Medlemsblad for

Dansk UNIX-system Bruger Gruppe

DKUUG-Nyt

Nummer 39, 1. april 1991

Indhold

Redaktionelt

DKUUG-Nyts redaktion består af Søren O. Jensen (ansvarshavende) og Christian Damsgaard Jensen.

Vi er naturligvis altid interesserede i indlæg fra folk. Det behøver ikke være lange artikler, men kan også være annonceringer, opfølgninger af tidligere artikler, eller andet. Hvis I blot har ønsker eller gode ideer til artikler, er I også meget velkomne til at kontakte os. Bidrag til bladet bør indleveres på maskinlæsbar form.

Indlæg, foreslag, ønsker, etc. til nr. 40 kan sendes med elektronisk post til redaktionen på adressen:

`dkuugnyt@dkuug.dk`

eller, hvis man foretrækker almindelig sneglepost, til:

Søren O. Jensen
Datalogisk Institut
Universitetsparken 1–3
2100 København Ø

Deadline for nr. 40 er d. 19. april
DKUUGs sekretariat har adressen

DKUUG, sekretariatet
Kabbelejevej 27B
2700 Brønshøj

Telefon: 31 60 66 80 (mandag, tirsdag og torsdag, kl. 13–14)
Telefax: 31 60 66 80 (autom. omskiftning telefon/telefax)
Giro: 1 37 86 00
Email: `sek@dkuug.dk`

DKUUGs netpassere: 31 39 73 22
Email: `netpasser@dkuug.dk`

DKUUG formand: 33 13 00 23
Email: keld@dkuug.dk

DKUUG Klubaften i København

Tema:
Pseudoterminaler og Remote Procedure Calls (RPC)
Foredragsholder:
Kim Dam Petersen
Teleteknisk Forsknings Laboratorium (TFL)

Tirsdag den 30. april 1991
Kl. 19:00 – 22:30

Datalogisk Institut (DIKU)
Universitetsparken 1

Kim vil give os en indføring i kommunikation mellem maskiner i netværk, distribueret databehandling og kunsten at få andre maskiner til at levere services til den lokale maskine.

Kim arbejder selv med en udvidelse af RPC kaldet Remote Modul Call (RMC), der tillader overførsel af procedure parametre (modul).

DKUUG har hermed fornøjelsen at annoncere den næste klubaften, der som sædvanligt holdes den sidste tirsdag i måneden.

DKUUG's klubaften-aktivitet i København tog sin begyndelse i marts måned sidste år og indtil videre har vi holdt flg. møder:

Marts 90	Lex og Yacc i dagligdagen
April 90	Sed's Hemmeligheder
Maj 90	Standardisering af UNIX
...	
September 90	UNIX og PC Systemer i netværk
Oktober 90	Streams programmering under Sys V.4
November 90	UUCP, smail og sendmail
December 90	Objektorienteret Programmering
Januar 91	X-Windows
Februar 91	Minix
Marts 91	GNU

Det er vores intention at disse klubaftener skal have et højere teknisk niveau i et mere uformelt, hyggeligt forum end de normale medlemsarrangementer. Hver gang har derfor et "aftenens emne", hvor en erfaren bruger eller programmør indenfor emnet holder et foredrag om emnet på ca. halvanden time, hvorefter resten af aftenen kan bruges til at fortsætte diskussionen af aftenens emne, eller tale med andre medlemmer om problemer og udveksle tips og ideer.

Klubaftenerne har således en klar faglig profil, der nok henvender sig til en snævrere kreds end de store medlemsmøder. Vi håber dog at også den “almindelige” UNIX-bruger vil kunne bruge klubaftenerne til at møde mere erfarne brugere og derigennem finde nogen til at hjælpe med evt. problemer, hvor de selv er kørt fast.

Klubben har fast mødested på Datalogisk Institut (DIKU) den sidste tirsdag i måneden kl 19:00. Døren til instituttet er normalt låst udenfor almindelig arbejdstid, så kom (rimelig) præcist. Tilmelding ikke påkrævet, men kan ske via E-mail til “klubben@dkuug.dk” eller på telefax 44 53 17 18 (Att. Myanne Olesen).

Vel mødt!

Internationaliseringsarbejdet i Dansk Standardiseringsråd

Af *Keld Simonsen*
DKUUG

Oversat af Søren Oskar Jensen
DKUUG-Nyt

Arbejdet i Dansk Standardiseringsråd (DS) med tegn og tegnsæt har baggrund i den specielle ISO/IEC JTC1/SC22 arbejdsgruppe om tegnsæts brug i programmeringssprog. De fremlagde i april 1989 en række krav om dette til SC2 — som svarede at de kun ville lave en lille del af arbejdet. SC2 har siden behandlet emnet tegnsæt i ISO/IEC DIS 10646:1990 — hvor mange (i sidste instans alle) af verdens tegnsæt er listede og udstyret med et unikt (langt) beskrivende navn.

DS-arbejdet har taget afsæt i 10646-standarden:

- Korte mnemoniske navne er blevet givet til omkring 1300 bogstaver og specialtegn og omkring 24000 ideografiske tegn fra 10646.

- Omkring 100 tegnsæt er blevet ordnet i tabelform med disse mnemoniske navne, inklusive næsten hele ISO (ECMA) registret og ca. 40 leverandør-tegnsæt.

- Omkring 300 navne og aliaser er blevet tildelt tegnsættene.

- Attributter til hver af disse tegn er blevet beskrevet i POSIX.2 localdef form (alpha, lower, upper, contrl, toupper, tolower, etc.).

- En sorteringsrækkefølge i POSIX.2 format er blevet defineret for sproget dansk. Den følger den danske standard DS 377 (1980). Sorteringsrækkefølgen er defineret for alle 25000 tegn, det er derfor muligt at have den samme sorteringsrækkefølge for de 100 tegnsæt der er på tabelform.

- Med indkodningen af de 100 tegnsæt er det blevet muligt at definere konvertering mellem næsten alle disse tegnsæt, med en reserve repræsentation der består af en indikator og det mnemoniske navn for tegnet. Rutiner til dette formål er blevet skrevet i C.
- Konverterings-rutinerne er blevet bygget ind i sendmail, således at multi-tegnsæts post understøttes. Dette bruges indtil videre af 10 danske email-sites.

Dette arbejde er gratis tilgængeligt fra DKUUG.dk via ftp, ftam og email (archive@dkuug.dk).

Projektet er nu dokumenteret eller anvendt i:

- ISO/IEC IS 9945-1:1990 (POSIX-kerne standarden) i det Informative Appendiks E.
- IEEE POSIX 1003.2 Shell og Utilities 10. udkast (og det kommende 11. udkast) i det Informative Appendiks F.

Resultatet er blevet fremlagt for SC22s specielle arbejdsgruppe om tegnsæts brug, for den europæiske workshop om åbne systemer (EWOS), for ISO/IEC JTC1/SC22/WG14 C-sprog gruppen, for RARE og IETF og nogle leverandører, herunder X/Open.

Der er igangværende arbejde i alle disse fora baseret på dette resultat, omend ingen af dem endnu har givet sin officielle tilslutning til det.

Endvidere er projektet blevet forelagt for ISO/IEC samt for JTC1/SC22/WG11 (programmeringssprogs-uafhængige features) som en mekanisme til at give tegnkodnings-uafhængige generelle tegnsæt.

Dansk POSIX locale

Til POSIX.2 standarden er der blevet lavet helt generelle tegnkodnings-uafhængig tegn-specifikationer og sorteringsrækkefølger.

Sammen med POSIXs charmaps kan dette bruges til at sortere ens i mange tegnsæt, inklusiv 10646 og næsten hele ECMA registret og ca. 40 leverandør-specifikke tegnsæt.

Der er mange forskellige komplikationer på forskellige niveauer i forbindelse med sortering. For eksempel på telefon-niveau, her på engelsk, hvor "Mc" er det samme som "Mac", tal bliver stavet, ord som f.eks. "the" bliver sidststillet, etc.

Dansk har også en række regler som disse, også i den oficielle sorterings-standard DS 377 fra 1980. Et andet niveau er det fonetiske — soundex, som dog er mindre kompliceret. Et tredie niveau er transkriberede tegn, som f.eks. bibliotekarer bruger når de ser et græsk "α" og sorterer det som et almindeligt "a".

Det niveau som DS har valgt til dets POSIX.2 locale er system-interface niveauet. Sorteringsrækkefølgen skulle kunne bruges i POSIX-værktøjer som `ls` og `sort`. Et krav har været at løsningen er deterministisk, dvs. at hvis strenge er forskellige skal de også være forskellige når de sammenlignes. Et andet mål har været effektivitet, POSIX har mulighed for at udskifte "Mc" med "Mac", men dette opfattes som for ineffektivt og undgås i den danske nationale eksempel-locale.

Problemerne med udtale og translitteration er ikke behandlet. Det er i stedet blevet betragtet som tilstrækkeligt at nøjes med at se på tegnene selv — kun at betragte tegnene på system-niveauet — og ikke lyde. Det niveau som den danske locale tilbyder er en service til at sammenligne strenge, tænkt som en afsløsning for den almindelige `strcmp()` o.lign. rutiner, bare i en lidt mere intelligent udgave og overholdende de danske sorteringsregler.

Vi har imidlertid stoppet så meget intelligens som muligt ind på dette niveau. De to bogstaver "aa" bliver sorteret som bogstavet "å", men "å" står før "aa" i homonymer (enslydende ord). Det canadisk-franske fire-niveaus plan for sortering bliver brugt, hvor de fire niveauer er bogstav, accent, case og specialtegn. Dette blev faktisk også speciferet i DS 377. Af hensyn til harmoniseringen besluttede vi at bruge den baglæns sortering af accenter som canadierne bruger. Det naturlige valg ville nok have været den forlæns sortering, men da de fleste af ordene alligevel er af fransk oprindelse besluttede vi at følge deres regler. For bogstavet "ß" implementerede vi hvad vi mener er den tyske regel (set i adskillige tyske ordbøger) at "ß" sorteres som "ss" men før det i homonymer.

Med hensyn til accenter er der nogle antydede regler i DS 377 og i den officielle danske retskrivningsordbog, men de er langt fra komplette. Jeg mener at vi har ca. 25 accenter sorterede.

Vi besluttede ikke at transkribere de ikke-latinske skrifter. Dette giver os mulighed for at bruge deres tegnsæts egne sorteringsrækkefølger, som α, β og γ på græsk. Græske og kyrilliske bogstaver med accenter og ligaturer sorteres korrekt.

Internationalt C-locale

Vi planlægger i ISO WG14 at tilbyde en internationalt C-locale, inklusiv en hel del tegn med accenter. Hermed kan en generel sorteringsspecifikation produceres. Den canadiske standardiserings-organisation har lavet en specifikation for fransk-canadisk som også kan anvendes for amerikansk-engelsk. Vi har planer om en sorterings-rækkefølge der kan bruges til fransk, engelsk, tysk, hollandsk, italiensk, japansk, kinesisk, arabisk, hebraisk, russisk, græsk og måske flere sprog, dvs. sprog som bare har den almindelige sorteringsrækkefølge.

Dette betyder at vi overvejer en "standard" latisk rækkefølge, sammen med en græsk, kyrillisk, arabisk, hebraisk, bopomofo og kana. En liste over udvalgte specialtegn og deres rækkefølge er også med.

At holde de forskellige sprog adskilt i sorterings-specifikationen giver den fordel at det er muligt at dække ganske mange kulturer uden at diskriminere. Japanerne har deres tegn i den "rigtige" rækkefølge, ligeledes russerne, araberne og de fleste sprog med latinsk alfabet.

Tutorials ved Tromsø-konferencen

EurOpen har annonceret en ekstra tutorial om netværks-sikkerhed ved konferencen i Tromsø. For fuldstændighedens skyld bringer vi her hele tutorial-programmet for konferencen:

Here is a list of the tutorials that will be on offer at the EurOpen (formerly known as "EUUG") Tromsø conference.

Tutorials are only available to members of National Groups affliated to EurOpen. Direct members of EurOpen may also book classes.

Further details, such as costs and discounted travel arrangements can be found in the Tromso conference booking brochure. If you have not already received a copy, then you will find it reprinted in the next copy of the EurOpen newsletter, or you get one sent to you by contacting the EurOpen Secretariat either by email (europen@eu.net), by phone (+44 763 73039) or by fax (+44 763 73255).

Details of the conference technical programme can be found from the same sources.

Tutorials are given in English.

EurOpen is always looking for new tutors, and welcomes proposals for classes that wil be of interest to its members. If you wish to propose a class, please contact me.

Neil Todd (neil@pio.gid.co.uk) EurOpen Tutorials Officer

Summary of classes

Monday 20th May

M1 Unix on Modern Architectures
Curt F. Schimmel

M2 Unix Standards
Susanne Smith and John Quarterman

M3 ISIS
Keith Marzullo

M4 Distributed Systems with Unix
Peter Meinen

M5 Network Security: The Kerberos Approach
John T. Kohl and Daniel E. Geer, Jr.

Tuesday 21st May

T5 RISC Architectures
Ashis Khan

T6 Writing Distributed Applications using ANSA
Cosmos Nicolaou

T7 Introduction to Mach
Nawaf Bitar

T8 Introduction to X concepts
Berry Kercheval

Note: All these classes are in an advanced state of preparation, but EurOpen cannot be held liable for events beyond its control that cause us to modify this programme.

Detail

M1 — Unix on Modern Architectures — Schimmel

This intensive tutorial presents the issues involved with porting the UNIX operating system to modern computer architectures that make use of multiprocessors and cache memories. Attendees will gain an understanding of the design considerations modern architectures present to the operating system and will gain insight into the design of new architectures intended to support the UNIX operating system. Examples of modern RISC processors and the computer systems built around them are used to illustrate the concepts.

The first section of the course investigates the effects of various cache memory systems on the UNIX System V porting base. After an overview of cache system architecture is presented, four different cache organisations ranging from pure virtual to pure physical caches are studied including the tradeoffs of each, the impact on the kernel, and how to modify the kernel to properly control the cache.

The second section presents tightly coupled, symmetric multiprocessors. This includes a discussion of the mutual exclusion, synchronisation, race conditions, and deadlock problems as they apply to the UNIX kernel. Several strategies for adapting the UNIX kernel to run on a multiprocessor are then presented, ranging from master/slave to multithreaded semaphore techniques, along with the tradeoffs of each approach.

The third section builds upon the first two by examining cache consistency in a multiprocessor system. An understanding of the cache consistency problems and the effects on the kernel is gained followed by an investigation of both hardware and software cache consistency algorithms for different cache organisations and multiprocessor kernel implementations.

The final section presents the differences between RISC Memory Management Units and the more traditional style MMUs. This includes Translation Lookaside Buffer (TLB) management, referenced and modified bit handling, and TLB flushing and replacement techniques. Emphasis is placed on the effects of the kernel and algorithmic changes needed.

Intended audience

This tutorial is targeted at system programmers with 6 months or more of UNIX kernel internals experience. It is ideally suited for those who will be porting UNIX System, V to a modern computer architecture in the future or those involved in the design of new computer architectures that need to support the UNIX operating system effectively. It is also for anyone who wants to learn more about operating systems and modern computer architectures.

Tutor

Curt Schimmel is an Operating System Architect at Amdahl Key Computer Laboratories. He received his M.S. in Computer Science from the Rensselaer Polytechnic Institute and his B.S. in Computer Science from the Rochester Institute of Technology with a minor in Computer Engineering. He has worked in many areas of UNIX kernel development and computer architecute including multiprocessor systems, demand paging, realtime, supercomputer enhancements, and kernel performance studies. For the past four years, he has been extensively involved in both the architectural and software design of large scale, high performance UNIX systems. Prior to this he was with AT&T Bell Labs where he worked on several releases of UNIX System V

M2 — UNIX Standards — Smith & Quarterman

In the past year, at least ten new committees to develop standards for the UNIX environment have been formed. In this expanding universe information that ties all the various work together is hard to find. This tutorial will provide just this sort of information. First, some of the reasons for standards are presented followed by the goals which standardisation hopes to achieve. Second, the processes used to make standards are discussed. The third part is an introduction to the bodies that actually form the standards. Fourth, and the bulk of the tutorial, is a description of the standards themselves. The following areas are presented: system interface, shell and utilities, conformance testing, profiles, real time, security, systems administration, networking and communications, programming languages and user interfaces. Last is a list of sources for additional information.

M3 — ISIS — Marzullo

Distributed systems are hard to program because their execution is asynchronous. Processes can fail at arbitrary times, messages can be delayed, and the real concurrency of a distributed system can make the resulting state of the distributed program very hard to reason about. The observation behind ISIS is that the (illusion) of synchronous execution can be captured without the resulting performance penalty of a fully synchronous system.

ISIS is a UNIX-based toolkit that has been ported to about a dozen different UNIX/platforms and distributed to over 750 sites, both academic and industrial. ISIS has been used to build or prototype a diverse set of programs, including factory floor managers, coarse-grain parallel graphic systems, stock brokerage systems, and fault-tolerant distribution services. In cooperation with Chorus and Mach groups, ISIS is currently being reworked to run as an external server, which we believe will lead to a very light-weight and flexible suite of tools.

This tutorial will consist of three parts. First the basics of fault-tolerance through active replication will be covered, including the structure and use of the protocols underlying ISIS and other similar systems. Then, the ISIS toolkit will be presented by working through several simple applications. Finally, an ambitious ISIS client, the Meta project, will be discussed. Meta extended the ISIS toolkit by supporting the decentralised control of distributed applications. This section will focus on taking a simple non-distributed application and using ISIS and Meta to make it distributed, fault-tolerant and adaptive.

Intended Audience

This tutorial will aimed for engineers and other practitioners of computer systems. A basic knowledge of UNIX, C and other communication will be assumed, but no other previous knowledge of distributed systems, replication or fault-tolerance will be necessary.

Tutor

Keith Marzullo is an assistant professor in the Computer Science Department of Cornell University, and is the co-principal investigator of the ISIS Research Project. His primary research focus has been the use and extension of ISIS to support reactive distributed control in both soft and hard real-time settings, and he is leading the development of the Meta project. He is also principal in ISIS Distributed Systems, a consulting firm specialising in distributed computing technologies.

M4 — Distributed Systems with UNIX — Meinen

The seminar introduces the basic concepts of distributed systems. It gives a survey of the architectures of distributed systems, the basic paradigms, possible problems, and the major design decisions to be taken.

The seminar is intended for system designers and developers who want to understand the design criteria of distributed systems and/or design and implement distributed systems with UNIX.

Prerequisites: basic understanding of UNIX; C for some examples.

Contents

- What are distributed systems?

  Motivation, application examples, history.

- Architecture of distributed systems

  Architectural models, design goals, client-server model, remote procedure call (RPC) model, RPC binding, RPC protocols, transparency issues, RPC semantics.

- Networks and protocols

  OSI reference model, connection oriented vs. connectionless services, examples of transport protocols (TCP, UDP, OSI-TP, OSI-CLTP); presentation protocols; purpose, examples (SUN XDR, OSI ASN.1); protocol compilers.

- UNIX Services at the transport layer

  Berkeley socket concept, Transport LEvel Interface (TLI) in Unix System V.

- SUN's ONC remote procedure call and presentation protocols

  RPC protocol, port mapping, XDR protocol, program generation with protocol compiler rpcgen.

- Design aspects of distributed systems

  Reliability, special RPC techniques, authentication by standard methods and by cryptography.

Tutor

Dr. Peter Meienen studied Mathematics and Computer Science at the Technical University in Munich.

1975-80 he was an assistant at the Computer Science Dept. of the Technical University, Munich, and a member of the computer architecture group.

1979 Doctoral thesis on computer hardware description languages.

1980 Softlab GmbH, Munich, designed and implemented UNIX-based software development tools for microprocessors. Trainer in UNIX, C and software engineering. ESPRIT project HTDS (host target development system). In 1987 he designed the distributed file system for Softlabs's MAESTRO-II IPSE and was responsible for the implementation of MAESTRO-II's UNIX-based file servers and mainframe communication links. 1989 he was dept head for OSI communication products.

In April 1990 he founded AKM Software GmbH, Munich, together with two colleagues. AKM specialises in UNIX-based distributed application systems and software engineering consultancy. Major projects include: Distributed high speed image processing system for medical prescriptions, control system for (geodetical) survey instruments.

M5 — Network Security: The Kerberos approach — Kohl & Geer

Overview

As the Internet has grown, its user community has changed from a small tightly-knit group of researchers trying to build networks into a loose gathering of people trying to use the global network. The amazing and constantly growing numbers of machines and users ensures that untrustworthy individuals have full access to that network. Given the increasing importance of the information transmitted, it is imperative to consider the basic security issues present as large, open networks replace isolated timesharing systems.

What you will learn

Attendees will gain an understanding of the kinds of security threats which result from operating in an open environment, such as one composed of a network of workstations and supporting servers. They will also understand approaches to meeting these threats. The focus will be on the Kerberos system developed at MIT, but public key techniques for ensuring privacy and authentication on an open network will also be covered. The X.509 authentication model will be covered as will the new Internet Privacy Enhanced Electronic Mail RFCs.

We will not discuss traditional timesharing or operating system security. The focus will be on what new challenges are presented by operating in today's open networking environment.

Syllabus

- Network Security Issues — generic security threats created when moving from isolated timesharing systems to network computing
- The role of encryption in security systems, public key and private key concepts
- The Kerberos Model — how the Kerberos system works and meets the network security threats
- The X.509 Model
- Comparison of the existing authentication models
- Existing Applications — how existing applications have been modified to incorporate network security.
- Privacy Enhanced Electronic mail
- Future directions — issues that still remain to be addressed

Intended audience

Systems developers responsible for networked workstation environments, particularly those whose environments may include networks which are not themselves physically secure (i.e. "open" networks). Systems managers concerned about the inherent lack of security for managing today's network-based environments (e.g. UNIX's .rhosts files).

Tutors

Daniel E. Geer, Jr., is now a member of the Technical Staff of Digital Equipment Corporation's External Research Program. For the previous 4.5 years, he was the Manager of Systems Development for MIT's Project Athena where he oversaw the creation of the Athena distributed computing environment, including the work that forms the basis for this tutorial. He holds a Bachelor of Science in Electrical Engineering and Computer Science from MIT, and a Doctor of Science in Biostatistics from Harvard University.

John T. Kohl is a Software Engineer with Digital Equipment Corporation, working at MIT's Project Athena for the past three years. He is the Project Leader for Kerberos development, and has worked on many other aspects of the Project Athena computing environment. Mr Kohl received a Bachelor of Science in Computer Science and Engineering from MIT.

T5 — Risc Architectures — Khan

The idea behind RISC architectures — relagating complexities to software — was well received in the industry. The first generation RISC processors attempted to achieve one goal: reduce the number of processor clocks taken to execute one instruction (C.P.I. or clocks per intruction) to one, by removing restrictions such as condition codes, complex addressing modes etc.

The second generation RISC processors saw two important developments to achieve higher performance. One was to increase the clock speed by taking advantage of exotic technologies such as ECL and the other approach was to develop newer pipeline techniques that allow more than one instruction being executed in one cycle.

The seminar will dwell on these two approaches. Higher processor clock rates widen the gap between the processor speed and the main memory speed. Various techniques had evolved to smooth this mismatch — e.g. multi-level cache, two-level translation lookaside buffer etc. Some implementation issues in ECL, GaAs and CMOS technologies pertinent to CPU design will be discussed. Case studies of commercial implementation will be presented.

The merits and demerits of new pipeline techinques, such as Superpipeline, Superscalar, VLIW etc will be covered. Real life examples will be studied.

The next topic in this seminar will be software for the RISC architectures. We will discuss what the various requirements for O/S and compilers are, and how architecture and implementations satisfy these requirements. Case studies such as MIPS Compilers and MIPS O/S implementations will be made.

Issues covered will include:

- Superscaler, Superpipeline, VLIW Architectures
- Architecture and implementation
- Register set design
- Computational Bandwidth and Latency
- Cache Architecture
- Pipeline design and compiler scheduling
- O/S Considerations in a given Architecture
- Effects of Compilers on Performance
- Optimisation Techniques
- Case study of Processor Architectures

Note: More complete details of this class can be found in the booking brochure

Tutor

Ashis Khan is from MIPS Computer Systems, Sunnyvale California. Ashis provides consultation on systems designing with MIPS RISC architecture and conducts seminars worldwide on contemporary RISC designs. He has published several articles on RISC architectural issues and has chaired sessions in MIDCON and WESCON conferences.

T6 — Writing Distributed Applications using ANSA — Nicolaou

ANSA is an architecture for distributed systems, developed partially under the ESPRIT Integrated Systems Architecture project. Work has included tackling the problems involved in writing applications for distributed heterogeneous multi-vendor computer systems. From this work a software suite known as ANSAware has evolved, as an exemplar implementation of the architecture. ANSAware provides a platform for the creation, deployment and management of distributed applications.

This tutorial covers a brief tour of the ANSA architecture and the problems inherent in distributed systems, followed by an in-depth look at the way ANSAware assists the applications programmer in dealing with these problems. This will be a hands-on course, with the opportunity to explore the facilities provided by ANSAware whilst creating a distributed application. The tutorial ends with a discussion of the future directions for distributed systems in general and ANSAware in particular. Familiarity with C is required.

Tutor

Cosmos Nicolaou is a member of the ANSA Research Team, with a special interest in multi-media, and plays a significant role in the continuing development of ANSAware

T7 — Introduction to Mach — Bitar

Mach is a novel operating system intended to recapture the original structural simplicity of UNIX while providing support for modern multiprocessor and distributed systems. The tutorial is intended for systems developers and technical managers who would like to learn more about Mach, which also forms the basis of OSF/1, the Open Software Foundation's first operating system release.

The tutorial will study the Mach operating system in detail. It will first cover the Mach architecture, philosophy and vision and continue with a thorough study of the three major subsystems that comprise the kernel: task/thread management, virtual memory management and inter-task communication. Topics include external memory management, copy-onwrite optimisations, message passing, and thread scheduling. Next will be a discussion of the facilities provided under the Mach environment including the Mach Interfae Generator, the Network Message Server. Finally, the tutorial will conclude with a presentation of Mach's future direction focusing on the micro-kernel architecture and dekernelisation of UNIX.

Tutor

Nawaf Bitar is a software engineer in the operating systems group of the Apollo Systems Division of Hewlett-Packard Company where he is the Project Engineer for the HP OSF/DCE project. Previously, while at the OSF Research Institue, he was part of a cooperative effort with CMU to develop a Mach 3 based system. Prior to joining OSF, he initiated the Mach project at Apollo Computer Inc and continued his integral involvement through its evolution to the OSF/1 project.

T8 — Introduction to X concepts — Kercheval

This tutorial will give the attendee a solid grounding in the concepts and techniques needed to begin writing applications for the X Window system. While the use of a toolkit such as Xt is encouraged, a good grounding in Xlib, the "assembly language" of X, is invaluable for comprehending the interaction of X clients with the server and workstation. The tutorial will cover the topics in a "depth first" manner, emphasizing the concepts rather than covering every single Xlib function.

Topics covered will include: The X Programming Model; Basic Definitions; Xlib linkage; Events; Graphics; Contexts; Drawing Graphics; Drawing Text; Colour; Resourses; Properties and Interclient Communication Conventions.

A sample program, especially written for this tutorial, will be analysed as the tutorial progresses.

Tutor

Berry Kercheval received his B.S. degree from university of California in 1977. Since then he has worked in Zehntel Inc., the Lawrence Livermore National Laboratory and most recently, Intelligent Decisions Inc., a small consulting firm located in Sunnyvale, California. His interests include: computer graphics, simulation and compiler design.

AALBUG - Aalborg Unix bruger gruppe

PERL – Et effektivt interaktivt sprog
Tirsdag den 23. april 1991
Kl. 19:00 - 22:30
Aalborg Universitetscenter
Institut for eletroniske systemer
Fredrik Bajersvej 7, bygning D2

DIFAN og DKUUG (Dansk Unix Bruger Gruppe) indbyder til månedlige klubaftner i “Aalborg Unix bruger gruppe” den næstsidste tirsdag i hver måned, tid og sted som ovenfor.

Intentionen med AALBUG er at give UNIX brugere og systemudviklere i det nordjydske mulighed for at mødes med ligesindede.

Lars P. Fischer der er kandidatstipendiat paa AUC, vil fortælle om aftenens emne PERL. PERL er et forholdsvis nyt sprog, der kombinerer det bedste fra "C", "sed", "awk" og "sh". PERL er på trods af, at det er et interkativt sprog meget effektiv og bliver blandt andet derfor brugt til at lave prototyper af programmer.

Der vil være kaffe og te ad libitum, men øl og vand kan købes. Tilmelding er ikke strengt nødvendig, men meld dig alligevel til, således at den fornødne proviant kan tilvejebringes.

Tilmelding:

E-mail: aalbug@dkuug.dk
telefax: 98 15 17 39 (att. Peter L. Petersen)
telefon: 98 15 85 22 (4816)
alm. post: Peter L. Petersen
Inst 8, AUC
Fredrik Bajersvej 7C
9220 Aalborg

Vel Mødt.

Gåhjem møde

DIFDATA indbyder medlemer af DIFDATA, DKUUG og DIFAN til gratis gåhjem møde.

Onsdag d. 17/4-1991. kl. 1500-1700
Kommunedatas udviklingscenter
Selma Lagerløfsvej 300, 9220 Aalborg Øst

Gennemgang af Kommunedatas nye kabelsystem.

Nettopologi.
Fiberoptik-udstyr.
Servere.
v. Jens Peter Johansen Kommunedata.

Kommunedatas OS/2-servere på nettet.
Designparametre,
Installationserfaringer,
Drift.
v. Bo Simonsen Kommunedata.

Nykredit's OS/2–servere på net.
Design parametre,
Installationserfaringer,
Drift,
Sammenfatning af erfaringer fra begge net.
v. Niels R. Pedersen Nykredit.

Med venlig Hilsen
Jan Schultz,
DIFDATA

The X Window System

Af *Per Bech Hansen*
DDE

Preface

This article gives an overview of the X Window System with an emphasis on the concepts that are most important for a good understanding of X. For application programmers and end-users, one of the most interesting aspects of X is the set of toolkits for building graphical user interfaces. Hence, the other purpose with this document is to explain the fundamentals of X based toolkits, and characterise a number of the toolkit products that are available in the marketplace.

Introduction to X

The X system contains a huge amount of code, but the archi- tecture is actually quite simple.

X Window System Concepts

X is a windowing system for bitmapped graphics displays. It supports colour as well as monochrome and gray-scale displays. The word *display* has a slightly unusual meaning here. It is defined as a workstation consisting of a keyboard, pointing device such as a mouse, and *one or more* screens. Thus, X allows multiple screens to work together, with mouse movement across screen boundaries etc. Usually a display has only a single screen, though.

The next thing to note is that X is a networkoriented windowing system. An application need not be running on the same system that actually supports the display. While some applications may execute locally, other applications may execute on other machines, sending requests across the network to a particular display, and receiving keyboard and pointer events from the system controlling the display.

The system that controls the display may even be dedicated to this sole purpose, i.e., no applications run locally. Such a system is called an *X terminal.*

At this point, only TCP/IP and DECnet networks are supported, but support for OSI is in the pipeline.

The program that controls a display is known as a *server.* This usage of the term server may seem odd. Usually, when you sit at a workstation, a server is something across the network like a file server or a print server. The thing to remember is that your display is accessible to other systems across the network, and for those systems the code executing in your system does act as a true display server.

Hence, the X system is built over *the client–server model*, and accordingly, applications are also referred to as clients.

X Window System Software Architecture

The rules that an X server and an X client follow when they exchange information across the network, such as mouse movements and keypress values from the user or bitmaps and drawing commands from the application, are known as the *X Protocol.* The programming interface to the X Protocol on the client side is called *Xlib.* Xlib contains almost 300 routines, and they seldom take less than 5 parameters. Xlib provides functions for connecting to a particular server, creating windows, drawing graphics, responding to events, and so on.

The X Window System may be shown as in Figure 1, overleaf. A particular client called the Window Manager is shown. This is just another program written with Xlib, except that it has special authority to control the layout of windows on the screen.

Versions of X

Most products for X are built for X11 release 3 at this point in time. X11 release 4 has just been released from MIT.

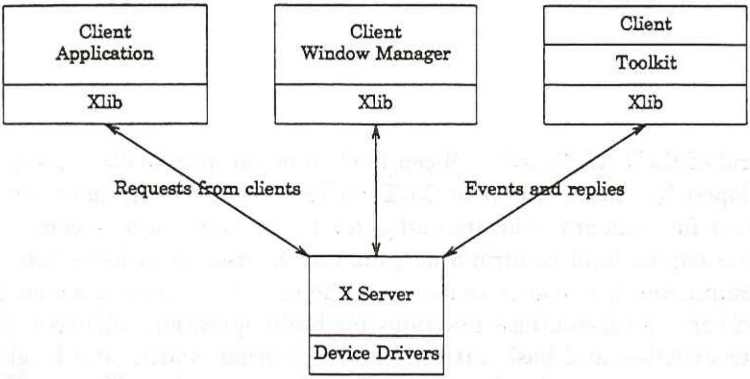

Figure 1. Clients communicate with the server via Xlib calls

X Toolkits

As alluded to above, writing programs to Xlib can be very tedious. X Toolkits are higher level routines collected in libraries which allow application programmers to code a certain type of applications much faster than with Xlib directly. User Interface Toolkits are the particular kind of X toolkits that help application programmers in writing user interfaces to X applications.

Another thing to note about toolkits, that is actually just as important as the increase in programmer productivity, is that they imply a certain *look-and-feel*, i.e., a specific set of controls such as menus, scrollbars, buttons etc., that look in a specific way and are driven by the user in a specific way.

Widgets and Gadgets

Toolkits provide an easy way to construct the controls of a user interface. Such controls – menus, pushbuttons, scrollbars, etc. – are often referred to as *widgets* or *gadgets*. Within a particular toolkit the words may have a more specific meaning, e.g., in OSF/Motif widgets are the standard ones, while gadgets are lean widgets that consume less memory.

Toolkit Architecture

Several of the X toolkits have been built upon an intermediate toolkit, developed by the X group at MIT, called Xt. Xt is actually both a set of fundamental widgets and a model for how more specialized widgets can be built to form a new toolkit. Xt uses an object oriented programming style to achieve its goal: The most fundamental widget is called core and it contains functions for basic operations like creation and destruction and basic attributes like position, width, and height, that are common to all widgets. Other widgets inherit operations and attributes from core and add further operations to be able to do their job.

The programming interface to Xt in the C language is called Xt Intrinsics. Since C is not an object oriented language, the Intrinsics use C in a rather unusual way, and it may seem weird at first. It is, however, all done in very consistent way and Xt has proved its value by serving as the basis of several toolkits.

Toolkits built upon Xt inherit the object oriented way of programming, and they get a programming interface that we will refer to as the *native API* of the toolkit.

A toolkit based application may use features from both Xt and Xlib as well as the toolkit. Hence, an application built with the XYZ Toolkit can be illustrated as in Figure 2.

Some of the toolkits that have been built with Xt are Athena widgets (MIT and DEC), AT&T OPEN LOOK toolkit, OSF/Motif, DEC Windows, and the Hewlett-Packard widget set (PM/X).

Further examples of XGUI toolkits are the Andrew Toolkit (MIT and Carnegie- Mellon), XView (Sun Microsystems).

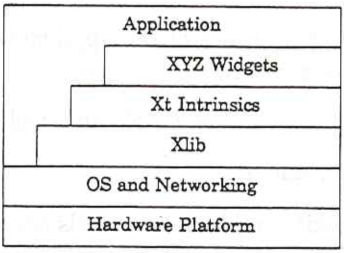

Figure 2. Application built with the Xt based toolkit XYZ

Toolkits and Look–and–Feels

The X system, Xlib, and Xt have explicitly been designed to be *policy free*, i.e., they can be used to implement any look–and–feel. An XGUI toolkit, on the other hand, implements a specific look–and–feel in the sense that an application, built with the toolkit, will conform to the look–and–feel of the toolkit.

A look–and–feel is specified in a *look–and–feel specification*, which defines exactly what the bitmaps for the controls should be like, how the user interacts with the controls, how the mouse buttons are to be interpreted in every situation, what the keyboard must be like, etc. Some- times the look-and-feel specification comes in a more read- able form with examples and recommendations of good user interface design. It is then usually referred to as a *Style Guide*. Examples of look-and-feels that have specifications independent of their implementations are:

- Apple Macintosh look-and-feel.
- OS/2 Presentation Managerlook–and–feel.
- Sun/AT&T OPEN LOOK look–and–feel.

It is important to note that the look–and–feel is (supposed to be) independent of the underlying technology. This is why it makes sense to talk about, say, implementing the Macintosh look–and–feel on UNIX under X.

Another thing that it is important to note, is that the look–and–feel is a property of the application. There are (at least) three ways to get a certain look–and–feel for an application:

1. Implement the user interface of the application according to the look–and–feel specification.
2. Use a toolkit that provides the look–and–feel.
3. Use amixture of 1. and 2.

The reason why toolkits and look–and–feels are always tied so close together, is that a toolkit is the only *practical* way to enforce a particular look–and–feel.

System Level Applications

One of the parameters, that have been used to promote XGUI toolkits, is is the number and quality of *System Level Applications*, that come along with the toolkit. Most toolkits have a Window Manager of their own because invocation of window manager functions like window resize, window movement, iconification etc. is often part of the look–and–feel.

But toolkits may have additional managers such as a Resource Manager, which lets the user configure the X environment in much more friendly way than otherwise possible, or a File Manager which provides a modern way of accessing the UNIX file system. One can argue that such additional applica- tions don't belong to the toolkit. They are merely applications built with toolkit and together they they form a nice environment.

Further Development Tools

The evolution of X does not stop with the toolkits. Tools that aid the application developer already exist. There are two types. One type is called *user interface management systems* (UIMSs). A UIMS allows you to design your interface interactively and then typically produces C code with native API toolkit calls. Examples of UIMSs for X are:

- UIMX from Visual Edge Software.

- TeleUse from TeleSoft AB.
- Exocode from Expert Object Corporation.
- Builder Xcessory form Integrated Computer.

The *Interface Builder* from NeXT is perhaps the most well–known example of this kind of tool, but Interface Builder and the underlying NextStep tools are *not* based on X Windows technology.

The other kind is somewhat more down–to–earth in that they need the application programmer to specify the widget–hierarchy in a separate module and language. The programmer can then have her C code functions, that contain the logic of the application, hooked onto the widgets through a binding process. The Motif *User Interface Language*, UIL, is this kind of tool.

Classification of Toolkits

The table below shows aspects of a number of different toolkits.

Toolkit	Vendor	Look-and-feel	API	Sys. Lvl. Appls.
OPEN LOOK	AT&T	OPEN LOOK	Xt based	WM, RM, FM
XView	Sun	OPEN LOOK	Close to SunView	WM
OSF/Motif	OSF	Motif (PM)	Xt based + UIL/MRM	WM
Athena Widgets	MIT	Athena	Xt based	None
X.Desktop	IXI	Motif (PM)	IXI's own	Full desktop
Looking Glass	Visix	OPEN LOOK	Visix's own	Full desktop
Looking Glass	Visix	Motif (PM)	Visix's own	Full desktop
HP Widgets	HP	Motif (PM)	Xt based	WM, ..
XUI	DEC	DEC Windows	Xt based + UIL/MRM	WM, ..
Object Interface	Solbourne	OPEN LOOK/Motif	Xt based (C++)	?

The abbreviations used in the table are FM, RM, and WM for File Manager, Resource Manager, and Window Manager respectively, and PM for the OS/2 Presentation Manager look–and–feel, which is virtually the same as the Motif look–and–feel. SunView is the toolkit for Sun's own networked windowing system (NeWS). The HP Widgets are sometimes referred to as PM/X. OSF/Motif was created by glueing together the look–and–feel part of HP's toolkit and the API part from DEC's XUI. Solbourne's toolkit is an OPEN LOOK toolkit that optionally supports a Motif look, i.e., the controls can have a Motif appearance wherever possible. It is only available in a C++ version.

References

- [1] "AT&T OPEN LOOK Graphical User Interface. Programmer's Guide and Programmer's Reference Manual". AT&T 1988.

- [2] "AT&T OPEN LOOK Graphical User Interface. Specification Guide" AT&T 1988.

- [3] "AT&T OPEN LOOK Graphical User Interface. Style Guide" AT&T 1988.

- [4] "AT&T OPEN LOOK Graphical User Interface. Trademark Guide" AT&T 1988.

- [5] "AT&T OPEN LOOK Graphical User Interface. User's Guide" AT&T 1988.

- [6] Jones, Oliver, "Introduction to the X Window System," Prentice Hall, 1989. A fine introduction to programming with Xlib; fairly good background to the X protocol; nice discussion of Xlib, the X library. ISBN 0-13-499997-5. (There is a new version. ISBN?)

- [7] Nye, Adrian, "Xlib Programming Manual, Volume 1" and "Xlib Reference Manual, Volume 2," O'Reilly and Associates, 1988. A superset of the MIT X documentation; the first volume is a tutorial. ISBN 0-937175-26-9 (volume 1) and ISBN 0-937175-27-7 (volume 2).

- [8] Nye, Adrian, and Tim O'Reilly, "X Toolkit Programming Manual, Volume 4," O'Reilly and Associates, 1989. The folks at O'Reilly give their treatment to programming with the MIT X11R3 Intrinsics.

- [9] O'Reilly, Tim, ed., "X Toolkit Reference Manual, Volume 5," O'Reilly and Associates, 1989. A reference manual for the MIT X11R3 Xt.

- [10] "OSF/Motif Programmer's Guide." Revision 1.0. Open Software Foundation. 1989.

- [11] "OSF/Motif Style Guide." Revision 1.0. Open Software Foundation. 1989.

- [12] Rosenthal, David S.H., "Inter-Client Communication Conventions Manual Version 1.0 (MIT Consortium Standard)." The first real ICCCM, available on the R4 tape; a version is also available from the xstuff mail-archive-server.

- [13] Scheifler, Robert, James Gettys, and Ron Newman, "X Window System: C Library and Protocol Reference," Digital Press, 1988. The bible on X. This is the most complete published description of the X programming interface and X protocol. ISBN 1-55558-012-2. DP order number EY-6737E-DP.

- [14] Unigram X. Jan 29 - Feb 2, 1990.

- [15] "UNIX in the Office." Patricia Seybold's Office Computing Group. Vol.4, No.9. September 1989.

- [16]McCormack, Joel, Paul Asente, Ralph R. Swick, "X Toolkit Intrinsics – C Language Interface". Part of the X11 release 3 distribution. Massachusetts Institute of Technology, 1988. [Newer version with X11R4].

- [17] Young, Doug. "The X Window System: Applications and Programming with Xt (Motif Version)," Prentice Hall, 1989 (ISBN 0-13-497-074-8). The excellent tutorial "X Window Systems Programming and Applications with Xt," (ISBN 0-13-972167-3) updated for Motif. [The examples from the Motif version are available on expo in ftp/contrib/young.motif.tar.Z]

Ny EUnet-manager

Af *Keld Simonsen*
DKUUG

EUnet (som forhåbentlig ikke behøver yderligere præsentation) har ansat Glenn Kovacs som manager. Hvem han er og baggrunden for hans ansættelse er emnet for denne artikel.

Baggrund

Baggrunden for Glenn Kovacs ansættelse er, at EurOpen (som før hed EUUG) gerne vil højne sit serviceniveau. Det blev således besluttet på EurOpens repræsentantskabsmøde i Nice efteråret 1990 at man skulle ansætte managers for nettet, konferencerne, standardisering og en general manager. Glen Kovacs er altså net-manageren, og der er forhandlinger med andre personer om besættelse af de andre poster.

EUnet er jo et glimrende net, men det er meget opdelt på servicene i de forskellige lande — og serviceniveauet (og priserne) er stærkt svingende fra land til land. Der har manglet en overordnet styring. Mange af EUnets planer er heller ikke kommet så hurtigt på banen som det var planlagt. Derfor har EurOpen besluttet at en styrkelse af ledelsen af Eunet var påkrævet. En af Glenn Kovacs første opgaver er at ansætte en person, der kan stå for udbygningen af EUnets arkiv-service. Desuden har han været involveret i en række vigtige forhandlinger, herunder forhandlinger om EUnet i Svejts og Nordunet forhandlinger.

Præsentation af Glenn Kovacs

Glenn har de sidste 14 år arbejdet indenfor software-udvikling og beslægtede områder. I slutningen af 70'erne og begyndelsen af 80'erne arbejdede han med udviklingen af TCP/IP.

I midten af 80'erne ledede han et center for Unix system-udvikling med 100 ansatte. På dette sted blev der bl.a. arbejdet med udvikling af netværks-programmel og real-time og multi-CPU modifikationer til Unix.

I de sidste par år har Glenn arbejdet som konsulent. Hans kunder har bl.a. været Unix International og Harris Semiconductor.

Samtidigt med disse erhvervs-aktiviteter har han været med til at oprette og drive en lokal-radio. Han har i flere perioder været i bestyrelsen for lokal-radioen, samt i en periode været formand. Han havde i flere år sit eget program på denne radio.

Glenn er uddannet på University of Illinois at Urbana-Champaign i matematik og psykologi.

OPEN SOFTWARE FOUNDATION

Af *Lars Kalsen*
Aalborg Universitets Datacenter

Åbne systemer er 90'ernes udfordring til EDB-verdenen. Samspillet mellem systemerne skal nu udnyttes fuldt ud, og en mængde tiltag er da også i gang hos brugerne, leverandørerne, softwareudviklerne og standardorganisationerne. Denne udvikling vil puste nyt liv i anvendelser af informationsteknologien, og samvirket mellem systemerne vil synliggøre helt nye anvendelser af teknologien. Informationsteknologien vil i 90'erne komme ind i en stejl udviklingsspiral, med de åbne systemer som akse.

Som et element i denne udvikling holdt Open Software Foundation (OSF) i dagene 20–22. februar et tredages teknisk seminar om de tre produkter denne organisationen indtil nu har på programmet. Det skete som led i en rundrejse til flere europæiske lande, og København var valgt som stedet, hvor de nordiske deltagere kunne mødes.

OSF

OSF oprettedes, som det måske er bekendt, i maj 1988 som en non– profit organisation af nogle af EDB-branchens største leverandører. Formålet var at stimulere udviklingen henimod åbne systemer og teknologien omkring dette koncept. Åbne systemer er blevet nært beslægtet med UNIX, og formålet har da også groft taget været at udvikle en fælles UNIX platform. Det er sket i skarp konkurrence med den anden store UNIX organisation UNIX International (UI), der lancerede sin UNIX version - System V, Release 4 i oktober 1989.

Siden starten er antallet af medlemmer i OSF vokset stærkt, og der er nu over 200, hvilket inkluderer system leverandører, hardware leverandører, uafhængige software leverandører, halvlederfabrikanter, servicebureauer, undervisnings- og forskningsinstitutioner, konsulentvirksomheder, offentlige institutioner og andre. Det er jo et interessant blandet forum, og man kan spørge sig selv, om der kan komme noget konstruktivt ud af et sådant konglomerat. Man kunne formode, at en dyne ville lægge sig kvælende over beslutningsprocessen, når så mange deltagere skal afveje deres synspunkter. Skal man dømme efter de ting, der blev fremlagt på seminaret, er det imidlertid ikke tilfældet. Idag står organisationen med tre forskellige produkter, klar til at blive brugt som platforme.

OSF/Motif

Motif er bygget på ingredienser fra Digital Equipment, Hewlett–Pachard og Microsoft Corporation. Systemet kører på en række forskellige hardwareplatforme, heriblandt følgende arkitekturer SPARC, MIPS R2000 og R3000, INTEL 80286, 80386, 80486, i860 og Motorola 68020, 68030, 68040 og 68000. Motif er udviklet til at understøtte en række standarder herunder ICCCM - Inter Client Communications Conventions, som gør det muligt at dele data og netværksressourcer med andre applikationer, der overholder standarden. Desuden overholdes X/Open's XPG3 standard, og selvfølgelig følger OSF/Motif X standarden for kommunikationen over netværket.

OSF/Motifs store konkurrent på markedet er OPEN LOOK fra AT&T. OPEN LOOK overholder i dag ikke ICCCM standarden, og det kan måske give problemer med samspillet i et heterogent netværksmiljø.

Hvem der vinder den endelige gunst hos brugerne er endnu ikke afgjort, men under seminaret talte jeg med en, der sagde at på den sidste udstilling i BELLA CENTRET i februar var der masser af demonstrationer af OSF/Motif, men OPEN LOOK var ikke til at drive op. Det giver måske et fingerpeg.

DCE – Distributed Computing Environment

Behovet for et distribueret miljø er vokset i takt med at PC'er, arbejdsstationer, LAN's og WAN's har vundet indpas sidst i 80'erne. Brugernes interesse er at kunne sammenstille et system af standardkomponenter fra forskellige leverandører, således at de uden videre kan spille sammen – eller samvirke, som det nye danske ord er. OSF har også følt behovet og har sammensat et koncept, som indeholder en række elementer. DCE er en samling af tjenester, der understøtter udviklingen af applikationer til et distribueret EDB–miljø.

OSF/1

OSF/1 operativsystemet er bygget op ovenpå en MACH kerne, der er udviklet på Carnegie-Mellon Universitet i USA. MACH kernen er speciel velegnet til at håndtere parallelprocessing, hvilket må forventes at få stor udbredelse i 90'erne, hvor EDB-maskinerne vil blive udstyret med flere end en processor. OSF/1 kan kort karakteriseres i følgende punkter

- MACH kerne hvorpå UNIX bygges.
- UNIX parallel udgave af 4.4BSD (Berkeley UNIX).
- Logisk volume manager (filsystemer kan f.eks. deles over flere diske).
- Stream support fra System V rel 3.
- Ny loader der blandt andet tillader loading ved run-time.
- Dynamisk konfigureing, således at f.eks. nye devicedrivere kan tilføjes til et kørende system.
- B1 sikkerhed overholdes, dog ikke certificeret.

Digital Equipment viste på deres stand et kørende OSF/1 system, og oplyste, at man eksperimenterede med at overføre binære programmer fra Ultrix til OSF/1, hvorefter mange af dem uden vanskeligheder kunne eksekveres. Det må formodes at OSF/1 i fremtiden bliver en fælles platform for medlemmerne af OSF, hvorpå de enkelte leverandører så bygger deres egne UNIX platforme. Softwareprodukter der kun udnytter de rene OSF/1 faciliteter vil derfor kunne afvikles på alle platformene.

Er standarderne vejen frem i 90'erne ?

Det bliver interessant at følge udviklingen fremover. Vil UNIX International eller OSF vinde kampen om at have udviklet 90'ernes åbne operativsystem og koncepter omkring åbne systemer, eller vil begge systemer leve side om side. Eller vil noget helt tredie dukke op, som et resultat af, at det viser sig, at den udvikling der nu er i gang, hvor organisationerne samarbejder om at definere nye produkter og sætte standarder, går op i limningen og ikke viser sig brugbar. Det kan kun fremtiden vise.

Deadlines

Dette nummer af DKUUG-Nyt er forsinket, dels på grund af påsken, men hovedsagligt fordi diverse skribenter ikke overholder bladets deadlines — dette gør i sidste ende bladet dårligere, idet annonceringer o.lign. ikke kommer ud i tilstrækkelig god tid.

Vi appellerer derfor endnu engang til folks samvittighed og anmoder om at bladets deadlines (se s. 2) overholdes.

Redaktionen

DKUUG - Dansk UNIX-system Bruger Gruppe
Bestillingsliste vedr. medlemstilbud

Listen sendes til:

DKUUG
Sekretariatet
Kabbelejevej 27 B
2700 Brønshøj

(telefon og telefax: 31 60 66 80)

Afsender:
Medlemsnr.:

att:

Medlems-
navn og
adresse:

Medlemsnavn og adresse tages normalt fra vor database, men bedes angivet her (gerne stempel) af hensyn til kontrol.

Prissatte medlemstilbud (priser i kr EXCL. moms)	Antal	Medl.pris	Beløb
UNIX-bogen (dansk udgave af "UNIX - the book")		170,00	
Dansk UNIX markedsoversigt, 3. udgave 1989-10		70,00	
UniForum products catalog 1990		450,00	
Beløbet er: [ ] vedlagt i check. [ ] betalt giro 1 37 86 00. Ialt: [ ] ikke betalt. I alle tilfælde sendes faktura (evt kvitteret).			

Overskydende sæt af foredragsnoter fra medlemsmøderne tilsendes mod et ekspeditionsgebyr på 50 kr + moms. Ring og hør, om vi har det ønskede på lager.

Øvrige medlemstilbud, der fremsendes gratis	
Tilmeldingsblanket/rekvisition til:	Antal
- Ekstra abonnement på DKUUG udsendelser (abonnementet er gratis, højst 2 stk pr. organisationsmedlem, højst 9 stk pr. stormedlem)	
- Ekstra abonnement på EUUG Newsletter og DKUUG udsendelser (350,- DKK pr. år, kun org.- og stormedlemmer samt studerende) ..	
- Medlemsskab af UniForum (tidligere /usr/group) (ca. ??,-/??,- USD pr år for associeret/generelt medlemsskab) ...	
- Abonnement på PC World og/eller Computerworld (50 % af normal abonnementspris)	
DKUUG's pjece (vær ikke for tilbageholdende)	
DKUUG Nyt specialnummer Kontor & Data 90 (også til PR-formål)	
DKUUG's medlemsliste (overdriv ikke)	
Medlemsinformation (vedtægter, formandsberetning, regnskab, budget) ..	
Netinformation (m. tilmeldingsblanket for login/post/nyheder)	
Magnetbåndsinformation (m. bestillingsblanket for "gratisprogrammel")	

Dato:

Underskrift:

Forbeholdt DKUUG:
Modt. d.

Eksp. d.

Medltilb 90-10-22

Oversigt over medlemsmøder i 1990-1991

Dato	Sted	Emne
18/04	København	Grafiske brugergrænseflader Multimedia
3/06	Odense	Kommunikation og netværker
4/06	Odense	Systemudvikling og CASE
20/06 †	København	Standardisering

De med † markerede møder er eftermiddagsmøder, som er gratis for medlemmer.

Tid, sted og program for hver enkelt møde vil blive annonceret i DKUUG-Nyt forud for mødets afholdelse.