wide variety of choices of cooking method, each with strange names. Now for some real work at last.

Tutorial #T3. UNIX System V Internals, with Maury Bach and Steve Buroff. An interesting mixture of some really simple stuff (how system calls work), and some extremely useful explanation of some of the SVr3 features, the File System Switch,



Remote File Sharing, Streams and the Transport Level Interface. About 75% extremely interesting 10% interesting and 15% boring.

Steve Buroff defined a "property" of a system to be a combination of "a feature and a restriction". A good definition! One property of shared libraries is that they are statically linked. This means that every shared library must have a fixed address range. AT&T will manage the allocation of address ranges for the 3B machines. What will happen for the other machines? Will AT&T be capable of managing this sensibly? (Shame on you Nick for doubting AT&T's management capabilities.)

RFS sounds good till you look underneath. Despite the File System Switch's ability to allow non-standard file system types to exist, RFS seems to have ignored this in slavish adherence to the SVID file standards. This is fine, but virtually prohibits sharing of files with non-UNIX file systems.

They did funny things to *copyout* too.

### 7. Tuesday Evening

It is raining. Not just drizzle, or light rain. This is a monsoon. Do they have monsoons here? You bet they do. About 6 inches of rain an hour. Sit in the Hilton waiting for a break. Race back to Westin Peachtree so that I can book Nigel in before the 6.00pm deadline. Find out that he has already made it. Start the hunt for him. Not in bar, but had a quick one whilst looking. Back to the Hilton. No sign. Quick check in the bar over a cocktail. Definitely not there. Go for a meal instead. These Americans definitely like to eat lots of food, and extremely tasty too; pity about the beer.

Back to the hotel, move to new room, two floors down. It still takes 3 minutes for the elevator to cover the 65 floors, and the Southern Baptists can get an awful lot of praise into 3 minutes.

### 8. Wednesday Morning

Finally find Nigel by means of calling his room. Try for breakfast in the hotel. Taxi to the Hilton. Well, maybe it is only two blocks, but it is awful hot out here.

First day of the conference proper. Nigel has had this idea, developed over his previous evening's entertainment with the other EUUG people. "Run the *errno* contest that went down so well in Florence here." The contest is announced during the opening session. Murmurs of approval from the masses.

The keynote address from Jon Bentley is entitled "Pictures of Programs". Not surprisingly, quite a lot of it is about graphics.

Coffee time; find a box for the *errno* contest. Tilbrook tries his hand at sign writing. Not too bad. Now sit back and wait for the entries. Will anyone beat ENOTABACCO, read on an empty pipe?

Mike Hawley presents his MIDI paper, this time round it is a history of Western Music since the Middle Ages, complete with numerous taped examples of everything from Gregorian Chant to some contemporary stuff (can't remember who by though!) Fascinating, but impossible to describe on paper.

Peter Langston follows hot on Mike's heels. Aside from writing games, Peter's main *raison d'être* is music. Music, I believe, of any sort whatsoever. Computer generated music most of all. Working at Bellcore he has of course access to all sorts of nice bits of hardware, and telephone equipment. So via a complex arrangement of Dec-

Talk speech synthesisers and music making synthesisers, anyone with US standard MF dial phone can dial in on (201) 644-2332, and listen to Eddie and Eedie taking you through some generated-on-the-spot music in their comical voices. These machines are truly tremendous. After the initial period, you really don't realise that this is computer generated. The voices have all the right intonation and inflexion, the music is spot on. Simple algorithms imitating a facile, unimaginative and slightly lazy guitarist playing random riffs to fill in bits ensure that the music remains interesting. So interesting that no-one would have minded if he had talked, and played us recordings of Eddie and Eedie, all day.

### 9. Wednesday Afternoon

Quick lunch from the rather good serve-yourself restaurant in the Hilton foyer, talking to Melvin from BT. These Brits get everywhere.

Back upstairs, the first of the *errno* entries have been submitted.

ENOGOOD	Invalid system error
ECHERNOBYL	Connection melted
EREALSOONNOW	Feature not yet supported
ECREAT	Missing vowel

About 20 so far.

Networks 1 in the Grand Ballroom, Secure Networking in the Sun Environment; A Framework for Networking in System V; and OSI and TCP/IP Protocols on a UNIX System V. Sounds good.

Secure Networking turns out to be all about public key cryptography and the DES encryption standard. Not as interesting as I'd hoped. Maybe I'll read the paper before going to the presentation next time.

The Framework for Networking in a System V presentation basically described the SVR3 TLI implementation. Good stuff, but there isn't SVR3 to hand yet.

The French described what they had done to some weird piece of equipment they call the SM-90 to put OSI and Internet protocols into their System V. A large part of this exercise was to put the 4.2BSD Socket stuff and most of the networking bits into their kernel. Read the paper.

Tea break. Another 30 entries in the *errno* box. Boy, this is beginning to take off. We already have about the same number of entries as we got for the whole of the Florence contest.

ENIH	There's a better way to do that
EXPORT	Feature restricted outside the US
ENOWARP	Out of Dilithium Crystals
EACDC	Wrong type of socket
ETHIOPIA	Out of resources
EGODS	Too many Baptists
EMAIL	Junk mail detected
EAT	You've been hacking too long, time to ...
E	2.71828...
EFLAT	String out of range

**EBADDOG** Pointer on carpet

The next session looks a bit heavy, Operating Systems 1 or Tools. Tools contains a talk on Pathalias. Definitely to be missed!

Give up after the first presentation. After all, I can read the proceedings later. Let's go and talk to all those interesting people about. Awful lot of Europeans about making the most of the tea.

Already another 20 *errno* entries.

Nigel has decided to move hotel to the Hilton. It is cheaper and easier. I think I will follow. Check out of the Westin Peachtree. Taxi to the Hilton. Cabbie doesn't like having to take me just 2 blocks. Stiches me up with the fare, so I don't tip him. Looks violent. Run for cover inside the hotel.

Only on the 24th floor here. Peanuts. Good view out over the hotel pool though, which is outside on the roof of the 10th floor, and directly below my window.

#### 10. Thursday AM

Another 50 entries in the *errno* contest. This is going to be fun.

<b>EFIXED</b>	No children
<b>ECONSTIPATED</b>	Operation would block
<b>EAT</b>	The computer is going down
<b>EMARCOS</b>	No longer supported
<b>EVITA</b>	Don't cry for me Argentina
<b>EBUNNY</b>	Multi-hop access attempt
<b>ELUSIVE</b>	Invalid pointer
<b>EUNUCH</b>	Unable to fork
<b>ENOTHEAVY</b>	E's my brother
<b>EIMPOTENT</b>	Unable to fork
<b>EMORTAL</b>	Can't kill process
<b>ELABORATE</b>	Too tense
<b>EIA</b>	Standardization violation
<b>ECRUFT</b>	Bit rot in program
<b>E&lt;unprintable&gt;</b>	Obscene kludge in program
<b>ENOUGH</b>	Time limit expired
<b>EEEEEEEE</b>	infinite loop in program
<b>EUUG</b>	Intercontinental junket

Lots of network and distributed files systems stuff today. Greg Chesson, chairing one of the Distrib FS sessions coins the term *Doofers*. This is an immediate winner. We immediately have people talking about *rufus*, *goofers*, *newfus*, and so on. Andy Rifkin does his Rufus talk. He's been to charm school recently, and almost makes it sound plausible.

#### 11. Thursday PM

Good lunch with the guys from Lachmann. Feel I ought to be present at the paper given jointly by them and us. Mike Wilde does a great job.

Ronald Hughes' doofer is called *truefus*!

Tea, and another 50-60 entries in the *errno* box. They are coming in thick and fast now. After tea, check out the mail session. Craig Partridge, of MMDF fame, elaborates plans for domain based mailing everywhere.

An interesting scheme for a /mail file system under the V8 FSS. Interesting, but impractical. Nice user interface though.

### 12. The Conference Dinner

Got talking to a guy working for AT&T who used to work in the UK. Offers me a lift to the conference dinner. We get lost, and drive around Atlanta for a while, eventually finding the Georgia Railroad Depot, a small, shack like building in the middle of a huge parking lot (I'm really getting to grips with this language).

Inside, things are really going strong. There is plenty of booze, even Heineken Lager which is somewhat better than the American gnats' water. And food! Wow, whole pigs, corn on the cob, mountains of food. The whole place is set up like a fair ground, with side shows, the lot. Won vast numbers of plastic toys. Getting decidedly merry.

The party finishes around 9.30, and it's back to find the hostility, sorry hospitality, suites that are still going. DEC play host, and the European committee (Nigel, Jean, Mike O'Dell, Jaap, DT and me) get down to judging the *errno* contest. There have been some 800 entries. 600 are weeded out immediately. They are too sick, too obvious or too unfunny. Sorry guys. That still leaves 200. After three more passes, each getting harder to throw things out of, we are down to a top 14

EDINGDONG	The daemon is dead
ELECTROLUX	Your code could stand to be cleaned up
ELECTROCUTION	Attempt by finger to reach socket
ENOTONHORSE	Mount failed
ENOPHONEBOOK	Directory does not exist
EEMILYPOST	Wrong fork
ENOCONTEST	The judges decision is final
ENOARMSCONTROL	Silo overflow
EFLAT	File system needs tuning
EGODOT	Endless wait
ECRAY	Program exited before run
EIEIO	Bug bug here, bug bug there
ENOSTRADAMUS	Predicted result
EMRED	A host is a host from Coast to coast, and nobody talks to a host thats close unless the host that isnt close is busy hung or dead

EMRED actually wins it, and must be sung. Special efforts in training singers must now be undertaken. It should be pointed out though, just what MR ED is all about. Mike O'Dell quotes:

Well, the show is about a horse, Mr. Ed, owned by one Wilbur Post, an architect and developer. It is set in some generic California (southern, seems like) location. The central gimmick is that Mr. Ed can talk, but only to Wilbur, who of course can't tell anyone else lest he be placed in the booby hatch. An episode I saw today (reruns) had Wilbur ghosting for Mr. Ed writing his memoirs "Confessions of a Palamino Playboy". Wilbur was ostensibly the author writing under "Mr. Ed" as a pseudonym. The conflict revolved around the book containing some non-fictional facts that clearly happened as portrayed in the book, but that Wilbur could not have known about, but Mr. Ed did.



Finally, my best recollection of the original lyric for the themesong.

A horse is a horse, of course, of course,  
and no-one can talk to a horse, of course,  
that is of course unless the horse  
is the famous Mr. Ed.

(reprise)

People yakkity-yak and speak  
and waste your time of day,  
but Mr. Ed will never speak  
unless he has something to say.

Go right to the source and ask the horse,  
he'll give you the answer that you endorse,  
he's always on a steady course,  
talk to Mr. Ed.

(spoken in character)

"I am Mr. Ed."

For you film bufs, you might will remember a series of B movies with Donald O'Connor and Francis the Talking Mule (whom Mr. Ed claims as a relative) which came out in the late 40's and 50's (a couple even starring an actor now playing as the President of the USA!). The show is clearly a knockoff.

It was a weekly series and ran for several years.

### 13. *Friday*

The discussion now is about the announcement of the *errno* contest winners. Should we do a stage presentation of the European entry **ENOGLIDER**, Pilot fell out? In the end we decide that this is just going to be too complex. Everyone is coached in singing **EIEIO** and **EMRED**.

The prizes are obtained during one of the morning sessions: several varieties of Champagne.

The morning session, and more on shared libraries. Some neat stuff on Real-Timing UNIX systems by adding pre-emption to the kernel at critical points. It looks impressive.

Final lunch, then off to the third Operating Systems session. Some interesting stuff on as a virtual machine environment, in other words, using to develop portable operating systems. Nothing too earth shattering though.

### 14. *Time to go home*

Finally it is time to leave. What have I learnt? Was it all worth it? Will I get rained on inside the aircraft on the way home? Will I ever remember that my name is not An-thhhh-oww-ny, but Nick?

Well, I can safely say that there was lots to learn, and I managed to pick my way between getting bored stiff attending every session, and having a great time talking to legendary people. Lots of good ideas picked up.

Definitely worth it! (Especially keen on the roast pig!)

I came back on British Airways, and their cabins don't leak.

```
(inset) login: anthony
Password:
login incorrect
login:
```

### 15. Some of the *ERRNO* Entries

There were around 800 entries to the *errno* competition. It would take me weeks to type them all in, but here are some of the top 300 or so.

E	2.718...
E2BRUTUS	Init killed by an adopted child
E2MANY	Too many error codes
E<unprintable>	Obscene kludge in program
E423	Addictive overflow
E<unprintable>	Obscene kludge in program
E=MC2+1	Illegal units conversion
E?	DMR system error
EACDC	Wrong type of socket
EAGLE	Disk full
EAR	Please repeat question
EASTER	Autoreboot in 3 days
EASY	Cray doing an infinite loop
EAT	Data file munched
EAT	The Computer is going down
EAT	You have been hacking too long, time to ...
EATME	Cannot mount face
EATME	request bytes
EATT	Legal fees exhausted
EATT	Running Sys 5
EATTABOY	Nice try - try again
EBAD	Bad error (not good). Naughty Naughty.
EBADDOG	Pointer on carpet
EBADMUSIC	Warning - muzak playing
EBADTASTE	Compiling an operating system written in PASCAL
EBANKRUPT	Out of cache
EBAPTIST	Busy hands are happy hands
EBB	Data flow reversal
EBCDIC	Dialect unknown

EBCDIC	Non ascii stream
EBCDIC	Non portable character comparison
EBFOREI	Invalid syntax
EBSB	Running BSD
EBSB	Your tape will be shipped real soon now, we promise
EBUNNY	Multi-hop access attempt
EC00Y	JCL error
ECHAOTIC	Chaotic or random error
ECHERNOBYL	Broken pipe
ECHERNOBYL	Connection melted
ECHIROPRACT	Disk problem
ECHO	Duplicate argument found
ECIA	You're not allowed to know
ECLAMUP	Shell Quit
ECONF	Too many sessions
ECONSTIPATED	Operation would block
ECRAY	Program exited before run
ECREAT	Missing vowel
ECROSSDRESSING	Violation of strong typing
ECRUFT	Bit rot in program
ECRUSADE	Religious error (SVRn)
ECT	Addressing, I/O, Wrong Command etc
EDDIE	Thank you for making a simple error-code generator very happy ...
EDFS	Not a tty
EDINGDONG	The daemon is dead
EDMR	A host is a host from coast to coast, and no-one can talk...
EDOofs	Beating dead horse
EDOofs	Overload
EDOWHAT	Operation unknown in Texas
EDRANO	Sync failure
EDROUGHT	Stream failed
EE	Stoned EE grad student code found
EE6NONE	Kernel non-existent
EEC	Common Market - no parity

EECHOO	Bless you
EECK	Mice in machine room
EECS	Invalid department
EECUMMINGS	Case Translator Finished
EEE	Center of terrorist activity
EEEEEEEE	Infinite loop in program
EEEEEEEEEE	Speaker too close to microphone
EEEEEEEEEEEEEEEEEEEEEEEE	Infinite loop detected
EEK	Dead mouse
EEMILYPOST	Wrong fork
EENYMEENYMINYMOE	Scheduler can't decide which process to run...
EFAULT	Earthquake
EFGHIJ	Long time no C
EFIXED	No children
EFLAT	File system needs tuning
EFLAT	String out of range
EFLOOD	Overflow in stream
EFORK	Routing table full
EFORK	Too many forks
EFREETRADE	Setquota failure
EGAD	Got another dump
EGAD	Surely you jest
EGADS	Sudden realisation were running VMS
EGADS	The system is astounded
EGGONYOURFACE	Memory fault: core dumped
EGOAWAY	System is busy now
EGODOT	Endless wait
EGODS	Too many Baptists
EGOEDEL	Maths Argument Undecided
EGOOD	Good error (everyone should make(1) one)
EGREP	String not found
EH	Canadian user error
EH	Say what?
EHEADACHE	Connection request denied
EHEADACHE	Dual processors not tightly coupled



<b>EHTRAP</b>	Hack ...
<b>EIA</b>	Standardization violation
<b>EIBM</b>	User error. Fix the problem and recompile
<b>EIEIO</b>	Bug bug here, bug bug there
<b>EIEIO</b>	Farms in Berkeley
<b>EIFORGET</b>	Out of memory
<b>EILLEGAL</b>	Illegal error (failable)
<b>EILLITERATE</b>	Can't read or write
<b>EIMPOLITE</b>	Bad fork
<b>EIMPOTENT</b>	Unable to fork
<b>EINF</b>	Infinite loop
<b>EJIHAD</b>	Religious error (BSD4.x)
<b>EKEN</b>	You are not expected to understand this
<b>EKNOCK</b>	Who's there
<b>ELABORATE</b>	Too tense
<b>ELAPSED</b>	Sins undeclared
<b>ELASTICK</b>	Clock needs rewinding
<b>ELAWFUL</b>	Lawful error
<b>ELECTROCUTION</b>	Attempt by finger to reach socket
<b>ELECTROLUX</b>	Your code could stand to be cleaned up
<b>ELF</b>	Tolkein ring passing error
<b>ELICIT</b>	Need more input
<b>ELIM</b>	Outside bounds of window
<b>ELUSIVE</b>	Invalid pointer
<b>EMACS</b>	Program too large
<b>EMAIL</b>	Junk mail detected
<b>EMAIL</b>	User is too big a flamer
<b>EMARCOS</b>	No longer supported
<b>EMEACULPA</b>	System error
<b>EMIRROR</b>	Hardware error - see <b>ESMOKE</b>
<b>EMISC</b>	None of the above
<b>EMISS</b>	Record skipped
<b>EMO</b>	Main processor overworked
<b>EMORDOR</b>	Name server bound
<b>EMORTAL</b>	Can't kill process

EMPOTENT	Pipe too soft
EMRED	A host is a host from coast to coast, and nobody talks...
ENAP	On wrong side of window
ENEEDWINDEX	Unable to do I/O to window
ENIH	There's a better way to do that
ENIXON	Tape problem
ENOAIR	Read on a full pipe
ENOARMSCONTROL	Silo overflow
ENOBOZOS	Stupid request
ENOCH	Race condition
ENOCHICKEN	Serious colonel problem
ENOCONTEST	The judges' decision is final
ENOCORN	Serious kernel problem
ENOCURRENT	Bad socket
ENODICE	Error in rand
ENOENO	Musician not found
ENOENO	Too intense
ENOERROR	Just kidding
ENOFUID	(Cray 2 only) CPU fluid level low
ENOGOOD	Invalid system error
ENOGREYHOUND	Bus error
ENOH2O	Read on an empty pipe
ENOHOPE	System V
ENOKISS	Buss error
ENOKNOW	Definitely hardware or software error
ENOMAAM	Missing token
ENOMULTIHOP	You should use NFS not RFS
ENON	No errors. Just kidding
ENONCOGNISCENT	Something is wrong and I won't tell you what
ENONO	You can't do that
ENOONE	Both endian
ENOPAPER	Read on an empty file
ENOPHONEBOOK	Directory does not exist
ENORMOUS	Program too large
ENOSALT	Nuke error: core dump

ENOSAUSAGE	Serious link problem
ENOSELECTRIC	Not a typewriter
ENOSTRADAMUS	Predicted result
ENOTATE	Not in art gallery
ENOTCOMPLETE	Dial again or ask your operator for assistance
ENOTHEAVY	E's my brother
ENOTME	Not my fault
ENOTMYFAULT	Some other process generated the error
ENOTONHORSE	Mount failed
ENOUGH	Infinite loop skipped
ENOUGH	No ugly hostnames allowed
ENOUGH	Time limit expired
ENOUGH	Too many errors
ENOUNIX	"But that's not UNIX" (you have invoked a system call...)
ENOVICE	Not in Miami
ENOWARP	Out of dilithium crystals
ENOWAY	No way to list all these error codes
ENSA	No such error
ENUF	No more errors please
EOOEAA	Invalid hex format
EOOPS	Program lost
EORGY	Mount table full
EOUTOFGAS	Bus error
EOW	Stack fell over
EP&p6	String not null terminated
EPDP8	Obsolete CPU
EPLUNGER	Unable to flush
EPRIVATE	Can't tell you error
EPUBLIC	Everyone knows better
EPUN	That's not funny
EPUNT	Now what?
ERABBIT	Too many hops
ERASE	Race condition
ERATRAP	Female programmer induced bugcheck
EREALSOONNOW	Feature not yet supported

<b>ERELIGION</b>	A 4.2/SV system call has been issued on a SV/4.2 system
<b>ERG</b>	Requires too much effort
<b>ERROR</b>	? (the user will know what this means)
<b>ERROR</b>	Error in perror
<b>ERROR</b>	Error
<b>ESALTI</b>	Deadlock detected
<b>ESARTE</b>	No exit
<b>ESCAPE</b>	Trap failed
<b>ESCHWA</b>	Extended character set unsupported
<b>ESMOKE</b>	Software error - see <b>EMIRROR</b>
<b>ESMOP</b>	Code not written; lazy programmer
<b>ESP</b>	Error coming soon
<b>ESPLASH</b>	Cannot push stream module
<b>ESPOON</b>	Too many forks
<b>ET</b>	Cannot phone home
<b>ET</b>	Phone home
<b>ETATJOES</b>	Commercial implementation error
<b>ETC</b>	Misc. error
<b>ETCOMEH</b>	Tbuf fault. Flushing and returning...
<b>ETeenager</b>	Runaway child process
<b>ETERNAL</b>	Attempt to sleep on wchan for which wakeup will never...
<b>ETHER</b>	Try STARLAN instead
<b>ETHEWRONGCHOICE</b>	BSD system call
<b>ETHIOPIA</b>	Out of resources
<b>ETICKET</b>	Failure to stop at signal
<b>ETOOMANY</b>	Too many remote file systems
<b>ETOW</b>	Denver boot (wheel clamp)
<b>ETPHONEHOME</b>	Hang up
<b>ETRAP</b>	Rogue fell into its own trap
<b>ETTY</b>	Is a typewriter
<b>EUNICH</b>	/etc/dev not found
<b>EUNICH</b>	Mount request not possible
<b>EUNUCH</b>	Unable to fork
<b>EUSSR</b>	Usr core destroyed



<b>EUUG</b>	Intercontinental junket
<b>EUUG</b>	Too many groups
<b>EVITA</b>	Don't cry for me Argentina
<b>EWATERGATE</b>	Extreme tape gap
<b>EWHAT</b>	Unknown error
<b>EWOODY</b>	This LAN is your LAN, this LAN is my LAN...
<b>EX</b>	Old version detected
<b>EXORCIST</b>	Cannot remove zombies
<b>EXPORT</b>	Feature restricted outside of the US
<b>EYAWN</b>	Algorithm boring
<b>EYOYO</b>	System about to crash
<b>EYUCK</b>	Are you sure that you wanted to mount that VMS file system...
<b>EYUCK</b>	Pipe backed up
<b>EZ</b>	It would be faster to do operation on paper
<b>EZ</b>	Menu in use
<b>EZBAD</b>	Program erased all source and died
<b>EZMUCH</b>	Funny error (Inexplicable without laughing or giggling)
<b>EZRIDER</b>	Irregular machine cycles
<b>EZZZ</b>	Interrupted sleep
<b>EZZZZ</b>	Process asleep
<b>Ep&amp;P6</b>	(null)

## The Manchester Competition

*Peter Collinson*

*Secretary, EUUG*

### 1. INTRODUCTION

The traditional competition was run as usual during the Manchester conference. The idea this time was to think up a meaningful phrase to go with a UNIX acronym. There were very many entries written up onto large sheets of paper. I then had the dubious pleasure of transcribing these into my notebook. In the process I expect that I missed a few, here are the ones I got.

adb	A dreadful bug
adb	Another damn bug
adb	Antediluvian bug eater
arp	Any respondents please?
at	Accumulate trivia
att	Another terrible trademark
bc	Brilliant crasher
bc	Bypass core
bsd	Bad syntax deterrent
c++	Cannot add
cat	Compose a tune (at Belcore)
cb	Crash and burn
cc	Change channels
cc	Core dump generator
cd	Can't define
cd	Concoct data
cdb	Cannot detect bugs
chmod	Can hinder many other developers
chown	Can't have ownership work nohow
chown	Chase owner
ci	Correction impossible
cip	Clip important parts
co	Correction ordered
cpio	Corrupt process in operation
cpio	Pipe construction process in operation
cron	Can run only nightly
cron	Continually repeat obnoxious no-good doers

cs	Can severely hinder
cs	Can't and shan't help
cs	Complete sh
cs	Crash system and halt
cs	Crash system hardware
cu	Cockup
cu	Completely useless
cu	Convert to unary
date	Display a time estimate
dbx	Damn bad 'xperience
dc	Divide and conquer
dc	Dumb crashes
dd	Danish design
dd	Data destruction
dd	Destroy disc
DEC	Disgustingly expensive computersa
ded	Delete everything directly
delta	Delete target file
delta	Destroy every last trace archived
ditroff	Do it read only for fun
doc	Describe our computer
dsw	Disc scribbling wanted
du	Device unavailable
dump	Destroy user's modified programs
dump	Don't use my program
dumpS	Drinking unmoderately makes people SNORE
ed	Eject disc
ed	Extremely devious
EIO	Execute invalid opcode (microcoded machines only)
em	Evacuate memory
emacs	Eats memory and cpu superbly
emacs	Editor managed auxillary command system
emacs	Eventually makes all computers sick
fsck	Failed system call, Ken
fsck (-p)	foolish suggestions can kill (less painfully)

fsdb	For sex, dial Berkeley
gfs	Green fresh sausages
grap	Graphical replicated array processor
graph	Generalised rough analysis presentation heuristic
grep	Get ready to eat the processor
grep	Greatest request ever produced
grep	Guess random element and print
grep	Gurus recovery expert program
hhcp	Hardly hopefully connection process
iso/osi	Inside out/outside in
ln	Lindsay's Nfs (implementation)
lp	Lose printout
lpr	Lost printout repeatedly
lpr -m	Mail me that you've lost printout repeatedly
ls	Last supper
ls	Less sex
ls	Let's see
ls	Long story
ls -l	Lots of silly lines
m4	Slow route to nowhere
mac	Mouse and computer
mail	Make and infinite loop
mail	Mangle and incarcerate protocols
mail	Many annoying interruptions from layabouts
mail	Move applicable info long-distance
make	Make aspiration knotted exhumation
man	Male aid needed
man	Many ambitious notes
man	Minimum assistance to novices
mh	Monster hack
mknod	Yet another distributed file system talk (zzz).
mmdf	Multi machine destruct facility
mmdf	My message didn't find you
modem	Moderate outlay data exchange medium
mv	Make vanish

mv	Make volatile
ned	Never easily done
netnews	Notify every techno nerd every week, slowly
news	No-one ever writes sensibly
nfs	No fear, Sun
nfs	Not fresh sausages
nl	Not listed
notes	Networks only total erasure system
nroff	Nearly random output file formatter
od	Obscures data
od	Oh dear
osi	Ossified system interface
osi	Outsiders slip in
osi	Over simplified for idiots
passwd	Pick a seven symbol word, dummy
ps	Print and smear
ps	Print suspects
ps	Profanely slow
rcs	Random code segment
rcs	Random code shuffler
readnews	Really esoteric articles deleted; nauseating essays were saved
reset	Repair Emacs settings
rfs	Real fresh sausages
rfs	Realistically, far from safe
rm	Read manual
rm	Realise mistake
rm	Reinitialize meter
rm -i	Realise mistake (after impatient response)
rm -r	Read manual — religiously
rmdir	Randomly modify directories in root
rn	Remote network
rpc	Random protocol choice
rsh	Replicate superior hackers (BSD)
rsh	Restrain pSeudo hackers (Sys V)
sccs	Scramble code, crash system

script	Some complete rubbish, I probably typed
sdb	Should do it better
sed	Sex evolves desire
sed	Strips extraneous data
sed	Suffer endless delays
sendmail	Simulate eventual (notional) delivered mail and immediately lose it
sendmail	Smart enigmatic non delivered mail archiver
sendmail	Someone else not doing my interpretation logically
sendmail	Start endless new daemons making all infinitely late
sendmail	Stomp every naive, defenceless mailer article into lumps
sendmail	Sundry elusive notes delivered misaddressed and instantly lost
sh	Shan't help
sh	System halt
shutdown	System hung up today due to our work on NFS
sleep	Snoring loudly, eventually nags people
slip	Should link in properly
sort	Shift, organise, rotate — truncate
su	Suspend user
suntools	Someone upset Newcastle to outhype Lindsay's system
sync	Sorry your network crashed
sync	Success yet no chance
sync	Swedish yacht near Chernobyl
tar	Totally amazing results
tar	Totally arbitrary results
tbl	Trademark, Bell Labs
teco	Type Esc to complete operation
tee	Tried everything else
telnet	This ether Lan needs extra transceivers
termcap	Terminal emulator requires mangled cursor addressing protocol
tip	Transfer incomplete programs
tr	Text rape
tr	Too risky
tr	Tortuous request
troff	Try running our fast formatter
uniq	Uniq neglects iterative quantities

unix	Umpteen new implementations expected
unix	Utterly nauseating in 'xtension
UNoltmaiXense	Think about it!
uucico	You see.... I see..... Oh!
uucp	UNIX under constant pressure
uucp	Useless under protocols
vi	Vacillating irradiator
vi	Very insensitive
vi	Violate intuition
vi	Virtually impossible
vi	Virtually incomprehensible
vi	Virtually intelligible
vi	Visually inept
vnode	Virtually no design
wait	Wow! Another idiot time consumer
whatis	Want help: ARGH this is sick
yacc	Yell at compiler constructor
yacc	Yet another core dump created
yacc	You are completely confused
yacc	Yum, another CPU cycle

The judging committee had a hard time with so many entries. Entries which were thought to be worthy of special mention were:

cc	Can't cope
df	Disc full
emacs	Eight megabytes and continual swapping
time	Totally inaccurate measure of execution
unix	Unlimited new interfaces 'xpected

The winning three were:

sh	Sans histoire
sendmail	Seems every new domain means adding information locally
patch	Please apply this clever hack

## Uniforum, January 1987

*Philip Peake*

*Axis Digital*

### *Washington DC*

This year, UNIFORUM chose to hold the exhibition and conference in Washington DC, which is somewhat closer to Europe than is Anaheim, which was the site chosen last year. Washington is somewhat more impressive than Los Angeles, but even so, still seems to lack something when compared to European cities.

I arrived there on Sunday evening. There was little to do, since I was tired after the flight, and didn't have much money left out of the 50 dollars I had taken with me (the taxi fare from Dulles airport to Washington is 30 dollars). Next morning, I planned to visit a bank and get some more money using my VISA card.

However, next morning was "Martin Luther King day", which basically means that everything (and in particular, banks) is closed. It was also pouring with rain. I walked to the White House, and was suitably disappointed by its unimpressive size.

I looked into the METRO, which seemed impressive, but since the tickets are sold by machines, and I didn't have the correct coins or notes, I couldn't try using it (and I couldn't change the notes I had, because everything was closed ...).

### *Washington Conference Center*

The conference and exhibition was held in the Washington Conference Center, and a bus was available to take collect people from the hotels and take them directly there. Before going to the center, I found a bank and managed to get some money. They wanted two means of identity before they would give me the money, so I gave them my french driving licence and french ID card (*carte de sejour*), which seemed to confuse them a little, but eventually they gave me the money.

Having missed the first bus, I now took the METRO, which is impressive. However, you can't help the getting the feeling that it is maybe designed more as a public bomb shelter than purely a METRO system.

The first day of UNIFORUM was reserved for tutorial sessions, since these were at extra cost, and not on subjects which seemed particularly interesting, I used the day to register for the conference, and to collect the relevant paperwork (lists of exhibitors, entrance badges, conference proceedings etc.). I then made contact with various other people attending the exhibition.

### *The Exhibition*

The next day the exhibition opened. As usual with UNIFORUM, it was a large exhibition. It took almost all of the three days of the exhibition to visit each of the stands and to see what they had to offer. The overall impression was that there was little new on offer. Although there were one or two vendors who seemed to have heard of Europe, and had started to "Europeanise" their products.

Europeanisation seems to mean that the product will handle 8 bit characters, and that there are various versions of the product with menus and (sometimes) diagnostic message in French or German. If you asked awkward questions about spelling checkers, or hyphenation tables for these things, they were always "Planned to be available in the next release".



Europe was represented by a stand organised by the X/OPEN group. I had the distinct impression that most of the visitors to this stand were, in fact, Europeans. But the name of X/OPEN has apparently been heard in the USA, since during the conference AT&T announced that they were joining the group. The other "rival" standard POSIX, or IEEE 1003.1, also announced during a presentation of their aims, objectives and future directions, that they were to work with X/OPEN in an effort to make the two standards converge, and eventually become a single standard. This is very good news, since it had seemed until recently, that there were going to be at least 3 different standards, SVID, X/OPEN and POSIX. Now it seems possible that there will be only one, which can only be good news for end users of UNIX systems.

Some of the largest stands were: SCO, AT&T, SUN and IBM. The SCO and SUN stands both seemed to be doing good business, but the other two seemed to be staffed by bored looking people who stood around in little groups talking to each other, waiting for people to venture onto their stands.

#### *The Conference*

This section should perhaps be called "The Conferences", since there was a complex system of parallel session. Some of the sessions being available only if you had paid the appropriate fee.

My personal opinion is that this could have been much simplified, by refusing to accept some of the presentations, which were little more than sales talks. I found it difficult to know in advance if I was going to hear something interesting, or just a description of someones latest product.

However there were some interesting presentations, amongst which were:

- UNIX utilities in ROM for the HP Integral PC
- MINIX - By Andrew Tannenbaum
- UNIX for real time (HP)

I would recommend finding a copy of the proceedings and reading these. There are others there which are probably interesting according to personal taste.

The MINIX system was perhaps the most talked about item on the timetable. It certainly helped to keep the Prentice Hall stand very busy. They had an IBM PC running MINIX on show, and at the user level it really does seem very much like a V7 UNIX system, and its performance on a floppy based machine is quite acceptable.

#### *The Snow*

No account of the 1987 UNIFORUM would be complete without mentioning the snow. It started one morning, with just a light cover. Then it started to snow harder, and it continued all day. By the end of the day 18 inches (40 cm) of snow had fallen. Needless to say, the shuttle bus drivers who were supposed to take us all back to our hotels had gone home a long time before the end of the day's proceedings at the conference center. There were long queues waiting for taxis. The taxi drivers filled their cabs to capacity. Apparently, local law permits them to double their fares when it snows. When I eventually got into a taxi, I (and all the other occupants) found that not only was the fare double, but that rather than it being shared between all the occupants, our driver wanted us *all* to pay double fare. Since it was still snowing hard, there was little choice ...

When it snows, not only do the taxi drivers become mercenary, but all the restaurant staff go home early. That night, Washington was full of pitiful small groups of UNIFORUM attendees wandering the streets looking for somewhere to eat.

*In Retrospect*

Looking back at the event, the most important items seem to be as follows:

— Europeanisation.

Most American companies now seem to recognise that Europe exists. Even if they don't currently have anything directly relevant to Europe most say that they plan to include us, and our peculiarities such as strange character sets, in their calculations for future products.

— Standards.

The convergence towards a single standard can only be a good thing. The SVID was a good idea. It had the problem of being tied directly to AT&T. X/OPEN was a better bet, being a consortium of manufacturers. POSIX is supposedly completely independent of manufacturers. With AT&T becoming part of the X/OPEN initiative, and the announced convergence of X/OPEN and POSIX there is now a real chance of a universally recognised UNIX standard.

— MINIX.

Andy Tannenbaum has done what so many people have said they were going to do, but never did. He wrote a UNIX look-alike from scratch. Not only that, he has made the source available with fairly minimal restrictions on its use.

It must be recognised that this system is *not* a replacement for a modern UNIX system, but could form the base of a future general purpose OS with most of the functionality of UNIX.

# EUUG

European UNIX® systems User Group

## Are YOU a UNIX® User?

If so, you can dramatically extend your knowledge of UNIX systems and share your views and expertise with other UNIX users throughout the world by becoming a Member of the

European UNIX systems User Group

The EUUG is a non-commercial organisation which is in-being to:

- \* Enable regular contact between UNIX users
- \* Provide a source of unbiased information
- \* Provide forums for discussion by means of exhibitions, conferences and symposia
- \* Improve knowledge of UNIX and thus improve the System itself

In addition to a quarterly **Newsletter**, the EUUG makes available its EUnet **electronic mail** and news services to all Members; supports the activities of its National User Groups throughout Europe; provides a **Software Distribution Service**; and has a large number of specialists within its Membership to help users with their problems.

Its **International Conferences** are now a very important part of the UNIX scene with future events scheduled to take place in Finland and Sweden (May 1987); Dublin (Autumn 1987); and London (Spring 1988).

If you want to play a bigger part in promoting the cause of the UNIX Operating System and — at the same time — benefit from the many services offered by the EUUG, then contact the Secretariat NOW for further details.

European UNIX® systems User Group

Owles Hall, Buntingford, Herts SG9 9PL, England Tel: 0763 73039 Network address: euug@inset.uucp

UNBIASED INFORMATION,  
ELECTRONIC MAIL,  
SOFTWARE DISTRIBUTION,  
INTERNATIONAL CONFERENCES



## Notes on the Birth of The UNIX Cult

*Peter Collinson (local name)  
AS/103/108/121/110 (Galactic designation)*

*Galactic Cult Investigation Team,  
Canterbury, Kent  
(Small Island off Continent 3, Sol 3)*

### *Editor's Note*

This paper was first presented at USENIX in January 1987.

*Notes from some recent archeological findings on the birth of the UNIX cult on Sol 3 are presented. Recently discovered electronic records have shed considerable light on the beginnings of the cult. A sketchy history of the cult is attempted.*

### *Background*

The UNIX cult is widespread across the Galaxy now and the surprise discovery of some ancient files in the archives of Intergalactic Brain Machines on Sol 3 triggered the dispatch of an inter-disciplinary investigation team. The files are extremely extensive, occupying all of a small island off the coast of Continent 3. It transpires that the island was taken over by Intergal in the aftermath of the Corporate Wars which plagued Sol 3 some centuries after the birth of the cult.

The team were asked to find out the original meaning of some of the incantations used in UNIX religious practice and also to shed some light on what it all meant at the start.

We should take this opportunity to use the ancient prayer:

UNIX is a trademark of AT&T in the USA and other countries.

Earlier versions of this prayer do seem to exist, it is unclear why the form of words altered. "AT&T" was the Corporation where the Creators of the cult worshipped. The Corporation totally disappeared in the wars and many of its original records were either destroyed or altered by the victor in an attempt to "re-write" history. The placement of the country USA on the four continents has been lost.

### *The Gurus*

There seems to be still no trace of the original Creators of the UNIX cult, so we start our examination of the records with a group calling themselves the *Gurus*. The etymology of this word is not quite clear but it does have associations with religious teaching and the High Priests of UNIX are given that title today.

Extract from electronic phone tap of someone nicknamed "dsw", believed to be Daniel Stuart Wilson:

"Of course, we were all isolated in the Version 6 days. When we changed things in the kernel we were on our own. There was no-one to phone to scream for help, in many cases there was no-one on the same site who you

could discuss the problems with. We just had to get down and read that C. Sorting things out yourself was painful but you benefited in the long run. You became a better software engineer. After a bit you became a Guru and could lead others. This is largely bluffing — given a new situation and a knowledge of UNIX you could extrapolate the problems easily and seem to be very clever. It didn't actually matter what you had or had not done, it mattered that you *appeared* to have done a lot and could talk confidently about RK05's, PDP-11/45's, typing *chdir* rather than *cd* and a myriad other things which showed that you were a Guru."

This extract shows clearly that the cult grew in small pockets across the globe. In each centre, a few individuals were given the task of installing the paraphernalia of the UNIX cult and of converting others to its use. In the beginning, it seems that these individuals had little or no contact with each other; curiously, this appears to have strengthened their ability rather than weakening it. Other extracts from the archive indicate that the early practices of the cult were small and simple making it easier for one person to grasp their full meaning. As the cult grew, the practices became more complicated and the understanding harder.

The term *software engineer* refers to a person whose task it was to feed instructions or programs into the primitive versions of the Overlords which were extant at the time. Judging from the emphasis made in the records about the need to generate "correct" programs, this was obviously an artistic task requiring considerable expertise. The reference to "C" implies that there was some official form of speech or perhaps a special religious language which was used to convey instructions.

It is not clear whether the "RK05" and the "PDP-11/45" were the names of the Overlords or some ancillary equipment associated with them. However, the word *cd*<sup>1</sup> meaning "to move from place to place" is still used in some arboreal societies on the planet US/115/110/105/120.

#### *The User Group*

In current UNIX religious practice, the term *User Group* is used to refer to the congregations at the Hologram Services. It seems that after a while the early practitioners of the cult began to have meetings (where people actually appeared together in the same room). Why these were called "User Group" meetings is unclear, since the early meetings were attended by Gurus and not by Users. At the time, a *User* seems to have been a derogatory term for the un-initiates. However, the head Guru at a site was given the title "Super-User"; perhaps this was to hide the evangelical nature of the Guru's task. The reason for the early meetings was mostly to allow Gurus to inform each other how best to perform religious conversion and how to get the most converts the fastest by *improving the Service to Users*.<sup>2</sup> As the word spread, the meetings grew in size, expense and quality of surroundings. They proved to be an exceptionally good way of upgrading low level Novices into higher level Gurus. This is a reflection on the early religious books which were aimed at Gurus and were often way above the head of the Novices. Novices were encouraged to attend the meetings to gain by word of mouth what they could not gain from the literature. At these later meetings, the nuts and bolts of UNIX practice were still discussed because many of the Novices were either acting in a support role to Gurus or planning to become evangelists themselves. After a

1. Pronounced *see-dee*

2. This is a contemporary term.

while, the Gurus got bored with discussing *inodes*<sup>3</sup> and other topics began to creep in. The meetings gradually altered in character, with the Gurus attending because they wished to see the other Gurus; and more and more Novices attending in the vain hope that they would learn something. The cult had spread so far by now that it was profitable for Corporations to become involved in the selling of UNIX paraphernalia and religious goods. Initially, these *Products* were advertised widely at the meetings. However, the Corporations often sent attendees with prepared scripts who were sometimes not even Users and who had little or no knowledge of intricacies of UNIX practice. The Gurus and Novices were dismayed. On their side, Corporations began to see that the word which they were spreading was falling on somewhat stony ground. This gave rise to the Great Split with a rival organisation being set up primarily aimed at selling and the original User Group concerning itself more with the cerebral activities of the cult. In the end, this was a good thing because the Gurus were able to start deriving benefit from the meetings again since there were now spare slots for educational talks. In fact, the rival organisation was wiped out in the Corporate wars because it had allied itself strongly with AT&T.

The User Group, then, provided some important functions. It supplied a forum for discussion of the practices of the cult. It provided a meeting place for the widespread Gurus who initially met to discuss their work but as time passed they went just to meet each other. It spread the word to Novices; and as the cult grew, it provided a place where new ideas on the direction which things should take could be discussed.

Another more hygienic method of worldwide communication grew out of the cult with the formation of the Network. This is discussed in the next section.

#### *The Communicators*

When the cult had grown to worldwide proportions, we begin to see the emergence of a global communication system — the "Network" or "Net". The activities on the Network can be deduced from the electronic archives but the material is so vast that scanning it for relevant information is proving difficult.

The majority of traffic on the network seems to have been communication between Overlords describing various error conditions. Here is a sample:

*<Various repeated unintelligible lines>*

**From: MAILER-DAEMON (Mail Delivery Subsystem)**  
**To: <uucp@a4los.uucp>**  
**Subject: Returned mail: Service unavailable**

— Transcript of session follows —

```
>>> DATA
<<< 554 sendall: too many hops (30 max)
554 <meqd>... Service unavailable: Bad file number
```

— Unsent message follows —

*<More of the same>*

There are very many other examples of the same type of message. However, these messages do place things in context — we now know that the communication system

<sup>3</sup> The meaning of this is totally lost.

was called *Mail* and from this we infer that the mechanism was intended to permit communication between people. In amongst all the Overlord messages we do find some files which appear to emanate from one person and be addressed to another. A high proportion of the sampled files were intended to probe the capabilities of the Network, often provoking an Overlord error message. We know this because the destination address is the same as the source; in some cases the subject is "Just testing" or something akin to that. This technique was perhaps used to investigate the ability of the Network to transfer files. We are left with a much smaller number of what might be termed "useful" messages. In many cases, these consist of personal trivia showing that the Network was supplying the useful social function of allowing people to make and maintain contact.

Other messages consist of hieroglyphics containing many braces "{" and brackets "("". This type of message shows some form of regularity in structure but syntax analysis is hampered by the presence of so many exceptions. It is possible that these messages contain portions of religious ceremonial or it has been suggested that this is the ritual language "C" and the exceptions are what were called "bugs". This last suggestion (by a post graduate student on the team) has been greeted with a certain amount of derision.

Mail messages have a recognisable format and can be distinguished from another type of message which occurs in much greater volume. It appears that these messages are part of a system called "The News" broadcast to many sites. The News permitted individuals to send a single message to many people across the world. A sample of the contents imply that the News took over as the main method of spreading the UNIX word when the User Group meetings ceased to be totally useful in this role.

Judging from the beginning of many News files, the News was split into many separate subject headings. Over large periods of time, we see the subjects come and go with abrupt changes in title occurring from time to time. The subjects do not seem to be confined to discussion of topics with direct relevance to the cult. There are many headings which just carry "talk" on various subjects. An analysis of the topics is being prepared as a background document since they fall outside the remit of the investigation.

It might be thought that the News would provide an excellent vehicle for the Super Gurus who must have existed by this time — but far from it — very few messages contain what might be termed confident information. Most seem to carry opinion which is contradicted in later messages. The evidence here is that most authors were Novices. The Gurus had either lost interest in spreading the word, or were simply too busy to wade through what must have been daily oceans of verbiage. It is also possible that Gurus just used the Mail to communicate, perhaps not wishing to impose their definitive opinions on the discussion with the thought that discussion is a healthy academic tool.

The Network spread slowly over most of Sol 3 with some areas being exempt because either they resisted conversion to the UNIX cult or were deliberately omitted for ideological reasons. A single News message from "kremvax", an otherwise silent site, seems to have been met with a storm of protest. This shows that the Network crossed political boundaries and proves the contention that Sol 3 was split into separate economic entities predating the rise of the Corporations.

#### *The Vendors*

The Corporations who were involved in the propagation of UNIX paraphernalia and religious goods were known as *Vendors*. The Vendors had a different view of the world from the Gurus, and this difference led to many schisms in the cult. The

Vendors and Gurus tried to maintain a separate identity at all times. For example, when the Vendors attended the User Group meetings they provocatively wore different religious vestment from the Gurus. They did this to make it easy for other Vendors to identify them and to allow their easy differentiation from the Gurus and Novices.

At the centre of the clash of opinion was a fundamental difference in the perception of "the User". The Vendors were always complaining that UNIX religious practices were not "User friendly". By this, they meant that the act of worship was hard to learn, and the rituals were cryptic. They wished their Users to only deal in simple concepts and never to learn more than the basics of the rituals. The Gurus objected to this view because they believed that the learning of cryptic rituals allowed the worship to proceed faster and that limited exposure to only the simple concepts restricted the Users in a way which was not desirable. The Vendors believed that the Users were fundamentally stupid and without any hope of redemption; and the Gurus believed that the Users were fundamentally stupid but might be saved given the correct tuition.

The early Vendors were peopled by Gurus who had often to undergo the necessary clothing transformation<sup>4</sup> to demonstrate that they had switched camps. These Vendors were responsible for the spreading of the cult to a much wider and naive User population. Some of the Vendors tried to set up their own breakaway cults to avoid the central control imposed by AT&T, the Corporation where the Creators worshipped. These attempts failed. Other Vendors created Sects which specialised in worshipping the many minor Overlords which had appeared about this time. These Sects were partially successful and converted Users to their way of thinking. The Sects created by this method had special names related syntactically to the word UNIX. Only XENIX has survived as an example of this.

Then seeing all this activity, AT&T decided to become a Vendor.

The Gurus were horrified when the Marketing Staff, regarded today as the front line fighting force of all Corporations, were put in charge of the development and promulgation of the cult. The Creators were no longer allowed to directly influence the development, instead they were to pass their ideas to the Marketeers who would decide what was acceptable or not. The Marketeers spent a lot of time producing a new religious tome to help to guide the developers. The book was known as the *Svid* and laid out the central tenets of the practice. The Marketeers were determined that the *Svid* should be elevated to the level of all the other religious books. To this end they tried to make the *Svid* as cryptic as all the others, and succeeded.

After some initial shock, the Vendors accepted the central control which AT&T imposed because they saw that it made their Products accessible to more Users. "Consider it Standard" became the watchword in a Crusade designed to eliminate the undesirable Sects which wished to differ from the *Svid*.

The Users of the Vendors' Products were certainly pleased with AT&T's decision to become a Vendor. It meant that they were no longer tied to the Products of one particular Vendor but could pick and choose without altering their worship. The Gurus were less pleased until they realised that AT&T were unable to change the cult to eliminate them. Every Overlord where the UNIX cult was practiced still

---

4. Often referred to as mv tie neck. This is inexplicable at present.



required a Guru to perform essential tasks.

### *The Sects*

The UNIX cult was always noted for its propensity to split into separate Sects. In the early days, the Creators had a release policy which made sure that all cult members possessed all the relevant facts in order to fully comprehend the implications of the worship. It was said that Users aspiring to be Gurus needed all the facts because the religious books were written for Gurus, and Users could make no sense of them. Unfortunately, access to the information was much abused because the Gurus immediately used the knowledge to alter things and many minor and major Sects of the Cult sprang into life. The ability of each site to generate its own Sect was somewhat curtailed by the cunning ploy of re-issuing the rituals and practices in a slightly altered form from time to time. The Gurus were soon tired of altering the same things every time a new set of rituals came through the door and they began to leave things alone. Also, by this time, the Vendors had made an appearance and because they believed in the Svid Crusade, they stuck with the orthodox mainstream AT&T view of the cult.

Even so, at the end of the period being researched there seems to have been two Sects with differing practices and rituals. The main rival to the orthodox view was a Guru-lead Sect called the "Berkeley System Devotees". This had sprung into existence in the early days of the cult, taking advantage of the knowledge imparted to them by the Creators. However, the early Berkeley Gurus cleverly distributed their rituals and practices in the much the same way as the Creators and this ensured a wider following. The prominence was noted by a higher power and they were chosen to master the revisions of the Cult which were needed to permit worship on some new Overlords.

These new Overlords had managed to conquer the restrictive memory sizes which plagued many of their forbears. It was said by many that the new Overlords had not got this right but at least they did it. As a result the new Overlords had the potential to allow bigger and more expansive practices. The Berkeley Gurus grasped this opportunity with the objective of creating "The Perfect Ritual"<sup>5</sup>. The Perfect Ritual was defined as one where all the letters of the alphabet were used as a "parameter" specifying a distinct action or phase in the worship. The Berkeley Gurus never managed to create the Perfect Ritual, but they came close.

The Berkeley Gurus distributed their rituals and these became popular because the Marketeers inside AT&T were so busy creating the Svid that they failed to notice that Berkeley practices were slowly being adopted on all the new bigger Overlords. The Berkeley rituals were also liked by Gurus because they were nearer the "Old Religion" laid down by the Creators. In a fit of pique, the AT&T Marketeers decided that they would no longer support the new Overlords and branded the Berkeley Gurus as heretics.

As heretics, the Berkeley Gurus decided to go one step further in altering their Sect. They proposed and executed a fundamental change of direction which was to become a "De-facto Standard". The new Sect was revolutionary because it allowed Overlords to talk to each other, but to do this a whole new litany had to be created. New words entered the vocabulary and new concepts were introduced. For many, worship in the new Sect was slow and unwieldy in comparison to what had

---

5. The rival Sects referred to this as "creeping featurism"; the precise meaning of this obscure phrase is under investigation.

gone before. But the new practices meant the easy ability to interconnect Overlords and this was demanded by the Overlords themselves.

Unfortunately for the participants in the Svid Crusade, many Users actually insisted on being able to use some of the Berkeley rituals. The pressure from the Users was such that we begin to see Vendors announcing their Products as being "with Berkeley enhancements". Finally, the AT&T Marketeers were forced to incorporate some features of the rituals which did not conflict with the teachings in the Svid. At this time, we also begin to see specially created Overlords which could be used to worship in the practices of either Sect simultaneously, this was known as the "Universe Concept".

The Berkeley Gurus were so broken by the gestation of the new Sect that many left, some to worship the sun and some to seek salvation in the noble task of Pixel Creation. It was thought that the new Sect would be the last to sally forth from hallowed halls of Berkeley because the staff were demoralised and without joy. The Svid Crusaders were pleased, "All we have to do is wait and do nothing", they said. Since they weren't noted for doing much anyway, this wasn't difficult.

However, much to the dismay of the Crusaders, many Novices amongst Berkeley group were promoted to Gurus, and these new Gurus worked to consolidate and strengthen the new rituals. The label of "slow and unwieldy" was not to be applied again. Time and motion studies were performed on the rituals for the first time in the recorded history of the UNIX cult and a little more than lip service was paid to the notion of efficiency of worship.

#### *The Standardisers*

One way of defeating the degeneration of the pure UNIX cult into Sects was by the creation of a "Standard". As we have seen, the Svid is one example of this. However, the Svid differed from other Standards because only the AT&T marketeers were in a position to generate a Standard without reference to anyone else. They seemed especially keen that no taint of the Berkeley heresy should appear in their work and so did not consult the Users.

In order that the Svid could qualify as a Standard, the AT&T Marketeers had to promise that it would not alter. They agreed to this because they were determined to ensure that the Svid gained religious significance. This was a good thing for other Vendors who were treating the Svid as a Standard but it is possible that the slavish adherence was detrimental to the fortunes of AT&T in the long run because they were unable to update the rituals and practices to keep up with the demand for change created by the Users.

All the other standards were either created by groups of Vendors working together and ignoring the Users; or by groups of Users working together and ignoring the Vendors. Gurus were rarely involved; they were either too busy and important to sit on committees or just plainly could not see the need for conformity.

The Standards rarely reflected the UNIX practices and rituals which were in use at the time of their creation. All of them seemed to have a speculative element, as if the Standardisers themselves tried to develop or perhaps rationalise the rituals in some way. As a result, the Standards were never standard.

However, the various Standards did give the Users an idea of what was expected of them if they desired to move from one Sect to another. Users who did this often, the so-called *Portables*, learned to use the minimum of ritual and to localise the Sect dependent areas of their worship.

The Portables might have been helped by a Standard for the religious language, C. They were surprised to learn that Standard C was, in fact, a different language from the original and almost no-one had an Overlord which could understand it. The changes were no doubt desirable but came from treating C as a "high-level language" rather than using it in the way which the Creators intended, a method of communicating intimately with the Overlords.

#### *Conclusion*

The archives are still being searched for other interesting material but enough has been found to demonstrate the fervent activity which followed the creation of the UNIX cult. We have found no trace of the Creators and barely a hint of the disciples who followed them inside AT&T. We hope that more research may provide some answers and respectfully ask for more funding.

---

④ Peter Collinson is really the Head of the UNIX Support Group at the University of Kent, Canterbury, UK. He has been involved with UNIX systems since 1976, when in the immortal words of Nigel Martin "UNIX changed me from a Lecturer in Computer Science into a Junior Computer Operator". At Kent, he has been responsible for the writing of Cambridge Ring networking software on the VAX, starting with 32V and continuing in the the Berkeley tradition ever since. He has passed through the stage of thinking that UNIX is everything and is now able to recognise the nice things about other operating systems and the nasty things about UNIX. He still thinks that the word UNIX should be allowed to be a noun.

He has not published much in the computer journals but has had many pages printed in the various UNIX newsletters across the world. Most of these words were reporting the activities of the EUUG of which he was a past Chairman and is now a Committee member. Some of the words were a paper "On the Design of the UNIX Operating System" (*login*, July 1984) which he refers to as "The Typing Paper" and did seem to manage to raise a few laughs across the UNIX world.

He still cannot work out whether he should use `***argv` or `**+argv` or `++*argv`. `getopt()` is for Users.

## The X/OPEN show in Luxembourg

*Theo de Ridder*  
*ridder@honhio.uucp*

On 27th February the X/OPEN group presented a demonstration of portability in one of the buildings of the Commission of the European Communities (CEC) in Luxembourg.

I went there as a delegate of the NLUUG, but also to witness the beginning of a new epoch. It was a one-day trip with mini-airplanes to see a mini-demo with maximal side-effects. Surely, the commitment of eleven major companies (BULL, ERICSSON, ICL, NIXDORF, OLIVETTI, PHILIPS, SIEMENS, AT&T, HEWLETT-PACKARD, UNISYS) towards a single standard is a fundamental breakthrough and establishes a critical mass for a successful UNIX explosion in the coming years.

The presentation of the program was in professional hands. So there was a real presenter, supported by a complete light, sound and video installation. The demo itself, compiling and running the 20/20 spreadsheet package, was done by the software house ACE from Amsterdam.

Having too much experience, I was kind enough not to inspect any machine or any source. Just by looking at the public video-screen I tried to make my conclusions. And it was rather embarrassing to see portability advertised by transferring source-code on 5¼" floppies, to see steps of a makefile without any usage of archive or SCCS facilities, and to see a rather suspicious -DXOPEN flag in any C compilation. Furthermore the idea that portability is just the transfer and compilation of source-code was wrong. The amount of work to get the result running is much more relevant, and there was no word about that issue.

The problem of any standard is that it is based on the past and not on the future. Still missing are important issues like networking, windowing and graphics. However, there is a second edition now of the X/OPEN Portability Guide in a much more practical shape than the previous one. It consists of five volumes covering commands and utilities, system calls and libraries, supplementary definitions, programming languages, and data management. Rather unusual for a UNIX environment is the inclusion of COBOL, ISAM and SQL.

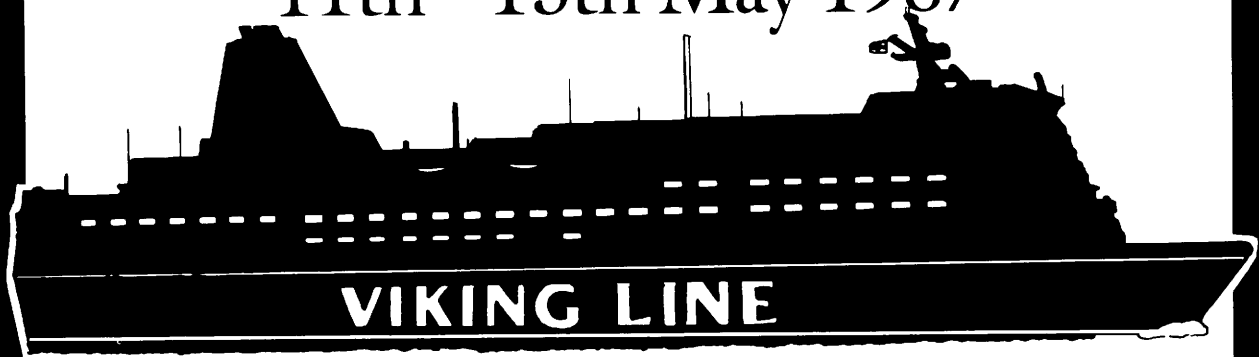
The X/OPEN group gave up its pure European identity, so maybe we are moving towards a single world standard unifying SVID + POSIX + X/OPEN. Whatever happens, there remains a lot of hard work to be done.

# EUUG

European UNIX® systems User Group

## “UNIX® GROWS UP”

A major 5-day event in  
Finland and Sweden  
on board the M/S Mariella  
11th – 15th May 1987



### The Event

The European UNIX systems User Group is to hold one of its most important events yet – in May 1987.

Its **Spring 1987 Conference and Exhibition** will be held over three days (12th – 14th May) on board the Viking Line's luxury passenger ship – the M/S Mariella – with **Tutorial Sessions** at the Hotel Dipoli Conference Centre in Helsinki on 11th and 15th May.

The **Conference and Exhibition** will be staged as the M/S Mariella sails between Helsinki and Stockholm.

### The Programme

Under the broad heading **UNIX® Grows Up**, the Conference will encompass a variety of both new and old developments in the UNIX field and will also look into the future through the eyes of such leading specialists as:

**Bill Joy** – Sun Microsystems Inc

**Rob Pike** – AT&T Bell Laboratories

**Brian Redman** – Bell Communications Research

**Bjarne Stroustrup** – AT&T Bell Laboratories

**Andrew Tanenbaum** – Vrije Universiteit, The Netherlands

---

### Nine different UNIX topics will be presented during the Tutorial Sessions

---

For further details on this major UNIX event, contact:

**The Secretariat: European UNIX® systems User Group, Owles Hall, Buntingford, Herts SG9 9PL, UK. Tel: Royston +44 (0) 763 73039 Facs: Royston +44 (0) 763 73255 Network address: euug@inset.uucp**

®UNIX is a Registered Trade Mark of AT&T in the USA and other Countries

## The CV Macros

*Neil Todd*

*Imperial Software Technology*

Introducing a new *portable* macro package for troff-like processors

Tired of messing around with your CV, trying to get it just right?

Naked troff getting you down?

Need to change it at short notice to fit the Job Ad?

Yes??

Well, in that case you need THE CV MACROS!!

A simple-to-use troff macro package that takes care of all those messy formatting problems and gives you some added features:

- Optional invisible typeface (use `.I`) that utilises the little known feature of most laser printers that allows the selection of a *white* toner cartridge. This specially formulated toner is completely invisible on white paper, so should your manager see it being printed he'll think you've just screwed up again. But get it home, pop it in the oven at *regulo 6* for 15 mins and hey presto! the toner turns black. You never have to be furtive in the print room again.
- A hype typeface (`.HY`) that selects a special toner that absorbs 24 in every 25 photons that strike it - resulting in a typeface that flashes twice a second under normal 50Hz lighting. A slightly increased flash rate is automatically selected when applying for North American jobs (to match the increased tempo of life there).  
*CV Health Warning:* Due to the energy absorbing nature of this typeface, prolonged exposure to bright lights may seriously damage your job prospects.
- A little *white lie* macro (`.WL`) sets the following text in a radioactive typeface that has a half-life (in days) selectable via an argument. As the toner decays the text becomes rapidly unreadable.
- A *photocopy glitch* macro (`.PG`) is available that causes the typeface to be altered by a random dither algorithm applied to the laser beam drive logic. Useful for those embarrassing A level or degree results.
- The *salary* macro (`.SL`) is available to generate suitable *current salary* and *required salary* lines. This is a complicated macro, but illustrates a number of the lesser known troff features, including *.ra* (the random number generator). This macro also takes an optional numerical argument, a weekly auto-increment for the required salary. This saves the interview weary job-hunter from editing the CV merely to ask for more money.

The features of this package are open to one and all. So long as you can use troff or similar you've got it made. To avoid excessive use of these novel features (these typefaces are expensive you know!) troff will only function with the `-mcv` macros if the input file name is `$HOME/cv`.

Oh, and one last thing, remember to de-encrypt it first!

# EUUG NATIONAL GROUPS

**AFUU** (France)  
c/o SUPELEC,  
Plateau du Moulon,  
91190 GIF-SURYVETTE,  
France.

**DKUUG** (Denmark)  
Kabbelejvej 27B,  
DK-2700 BRØNSHØJ,  
Denmark.

**EUUG-S** (Sweden)  
NCR Svenska AB,  
Box 4204,  
17104 SOLNA,  
Sweden.

**FUUG** (Finland)  
OY Penetron ab,  
Box 21,  
02171 ESPOO,  
Finland.

**GUUG** (Germany)  
Mozartstrasse 3,  
D-8000 MUNICH 2,  
Federal Republic of Germany.

**IUUG** (Ireland)  
Glockenspiel Ltd.,  
19 Belvedere Place,  
DUBLIN 3,  
Ireland.

**i2U** (Italy)  
Viale Monza, 347,  
20126, MILANO,  
Italy.

**NLUUG** (Netherlands)  
Xirion bv,  
World Trade Center,  
Strawinskylaan 1135,  
1077 XX,  
AMSTERDAM,  
The Netherlands.

**NUUG** (Norway)  
Unisoft AS,  
Enebakkvn 154,  
N-0680 OSLO 6,  
Norway.

**UKUUG** (United Kingdom)  
Owles Hall,  
Buntingford,  
Herts. SG9 9PL,  
United Kingdom.

**UNIGS** (Switzerland)  
c/o Institut für Informatik,  
ETH Zentrum,  
8092 ZÜRICH,  
Switzerland.

**UUGA** (Austria)  
TU Wien,  
Inst für Praktische Informatik,  
Gusshausstr 30/180,  
A-1040 WIEN,  
Austria.



The Secretariat: European UNIX® systems User Group, Owles Hall, Buntingford,  
Herts SG9 9PL, UK. Tel: Royston +44 (0) 763 73039 Facs: Royston +44 (0) 763 73255  
Network address: euug@inset.uucp

NEWS FROM THE NATIONAL GROUPS

News from the National UNIX User Groups

*Contributors:*

Philip Peake	AFUU	France
Keld Jørn Simonsen	DKUUG	Denmark
Frances Brazier	NLUUG	The Netherlands
Daniel Karrenberg	GUUG	Germany
Sunil K. Das	UKUUG	U.K.

*Editor's Note*

This section is planned as a regular feature, and is expected to be longer in future editions of the Newsletter.

Contributions will be very welcome, and can be sent by email to

*euug-n@mcvax*



# EUUG

European UNIX® systems User Group

## AUTUMN MEETING

UNIVERSITY OF DUBLIN

Trinity College

21st - 25th  
September  
1987



- Software Engineering
- User Interfaces
- Graphics

For further details contact:

The Secretariat: European UNIX® systems User Group,  
Owles Hall, Buntingford, Herts SG9 9PL, UK.

Tel: Royston +44 (0) 763 73039 Facs: Royston +44 (0) 763 73255  
Network address: euug@inset.uucp

®UNIX is a Registered Trade Mark of AT&T in the USA and other countries.

## AFUU Diary

*Philip Peake*  
*philip@axis.uucp*

### *Axis Digital*

#### *Introduction*

Since the last report from the AFUU was written, many things have happened. This article is to keep you up to date on events in France.

#### *The AGM and technical conference*

Due to the numbers of people interested in attending, this was held not at SUPELEC (as it was in 1985) but at La Defence, which, for those of you who don't know Paris, is just to the west of Paris, and does a very good imitation of Dallas with its huge glass sky-scrapers. In fact, some people claim that it is more impressive than Dallas. Anyway, our meeting was held in one of these huge glass things. The only real disadvantage was that the conference theatre was at level -1, and some of the tutorials and exhibitions were at level 33. That minor inconvenience apart, all passed smoothly.

The only part of the event restricted to AFUU members was the AGM, at which a new governing committee was elected. Otherwise, the conference/exhibition was open to members of the public. They evidently liked what they saw, since we have 58 new members as a direct result of this meeting. If the membership follows the pattern of previous years, we should just manage to reach a membership of 400 by the end of this year.

#### *UNIX SYSTEMS '87 and AFUU Technical Conference*

During the second week of May (yes, just before the EUUG conference ...) The AFUU are organising another technical conference, which will run in parallel with the UNIX SYSTEMS 87 exhibition. This exhibition is the most important UNIX exhibition in France, and attracts more and more visitors every year. The provisional conference timetable is shown over the page.

This year, not all of the talks will be in French, although that will be the official language of the event. Anyone wanting information on this event, or a more complete timetable, can contact Anne Garnery (the AFUU secretary), either by telephone, or by email:

AFUU  
c/o SUPELEC  
Plateau du Moulon  
Gif sur Yvette  
91190  
FRANCE

Tel: +33 (1) 60 19 10 13  
uucp: anne@afuu.uucp

	5th May Standards	6th MAY Work Stations	7th May Networking
9h15	Users' market	Performance evaluation	streams
	Constructors' market	OS for workstations	C/Pascal
10h45	X/OPEN part 2	Unix Servers	Communications with OS
	C Standards	Project EMERALD	NCS
12h30	LUNCH	LUNCH	LUNCH
14h00	Europeanised Applications	PC/RT (6150)	Management
	International XENIX (386)	NEWS software	RTU
15h45	SCO - 386 XENIX	X Windows	Fault tolerant systems
	Standard Interfaces	?	?
17h47	FINISH	FINISH	FINISH

## News from the Danish Group

*Keld Jørn Simonsen  
keld@diku.uucp*

### *DKUUG*

Yes, we have got our own machine now! The Ericsson Information System A/S firm just happened to have a Sun 2 work station with a handsome 130 Mb disk on it, that it wanted to provide to DKUUG. How come? Well, the DKUUG board spread the word that we thought we needed a machine of that size for our network, and that we accepted gifts, and some of our fellow members are really nice to us. We also earlier got modems and software in this manner...

So that machine should nicely fit our need for a backbone machine, offloading our yo-yo VAX 785 (diku). Well it is not so much the VAX that has given us troubles the latest 2 months, but our cooling system. So we have been running the diku machine for open windows for some time. Then both our x.25 PAD lines were out of work at the same time for about a week, so the Danish network has not been that amused lately. Let's see what happens when we get the new machine in production.

The new machine will also serve as an archive site of news and the EUUG public domain tapes. We have run an experimental automatic archive service based on mail for quite some time now, and it has been almost problem free. Well, it seems that the demand is not that big either, about 1 Mb of files transferred a month from in all 15 sites out of the 35 Danish EUnet sites. Compare this to our total network load of about 70 Mb a month at diku.

Our machine was just in time so we could exhibit it on the DKUUG stand on the big MicroData 87 exhibition here in Copenhagen. And it generated quite some interest together with those ordinary looking people on the stand. "WOT! No tie?!" was one of the most frequent remarks heard. Well, the stand people consisted exclusively of the Danish board members and hackers from the nearby university, as all our commercial supporters were engaged at the other stands, so we did look a bit strange. It seemed that we did generate a lot of good contacts, with both our members and also new contacts, so look out for DKUUG to soon become the biggest group in EUUG!

The Danish group has become a member of /usr/group international, and we are considering affiliating /usr/group. This should not affect our engagement within EUUG.

We usually have 4 meetings a year and the schedule looks like this:

<i>Date</i>	<i>Location</i>	<i>Subject</i>
early April	IBM Headquarters	A meeting about UNIX on big IBM machines, and 4th generation languages
21 May	Copenhagen	UNIX tutorials
10 September	Aarhus	
26 November	Copenhagen	General assembly and yearly meeting

## News from the Netherlands

*Frances Brazier*

*Secretary of the NLUUG*

### *Conferences*

The NLUUG organises two conferences a year: one more technically oriented conference and a larger conference with both industrial sessions and technical sessions, and an exposition. Our last conference in November was of the last category: its title "UNIX > OS?". A variety of topics was covered in the technical sessions, ranging from networking to benchmarking. The exact titles of the talks given:

1. Benchmarking.
2. TARDIS - a system for interprocess communication.
3. How has Minihouse achieved machine independence?
4. CADESE - computer aided distribution environment for software engineering.
5. Connection between ECONET and BSD networking.
6. MANIP - a modelling and simulation system for satellites.

As a change, tutorial-like sessions were organised instead of the more commercially tinted talks. Representatives from a number of commercial training centres were asked to provide an overview of specific aspects of UNIX. The program included the following titles:

1. The popularity of UNIX.
2. Aspects to be considered when choosing a training centre.
3. The development of administrative applications with UNIX.
4. Is UNIX real-time (or can it be made real-time)?.
5. UNIX for many - more than 500 - users.

### *Services offered to all members*

The NLUUG has its own office which can be approached by its members and interested parties during office hours. Most questions can be answered immediately, those which can not are answered within a reasonable period of time

The NLUUG has provided its members with copies of the Manchester proceedings and with a selection of the papers presented in Florence and Copenhagen. We are planning on continuing this service as long as it's financially feasible.

### *Activities for our industrial members*

Due to a linear growth in the number of industrial members and stability in the number of academic members, the proportion of commercially oriented members of the NLUUG has grown considerably in the last few years. As the NLUUG is trying to provide a platform for all professional UNIX users this has meant a slight change in policy. Upon the request of a number of our industrial members and on their behalf an external Public Relations Advisor has been contacted and presented with the task of finding out what can be done to stimulate the acceptance of UNIX in general. A number of suggestions had been proposed by the industrial members involved: the elaboration of these ideas has been left to the professional advisor. Having defined the different target groups involved and the types of actions which would be appropriate, the NLUUG is now awaiting a report in which an overview of

the financial consequences involved will be specified. Within the very near future this report will be sent to our industrial members along with the price tag and the discussion on its feasibility will commence.

*Other activities, not really initiated by the NLUUG*

A number of industrial members have employed the same Public Relations Bureau to define the UNIX market in the Netherlands. The results of this report will be made known to the NLUUG, all companies interested will be given the opportunity to order the report at a later date.

Letter from Germany  
or  
GUUG is Alive and Well

*Daniel Karrenberg*  
*dfk@cw.nl*

*GUUG*

Since the Autumn '86 GUUG conference in Karlsruhe, we have received a lot of new confirmations of membership. At this moment (March '87) GUUG has 420 members. GUUG maintains an office that can be reached between 8am and 1pm each working day at +49 89 5380503. During the remaining time callers can leave a message on tape. An increased number of phone calls handled by Frau Bester, the GUUG secretary, reflects the growing local interest in UNIX. Most callers want information about GUUG and EUUG, especially date and time of the next meetings. This information can easily be given based on the information material we have at hand.

The more difficult and still quite frequent questions are about application software for UNIX. Thus we decided in Autumn '86 to prepare a software catalogue which will be available April/May '87. The catalogue is open for hard- and software manufacturers that supply and maintain products for the German market. It should reach as many German UNIX users as possible and the first edition will contain some 150 entries. GUUG members will receive a copy of the catalogue automatically, additional copies and copies for non-GUUG members can be purchased at DM 27,- per copy. The catalogue will be regularly amended. New entries, changes and cancellations will be continuously registered at the GUUG office and published in subsequent editions.

The biggest problem of GUUG is at present - sad to say - EUUG services. Mainly the regular appearance of the EUUG newsletter was sadly missed by many members! Especially so, since GUUG had to increase its membership fees substantially in order to pay the increased EUUG dues. This has been a matter of high controversy amongst GUUG members.

GUUG-Nachrichten still appear four times a year, and serious efforts are being made to space the copy dates more evenly around the year. The German part of EUnet is also alive and well with about 100 sites. Unido, our backbone, maintains gateways to the German part of EARN (BITNET) as well as the German national research network DFN.

GUUG will hold its Spring meeting on April 23rd and 24th at the Technische Universitaet Muenchen. At Spring meetings the various special interest groups meet and report to each other about their work. There also will be some general interest talks about User Interfaces, UNIX and PCs (such as the ATAR STs), Databases and - yes you guessed it - standards. Watch out for a report in the next "Letter from Germany". Visitors from other national groups are of course welcome and should ring the GUUG office for further information.

In September GUUG will hold a conference and exhibition in Karlsruhe as in 1986. At the time of this writing the exact date is not yet clear, because there are problems with the venue which could lead to an unwanted collision with the Dublin EUUG meeting. There are also plans under discussion to hold the Autumn '88 GUUG



conference in conjunction with the Autumn EUUG conference somewhere in northern Germany.

## Forthcoming UKUUG Events

*Sunil K. Das*

*Chairman — UKUUG*

### *1. Forthcoming UKUUG Events*

#### *UKUUG Summer Technical Meeting*

Tuesday 7th and Wednesday 8th July, 1987 at Newcastle University  
Keynote speaker: Michael Lesk

#### *UKUUG Winter Technical Meeting*

Monday 14th and Tuesday 15th December, 1987 at City University, London

#### *EUUG 1988 Conference*

Tuesday 7th thro' Thursday 9th June, 1988 (Tutorials: Monday 6th and Friday 10th June) at Alexandra Palace, London

Run in parallel with /usr/group/UK and EMAP Ltd UNIX Exhibition and commercial / industry sessions.

### *2. Winter 1986 Conference*

Two papers from the Winter 1986 UKUUG conference follow:

GKS in C++

John E. Richards

An NRS Processor in C and the Future

Piete Brooks

Full copies of the conference proceedings can be obtained from:

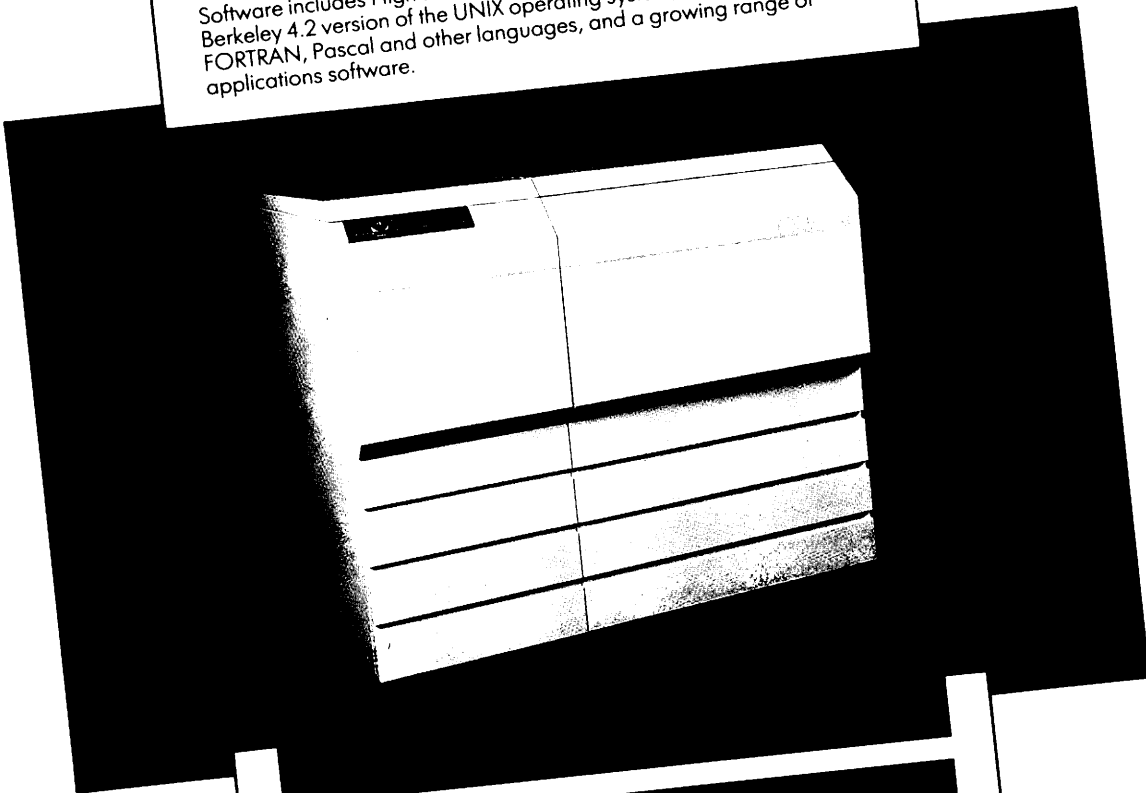
The Secretariat,  
The UKUUG,  
Owles Hall,  
Buntingford,  
Herts.  
SG9 9PL  
U.K.

# ORION 1/05

The ORION 1/05 processor is an advanced high-performance 32-bit computer architecture which draws on mainframe/supercomputer techniques to provide outstanding performance at reasonable cost. The CPU includes an IEEE 754 standard floating-point unit and two combination cache-plus-memory management units, one for data and one for instructions. It makes extensive use of concurrency, including a sophisticated pipeline and separate buses for instructions and data.

The processor is complemented by a range of autonomous microprocessor-based I/O subsystems that connect to the 32 Mbyte/sec ORION synchronous backplane. A variety of peripherals can be attached, including disks from 168-474 Mbytes, 1/4 inch cartridge and 1/2 inch reel-to-reel tape drives, and the ORION LaserWriter. Communications capabilities include support for Ethernet, RS232-C and X.25.

Software includes High Level Hardware's implementation of the Berkeley 4.2 version of the UNIX operating system, compilers for C, FORTRAN, Pascal and other languages, and a growing range of applications software.



## HIGH LEVEL HARDWARE

**High Level Hardware Limited**

PO Box 170, Windmill Road, Headington, Oxford OX3 7BN, England  
Telephone: (0865) 750494 Telex: 838854 - HLH G

## GKS in C++

*John E. Richards  
Richards@uk.ac.bristol.qgb*

*Praxis Systems Plc*

*Aspects of a binding of GKS, the ISO standard for 2D graphics, to the C++ programming language are presented. The binding makes use of classes and derived classes to define GKS concepts such as segments and workstations. Operator overloading is used for some GKS functions.*

### *Introduction*

GKS [1] is defined in a language-independent manner. Bindings of GKS to programming languages have to be defined in order that different implementations of GKS in one language present the same interface to the programmer. Work on bindings has been done on several languages [2], in particular, Fortran [3], Pascal [4,5,6], C [7], Algol 68 [8] and Ada [9,10]. Some work has also been done on bindings for non-procedural languages such as Prolog [11,12] and Smalltalk-80 [13].

This paper presents some ideas for a binding in a new programming language, C++ [14]. Although C++ is a descendent of the C programming language [15], it has some novel features which result in a binding considerably different to the proposed C binding [16].

Bindings can be categorised into two types: the radical and the reactionary. The radical binding attempts to take full advantage of the language, using the argument that people choose to program in a language because they find its facilities useful. The reactionary view is that bindings should be as similar to each other as possible, which means in practice that they should all resemble the lowest common denominator, the Fortran binding. The latter approach has the advantage that programmers can move from one language to another without learning a completely new binding. The aspects of the binding presented in this paper tend towards the radical view. As such, it is intended that they should provoke discussion.

There can be no standard language binding unless there is a standard for the language. C is caught in this predicament; until the current standards work is complete the C binding can only have the status of a technical report. A C++ binding would be in a worse situation. C++ is an evolving language; there is no standard, or work in progress on a standard, and the creator of the language has stated [14] that it will change to reflect any changes to C. The main reason for investigating a C++ binding is to examine what novel aspects are introduced by the use of the language. This paper shows that C++ provides some interesting facilities which can be used in a natural way in a graphics programming environment.

### *C++*

The C++ programming language is a superset of the C programming language, with a few minor exceptions. The major enhancement it contains is the concept of *classes*, which provide the programmer with a method of defining new data types. It is possible for the programmer to:-

- specify how a new type can be accessed,

- specify the operations that can be performed on it.
- define operations to be performed when an object of the new type is created (by a *constructor* function).
- define operations to be performed when an object of the new type is destroyed (by a *destructor* function).
- provide user-defined conversions from one type to another.
- define a new type in terms of a previous user-defined type.
- manipulate lists of objects of similar types, with dynamic selection of the appropriate functions.

An example will serve to illustrate a number of these points. The following is a definition of a new type, or **class**, called **point**.

```
class point {
    float  x, y;
public:
    point (float, float);    // constructor
    void add (point);
}
```

A **point** consists of two **float** variables, *x* and *y*. As *x* and *y* are defined above the keyword **public**, they are considered to be *private members* of the class and they cannot be accessed except by means of any functions (the *member functions*), that belong to the class. In this case, there are just two member functions, the constructor function, which takes two **float** values as arguments, and the function called *add* which takes a **point** as an argument. When a new **point** object is defined, the constructor function will be called.

The constructor function can be defined as follows:

```
point::point (float xx, float yy) {
    x = xx;
    y = yy;
};
```

A new **point** object is created by a definition:

```
point centre(0.1, 0.7);
```

The *x* and *y* members of the *centre* **point** will be set to 0.1 and 0.7 respectively. A compilation error would occur if no initial value was specified.

The *add* function is used to add one point to another, and is written as follows:

```
void point::add (point p) {
    x += p.x;
    y += p.y;
}
```

This adds the components of the point specified as the argument to the components of the point which is used to invoke the function. The following code shows how it could be used:

```
point centre(0.1, 0.7);
point offset(0.2, 0.1);
centre.add(offset);
```

After executing this code, *centre.x* would equal 0.3 and *centre.y* would equal 0.8.

C++ also permits overloading of function names. Two different classes can have member functions of the same name, or two functions can be defined with the same name, but take different types or number of arguments. The compiler is responsible for resolving any conflicts. For example, two functions called *print* could be defined: one to take an *int* and one to take a *char\** argument. The following sort of code is then possible:

```
print(2);
print("The quick brown fox");
```

Also, operators can be overloaded. This permits the definition of suitable operations for new user-defined types. For example, the addition and subtraction operators could be defined for the class *point*, making it possible to write code like:

```
point p1(0,0), p2(-4,8), p3(2,9);
p1 = p2 + p3;
```

#### *Mapping of GKS Data Types*

The data types integer, real and string are mapped to the appropriate C++ types, *int*, *float* and *char[]*. These types are given the names *Gint*, *Gfloat* and *Gchar[]*, to correspond to the names used in the proposed C binding.

The GKS enumeration types are mapped to C++ *enum* types.

The GKS point type is mapped to a C++ class called *Gpoint*:

```
class Gpoint {
    Gfloat x;
    Gfloat y;
public:
    Gpoint() {}
    ...
};
```

The GKS name types are mapped to the names of objects. This is explained in the following sections.

#### *Objects*

A major advantage of C++ is the ability to work in an object-oriented way. There are several suitable candidates for objects in GKS. This binding uses concepts such as workstation, segment, polyline, polymarker index, window, and GKS itself, as classes. The GKS functions are then mapped to C++ functions that operate on objects of these classes.

There are several advantages to this approach:

- there is a well-defined set of operations on each object,
- it is possible to perform a set of operations on a list of heterogeneous objects by using derived classes,
- there is a reduction in the number of distinct subroutine names,
- the programmer is given the opportunity of defining new classes derived from the ones defined in the binding.

#### *The GKS Class and Object*

The binding contains the definition of a class called *GKS*. This class contains the items of the GKS state list, together with a set of appropriate GKS functions.

For example, the CLOSE GKS function is a member of the *GKS* class and has the name *close*. The OPEN GKS function is mapped to the constructor function for the

class. Opening GKS is achieved by defining an object of the **GKS** class. The following code opens GKS by creating an object called *g* and then immediately closes it.

```
GKS g(cerr, bufsize);
g.close();
```

The parameters of the OPEN GKS function are passed to the constructor function, which can be designed to detect an attempt to open GKS when it is already open. It is impossible to close GKS before it has been opened, because the *close* function cannot be called unless there is a **GKS** object in existence.

#### *Workstations*

The **Workstation** base class is defined as a workstation state list plus the set of workstation control functions. Other classes are derived from the base class for each different type of workstation. It is possible to describe the relationship between similar workstations. For example, a class **Tek410x** could be defined as a derived class of **Workstation** to represent the Tektronix 410x series of graphics terminals. Then, classes **Tek4105**, **Tek4107** and **Tek4109** could be defined as derived classes of **Tek410x**. This provides a natural way to construct workstation drivers, where there is often a large amount of code shared between similar devices. Only functions that differ in a family of workstations need to be specified; the remainder are inherited from the parent class.

The OPEN WORKSTATION function is performed by the constructor function in a similar way to OPEN GKS. The following code opens, activates, deactivates and closes a workstation:

```
Tek4107 display(conid, "Tektronix 4107");
display.activate();
display.deactivate();
display.close();
```

The first line creates an object called *display* of the class **Tek4107**. This is the OPEN WORKSTATION call. Subsequently, all workstation functions are invoked by calling member functions of *display*. No integer workstation identifier is passed as an argument to these functions because it is not needed; the workstation is identified by the object name. The choice of name for the object is entirely the responsibility of the programmer.

#### *Output Primitives*

A class is defined for each type of output primitive. There is a base class for the output primitives called **OutputPrimitive**. This has four derived classes, **Text**, **CellArray**, **GDP** and **GpointArray**. The latter is used as a base class for **Polyline**, **Polymarker** and **FillArea**. **GpointArray** is defined as follows:

```
class GpointArray {
    Gpoint* point;
    Gint    number;
public:
    ...
};
```

An object of an output primitive class is created by a constructor function when the object is defined. The constructor function takes arguments which are used to allocate sufficient space for the object and to initialise it. For example, an object of the **FillArea** class, called *triangle*, would be allocated with sufficient space to hold 3 points, by:

```
FillArea triangle(3);
```

Having allocated space, values have to be assigned to the triangle. The most natural way of doing this is to make the triangle appear to be an array. This can be done in C++ by redefining the array subscripting operator, [], for the `GpointArray` class. Values are assigned as follows:

```
triangle[0] = Gpoint(0.1,0.1);
triangle[1] = Gpoint(0.9,0.3);
triangle[2] = Gpoint(3.0,2.1);
```

The most important member function for the output primitive classes is the one that actually draws the object. The obvious approach is to define a function called *draw* for each output primitive. To draw the triangle it would be necessary to say:

```
triangle.draw();
```

As this is the commonest function performed on output primitives, it was decided instead to redefine the function call operator, (), to perform the draw function. Thus, to draw the triangle:

```
triangle();
```

This notation leads to a concise form of programming. There is no need to specify the number of points or an array of points in the function call as that information is already contained in the object. The following code shows how to use `Text` and `Polyline` objects.

```
Polyline line(100); // allocate a Polyline
                  // and initialise it
for (int i = 0; i < 100; i++)
    line[i] = Gpoint(i*0.2,sin(i));
// Now allocate and initialise a Text object
Text title(Gpoint(9,-1),"Sine Curve");
// Draw the line and title
line();
title();
```

Attributes Each GKS attribute is defined as a class. An attribute class will typically consist of one or more variables, together with a SET and INQUIRY function for that attribute. For example, the MARKER TYPE attribute class is defined as follows:

```
class Gmkty : public GintAttr {
    Gint v;
public:
    Gerror set(Gint);
    Gerror inq(Gint&);
};
```

Where attributes share common types and member functions, base classes are defined. In the above example, `Gmkty` is defined to be a derived class of `GintAttr`, which is a base class for integer attributes. Similarly there is a `GfloatAttr` base class for real attributes. As is shown below, this enables the programmer to manipulate lists of attributes in a straightforward way.

Attributes that belong to the GKS state list are specified as members of the GKS object. If a GKS object called *g* has been created, MARKER TYPE and POLYLINE INDEX are referred to as *g.markertype* and *g.lineind* respectively, as in the following extract of source code:



```
g.markertype.set(2);
g.markertype.inq(i);
g.lineind.set(i);
```

Attributes stored in the workstation state list are referred to by using the name of the appropriate **Workstation** object. Some attributes are stored as arrays. For example, POLYLINE BUNDLE number 3 on the *display* workstation is referred to as *display.linerep[3]*. This means that a call such as SET POLYLINE REPRESENTATION, which would be the following in the C binding:

```
gsetlinerep (display, 3, bundle);
```

is considerably different in form in the C++ binding:

```
display.linerep[3].set (bundle);
```

### Segments

Segments are difficult to handle in the C++ binding as there is a requirement for a segment to have both an internal and an external name. The natural way to define a segment is as an object of a class, **Segment**. The CREATE SEGMENT function can be bound to the **Segment** class constructor function. The following code will create a new segment, called *s*, by calling the constructor function.

```
Segment s; // create a new segment
```

Unfortunately, an external name is needed to permit the storage and retrieval of segments in metafiles, so the approach taken in Rosenthal and Ten Hagen's C binding [7] has been adopted and a segment is allowed to have both an internal and an external name. The external name is only used when creating the segment.

```
Segment s(3); // internal name: s, external name: 3
s.close(); // close it
s.del(); // and delete it
```

Segment attributes are handled similarly to other attributes. For example, SET SEGMENT VISIBILITY is coded as:

```
s.vis.set (GVISIBLE);
```

Each segment attribute also has an inquiry function called *inq*. INQUIRE DETECTABILITY is written as follows:

```
s.det.inq (detect);
```

### Overloading of Function Names

Wherever possible, the minimum number of distinct function names is used. For example, GKS, workstations and segments are all closed by a *close* member function. Attributes are always changed by a *set* function and their values interrogated by an *inq* function. The result is a big reduction in the number of function names at the expense of an increase in the number of different data types.

### Overloading of Operators

Some functions have been bound by redefining operators. Examples have already been seen above in the treatment of output primitives. Three other examples are COPY SEGMENT TO WORKSTATION, ASSOCIATE SEGMENT WITH WORKSTATION, and DELETE SEGMENT FROM WORKSTATION.

Copying a segment to a workstation is bound to the << operator. This operator, which is the shift left operator by default, is used by C++ as an output operator and so seems appropriate for this function. If *display* is an object of class **Workstation**, and *seg1* and *seg2* are objects of class **Segment**, then the following

will copy *seg1* and *seg2* to *display*.

```
display << seg1;
display << seg2;
```

However, it is possible to concatenate these operations by defining the operator in a suitable way. The following code is equivalent to the code above:

```
display << seg1 << seg2;
```

This is considerably terser than the equivalent C binding code:

```
gcopysegws(display, seg1);
gcopysegws(display, seg2);
```

The notational convenience increases with the number of segments that are copied to the workstation in the one statement.

The += operator is redefined for the ASSOCIATE SEGMENT WITH WORKSTATION function, and the -= operator for the DELETE SEGMENT FROM WORKSTATION function.

```
display += seg1 += seg2 += seg3;
display -= seg2;
```

The += and -= operators need to be defined very carefully. C++ does not provide a way of altering the order of evaluation when an operator is redefined. So,

```
display += seg1 += seg2 += seg3;
```

is evaluated as:

```
display += (seg1 += (seg2 += seg3));
```

Parentheses can be used to enforce the desired result, but this is tedious:

```
((display += seg1) += seg2) += seg3;
```

The solution adopted was to redefine the += operator, when both the lvalue and rvalue are segments, to create a "last-in, first-out" stack and place the segments on it. The operator returns a pointer to the stack. If the += operator is passed a segment as an lvalue and a pointer to a stack as a rvalue, it adds the segment to the stack and returns a new pointer. When the last += or -= in the expression is executed, the segments are popped off the stack and the appropriate GKS functions invoked. This ensures that the functions are invoked in the correct order and removes the need for parentheses.

The -= operator is handled in an similar way.

Operator overloading is also used for the metafile functions.

### *Metafiles*

Metafiles are defined as derived classes of type **Workstation**. One class is required for input metafiles, **MetaIn**, and one for output metafiles, **MetaOut**. This is because input and output metafiles have different sets of member functions. Metafile items are defined as objects of type **MetaItem**.

Opening a metafile is achieved by defining an object of the appropriate class, as for other OPEN WORKSTATION calls, e.g.

```
MetaOut gksm_out (cout, "GKSM output");
```

The **MetaItem** class contains fields for the item type, item length, and a data record. Once the programmer has placed the appropriate information in the item,

the WRITE ITEM TO GKSM function can be used. The function is mapped to the << operator, using the same paradigm as is used for streams output in C++, e.g.

```
MetaItem item;
    // place values in the item
item.type = HEADER;
item.length = 80;
item.data = "This is a title";
    // write it to the metafile
gksm_out << item;
```

The GET ITEM FROM GKSM function is mapped to a *get* member function of the *MetaIn* class. READ ITEM FROM GKSM is mapped to the >> operator, by analogy with WRITE ITEM TO GKSM. As the *MetaItem* class contains the maximum item data record length as one of its members, one of the arguments to READ ITEM FROM GKSM can be eliminated.

The INTERPRET ITEM function is mapped to the overloaded function call operator, (), for objects of class *MetaItem*. For an example of using metafiles, see Section 14.1.

#### *Heterogeneous Lists*

A major advantage of the C++ binding is the ability to use heterogeneous lists. A list can be defined to hold pointers to objects of a base class. The same list can then be used to hold pointers to objects of any derived class. Operations can be applied to objects on the list in a uniform way without the need to know the type of the object that is being manipulated. The meaning of the operation that is being applied depends on the actual type of the object and this is only known at run-time.

For example, it is simple to define a list that can hold integer attributes:

```
struct aulist {
    aulist* next;
    GintAttr* at;
};
```

Any object of the *GintAttr* class, or any of its derived classes, can be placed on such a list. Then an operation can be performed on each object on the list, e.g. to set each integer attribute to the value 3:

```
// p is a pointer to an aulist
while (p != 0) {
    p->at->set(3);
    p = p->next;
}
```

A more interesting and useful example is a list of output primitives. A similar structure to the integer attribute list can be defined by:

```
struct plist {
    plist* next;
    OutputPrimitive* at;
};
```

Objects of any of the derived classes of *OutputPrimitive* (e.g. *Polyline*, *FillArea*, *Text*) can be stored on the list and then drawn by stepping through the list.

```

// p is a pointer to a plist
while (p != 0) {
    (*p->op)(); // draw it
    p = p->next;
}

```

In effect, the above piece of code implements a simple graphics display list, or segment store.

### Examples

#### Example 1

Overloading operators can make some programs much shorter. The following C++ code reads and interprets items from a metafile:

```

// declarations omitted
MetaIn gksm_in(cin, "GKSM input");
while (gksm_in.get(item) != EOF) {
    gksm_in >> item;
    item();
}

```

This is considerably shorter than the equivalent code using the C binding:

```

gopenws(gksm_in, cin, "GKSM input");
ggetgksm(gksm_in, result);
while (result->type != EOF) {
    greadgksm(gksm_in, length, record);
    ginterpret(result, record);
    ggetgksm(gksm_in, result);
}

```

#### Example 2

The ability to derive classes from other classes makes it possible for the programmer to define new output objects. In this simplified example a new type, called **Block**, is defined to represent filled upright rectangles. The only operations permitted on a **Block** are to define a new one, to move it, and to draw it. The class is defined as follows:

```

class Block : public FillArea {
public:
    Block(Gfloat, Gfloat); // constructor
    void move(Gpoint);    // move it
    void operator() ();   // draw it
};

```

The **Block** class is a derived class of **FillArea**. No data items are declared as they are all inherited from the **FillArea** class. The constructor function takes two arguments, the width and height of the block, and initialises the data.

```

Block::Block(Gfloat w, Gfloat h) : (4)
{
    (*this)[0] = Gpoint(0,0);
    (*this)[1] = Gpoint(w,0);
    (*this)[2] = Gpoint(w,h);
    (*this)[3] = Gpoint(0,h);
}

```

The (4) term on the first line of the constructor function is automatically passed to the `FillArea` constructor function and allocates space for an object with four vertices. The *this* pointer is always available in a member function and points to the object for which the function was invoked.

The *move* function changes all the coordinates by a displacement.

```

void Block::move (Gpoint p)
{
    for (int i=0; i<4; i++)
        (*this)[i] += p;
}

```

The *draw* function, which, in common with other output primitives, is performed by the overloaded function call operator, first sets the interior style to solid and then invokes the function to draw a `FillArea` object.

```

void Block::operator() ()
{
    g.fillintstyle.set(GSOLID);
    FillArea::operator() ();
}

```

Once the code has been written to define the new class, it can be used in the following way.

```

Block box(2.5,3.5);
box.move(Gpoint(1.0,1.5));
box();    // draw the box

```

Objects of the `Block` class can be treated identically to objects of the `FillArea` class.

```

// box can be put on a plist
p->op = &box;
// box will be drawn when
// stepping through list.
while (p != 0) {
    (*p->op)();
    p = p->next;
};

```

#### *Conclusions and Future Work*

The production of a binding of GKS in C++ has shown the suitability of the language for writing graphics applications programs. GKS concepts can be translated into classes in C++ in such a manner that the programmer is encouraged to think in an object-oriented way. The use of function and operator overloading results in shorter and easier to write programs. The ability to use heterogeneous lists of objects and the possibility of deriving new types from the basic GKS ones are notable benefits.

The number of distinct GKS function names is reduced to only 31 in the C++ binding, compared to over 210 in the C binding. This is achieved by using overloaded function names, and is largely due to the use of *set* and *inq* member functions for each attribute. However, the number of data types is increased from about 120 in the C binding to over 160.

Slater [5] says that there are several ways of producing a Fortran binding but all reasonable bindings would be structurally similar; whereas for Pascal a number of structurally dissimilar bindings can be devised. In C++, the number of possible bindings is probably greater still. In many ways the design of a binding is a matter of personal taste, and the one presented here may well change drastically to accommodate other people's ideas.

The next step is to test the binding on top of a GKS library [17,18] which is written in Fortran 77 and uses a Fortran binding. This should give indications of the usability of the binding and some ideas for improvements.

#### *Acknowledgements*

Dr. Mark Rafter of Warwick University for suggesting the method of handling the ASSOCIATE SEGMENT WITH WORKSTATION and DELETE SEGMENT FROM WORKSTATION operators.

#### *References*

- [1] ISO  
Graphical Kernel System (GKS) - Functional Description  
ISO IS 7942 July 1985
- [2] M. Sparks  
Graphics Language Bindings - the What and Why  
Computer Graphics Forum Volume 4 Number 4 Pages 387-392 December 1985
- [3] ISO  
Graphical Kernel System (GKS) Language Bindings - Part 1: FORTRAN  
ISO DP 8651/1 December 1984
- [4] S. Antoy and G. Dettori  
Towards GKS Binding to Pascal  
Proc. Eurographics 83 Zagreb, Yugoslavia September 1983 Pages 211-214
- [5] M. Slater  
GKS in Pascal  
Computer Graphics Forum Volume 3 Number 4 Pages 259-267 December 1984
- [6] ISO  
Graphical Kernel System (GKS) Language Bindings - Part 2: Pascal ISO DP  
8651/2 1985
- [7] D. S. H. Rosenthal and P. ten Hagen  
GKS in C  
Proc. Eurographics 82 North-Holland Manchester 1982 Pages 359-369
- [8] R. R. Martin and C. Anderson  
A Proposal for an ALGOL 68 Binding of GKS  
Computer Graphics Forum Volume 4 Number 1 Pages 43-57 January 1985
- [9] M. Mac an Airchinnigh  
The Specification and Implementation of GKS Application Software in ADA  
Computer Graphics Forum Volume 3 Number 2 Pages 153-167 June 1984

- [10] ISO  
Graphical Kernel System (GKS) Language Bindings - Part 3: Ada ISO DP 8651/3 1985
- [11] P. Sykes and R. Krishnamurti  
GKS Inquiry Functions within Prolog  
Proc. Eurographics 85 Nice, France September 1985 Pages 185-191
- [12] W. Hubner and Z.I. Markov  
GKS Based Graphics Programming in PROLOG  
Computer Graphics Forum Volume 5 Number 1 Pages 41-50 March 1986
- [13] P. Wisskirchen  
Towards Object-Oriented Graphics Standards  
Proc. Eurographics 85 Nice, France September 1985 Pages 391-400
- [14] B. Stroustrup  
The C++ Programming Language Addison-Wesley 1986
- [15] B. W. Kernighan and D. M. Ritchie  
The C Programming Language Prentice-Hall [D 1978
- [16] ANSI  
C Language Binding of GKS (ISO version) ANSI X3H34/83-12R5 (Work Item)  
January 31 1986
- [17] C.D. Osland  
Case Study of GKS Development  
SERC Rutherford Appleton Laboratory August 1983
- [18] GKS-UK User Guide Program Library Unit, University of Edinburgh 1986

## An NRS processor in C and the future ...

*Piete Brooks*

*University of Cambridge Computer Laboratory*

*The NRS derived file format is the standard for interchange of information about JANET and PSS names and addresses. A set of interface routines are provided in the portable language, Fortran.*

*An alternative (written in C) has been provided which does not provide a direct interface for the users, but produces tables for other utilities (such as MMDF, sendmail, Yorkbox, VMS ...). This is largely described in its own documentation (A Quick Jog Through Configuring The Nrs Processor); this document describes future plans, namely a brief description of the proposed FORMAT3, a local NRS server and a proposal to adopt a standard NIFTP suite for UNIX.*

### *1. The C NRS processor*

The current programme will accept format 1 or 2 files (monolithic or as filettes) and produce from these tailorised tables for MMDF, UK-1.4 sendmail (and Reading's smail), Yorkbox (R2.2 and R2.1), VMS coloured book, x25hosts, text, UCL dbextract format DBM files, etc.. It consists of a parser for the tailoring info (e.g. what format, which domains to strip, etc.), the main munger which converts to an internal format and the format specific output routines.

A site will typically process the data twice; once to generate site name lists for the mailer (i.e. picking out the known MAIL hosts and the application relays) and again for the X25 level code to map between names and addresses.

This has been written in such a way that (so far) addition of a new input format or output format have been very minor matters, and porting to non-UNIX systems (VMS and PRIMOS) were also fairly painless. As such, this software is of little interest — it does what is expected of it.

### *2. FORMAT3*

After Salford's minor changes to FORMAT1, they realised that the existing format was not suitable for the extension to ISO and that the communication load on their server is too high, so they are designing a very different FORMAT3 (this has still to go before the steering committee). The two compression methods considered are removing duplicated strings and some form of incremental update. (They appear not to be interested in reducing the load by a factor of four by passing the file through compress) The incremental update involves having each section split into sorted and unsorted parts with the hope that the sorted will be much larger, so that the sequential search through the unsorted part should not slow it down significantly.

The ISO extension consists of additional contexts ftam, jtm, jtm-reg, vt and motis, the new network OSI-CONS, and the registration of NSAPs. The duplicated string removal is helped by splitting names and addresses into sections so that common substrings can be shared (i.e. names are prefix [uk.ac], institution [cam] and entity [cl.jenny] and addresses are DTE, YBTS or CUDF, Application Relay and ISO



information). Gateway information and institution descriptions have also been added.

Together these effectively force an implementor to explode the information to obtain complete strings which can be hashed internally rather than sorted externally.

There are several cases where it would be nice to have some keyed information available which although not essential is highly desirable in the real world, such as this machine should not have multiple transfers in a single call. Is there any simple way to extend the database to include this extra data? I can image that there may be a lot more black magic when the ISO code appears (such as the information that GECs only accept one transfer per TS connection).

Should the single centralised nameserver, now discarded elsewhere, be replaced by distributed servers? The obvious unit is the institution, with the current **INAMES** information available on all servers.

### 3. *Local NRS server*

As LANs have become the norm rather than the exception and sites are installing transport service relays between the networks they use, the number of hosts that can use WANs is rapidly increasing. If a host wants to make use of other networks it needs to have a complete and up to date copy of all the tables. The current lookup techniques tend to go for speed of lookup rather than keeping the data compact, so this is likely to burn up a lot of disc space if a distributed filing system is not available.

A solution would be to implement the **CHAIR** protocol (as defined by Salford) as it is the UK "standard". There are very few implementations available at the moment, so it may be better to implement a very simple ad-hoc connectionless protocol which would effectively map the user lookups into a very simple RPC. Somewhat more reasonable in the Berkeley world, would be to use **bind** to do the mapping. Some systems (e.g. **MMDF**) can already make use of **bind** information. If this could be extended to work on **SysV**, this would be the obvious candidate.

### 4. *Standard NIFTP*

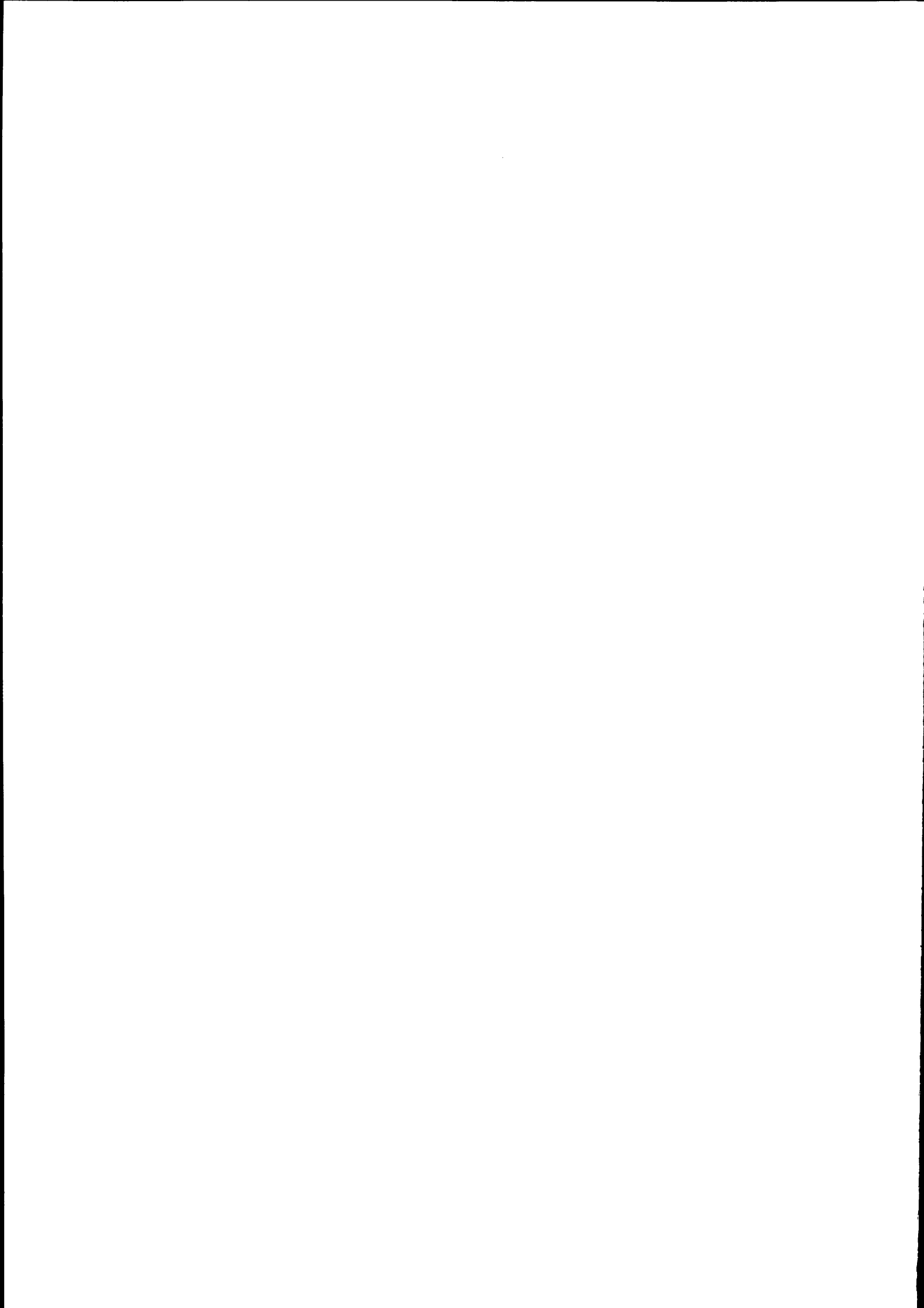
With the omission of a serious **standard** implementation of **NIFTP** for **UNIX** by the **JNT**, it appears that the user community will have to provide its own. With the final demise of the York contract, the **JNT** are continuing to throw vast sums of money at **Spider** in the short term, but they are planning a new scheme to have a manager so that companies wishing to offer **UNIX** coloured book software can take a standard package and need only write the interface to their own hardware or kernel.

**UKNET** has repeatedly been able to do this where all others have failed, so if we can get our act together and show the **JNT** that we can provide such a service, we may be able to get some support. The distribution network is already in place, including a dedicated machine. For the user the joy of this is that a new site can simply tar in a file, run the **Configure** programme, type **make install** and have a running system. That is all very well, but not my current prime concern.

I am more concerned about the future existence of a working, up to date suite for **UNIX** that can be used to make a clean transition to the new **ISO** implementations which are almost upon us :-). The sites which currently provide **UNIX** implementations are all having serious staff shortages, particularly in the communications area. Either they will have to freeze their software and live with the bugs, or divert effort from elsewhere. Now would seem the time to consolidate the old **NIFTP** code into a single source (several implementations already support

many underlying networks and interfaces) and have a UKUUCP style of update moderation. When the active contributors have been found we should be ready to plan the transition to OSI and be able to make considered representation to the JNT rather than having the current unfortunate situation where they have drifted away from their community.

The current candidate (the UCL code) supports sockets (ether, UBC X25, UCL ICPS), the Yorkbox, Camtec's dexpand and the ukc io module (which itself supports DECNET, ...). This is currently running with MMDF and UK-1.4 Sendmail. If this (or an equivalent package) could be supported then a large amount of the wasteful duplication of effort could be avoided.



---

The Secretary  
**European UNIX<sup>®</sup> systems User Group**  
Owles Hall  
Buntingford  
Herts. SG9 9PL  
Tel: Royston (0763) 73039.

\*UNIX is a Registered Trade Mark of AT&T in the USA and other Countries