DKUUG-Nyt

Nr. 71 — juli 1994

Nestede databaser

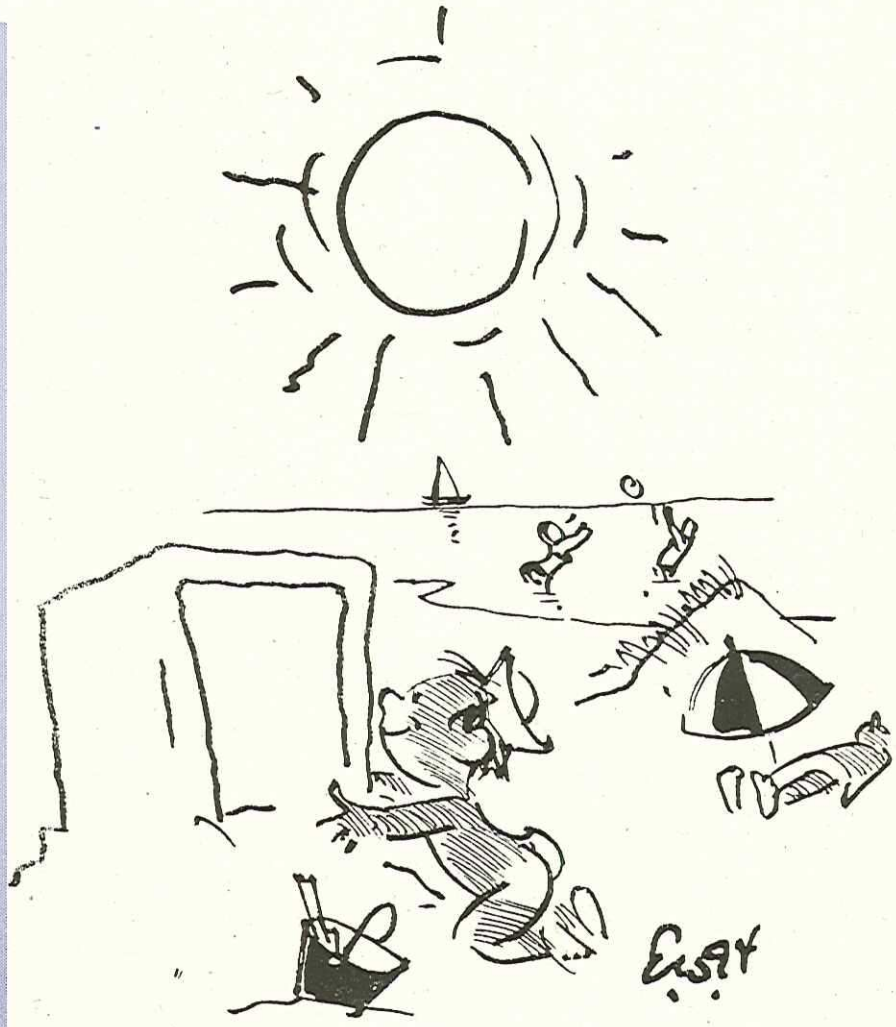
Traditionelle relationsdatabaser har en række begrænsninger, som man nu forsøger at komme ud over med nestede relationsdatabaser.

Fred mellem USL og Berkeley

De gensidige sagsanlæg mellem USL og Berkeley-universitetet er nu blevet trukket tilbage.

Sommer

Ja, og så er det helt uventet gået hen og er blevet sommer!



Indhold

Sommer...

Vi husker det alle, gyseren fra folkeskolen: stilen om sommerferien. Ikke nok med at man selv skulle forsøge at finde noget fornuftigt at skrive om de få ikke-kompromiterende ting man havde lavet i den alt for korte sommer, næh det værste var næsten, at læreren bagefter afstraffede klassen kollektivt ved at tvangsindlægge os til en nærmest uendelig række oplæsninger af diverse dydsmønstres selvforherligende (og sikkert dybt løgnagtige) pseudo-litterære feriebeskrivelser.

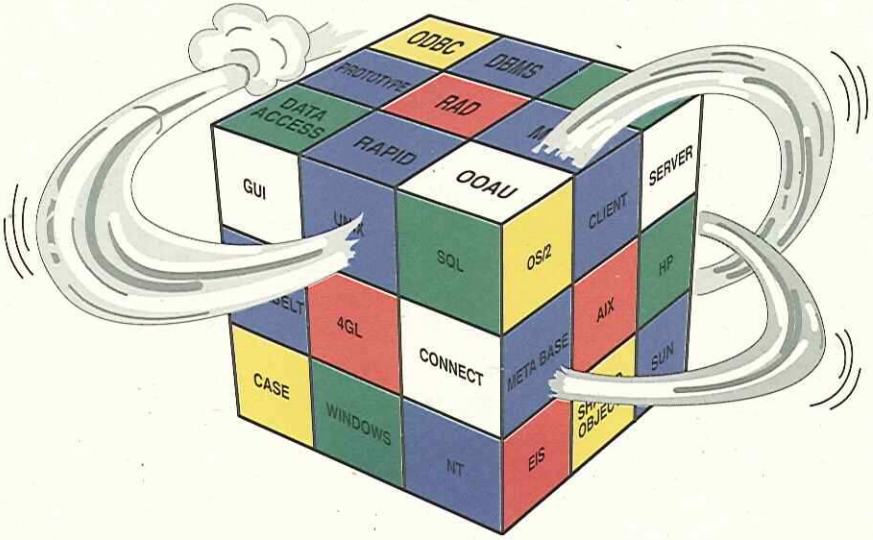
Hvorfor nu bringe disse smertefulde minder (dvs. minderne om familiefaderens reaktion ved konfrontationen med lærerens mening om sommerferiestilen) op? Tjah, hvis læreren i sin tid har følt, at det var nærmest umuligt at hive et par sider ud af en flok elever, så skulle han bare have prøvet at gøre noget tilsvarende med EDB-folk i sommervarmen. Mærkeligt nok vil folk meget hellere ligge ude i haven med en god manual end skrive artikler — og lige meget hvad reklamerne så ellers viser, så kan det under ingen omstændigheder anbefales at slæbe en bærbar computer med på stranden — ikke nok med at man får ondt i hovedet af reflekserne i skærmen, sand i diskettedrevet, etc. men endnu værre bliver man straks kategoriseret som en total nørd af alle de andre strandgæster.

Så summa summarum, dette er et lille tyndt (såkaldt handy) nummer af DKUUG-Nyt, læs det hurtigt igennem og drøn så ud til en af vore EU-godkendte strande og hop i bølgen blå.

GOD SOMMER

❑

SAS Institute lancerer applikationsudvikling under UNIX



Hurtig, effektiv og fleksibel applikationsudvikling er naturlige krav til udviklerne i mange virksomheder. SAS© Systemet opfylder i dag disse krav under Windows, NT, OS/2, UNIX, VMS, CMS og MVS.

Med den nye version af SAS Systemet under UNIX sættes der helt nye standarder for applikationsudvikling. SAS Institute introducerer således RAD (Rapid Application Development) i UNIX verdenen.

Objektorienterede teknikker, client/server, portabilitet, 4GL, GUI, DBMS access etc., er fuldt integrerede dele af SAS Systemets nye udviklingsmiljø.

Har du lyst til at høre mere om Rapid Application Development i client/server miljøer, så kontakt os på telefon 33 12 42 33 og få mere information.

SAS®

Den hurtigste vej fra data til information

SAS Institute A/S, Købmagergade 9A, DK-1150 København K, Tel.: 33 12 42 33

Den klassiske relations database trænger til en ansigtsløftning!

Bjørn Johannesen
Unidata (Scandinavia) Ltd

Relationsdatabasen er uden diskussion den foretrukne model indenfor administrativ databehandling.

Fordelene er mange og funktionaliteten er rig: Fleksibilitet, SQL, integritetskontrol, sikkerhed, triggers, stored procedures, online backup osv, osv. Desuden er de moderne relationsdatabasesystemer portable, og nogle af dem er også åbne, med mulighed for at udnytte den øvrige information i virksomheden.

På trods af disse fordele, samt mange andre, jeg ikke har nævnt, er der ikke alle, der kan anvende relationsdatabaser.

Hvorfor ikke?

Fordi relationsdatabasen, som den oprindeligt blev beskrevet af C.D. Codd, ikke lever op til dagens krav i et online transaktions-miljø.

Jeg vil her beskrive, nogle af de ulemper, jeg personligt har oplevet i forbindelse med de traditionelle relationsdatabasesystemer. Derefter vil jeg komme ind på de seneste tanker indenfor relationsdatabaseteorien og beskrive hvorledes disse er implementeret i praksis.

Den fysiske implementering afspejler ikke den logiske data model

Lad os se på en datamodel, der logisk set ser ud som figur 1 på modsatte side.

Der er et mange-til-mange forhold mellem ordre og varer. Vi er derfor tvunget til at indføre et nyt objekt, for at kunne afspejle dette i den tradionelle relationsdatabase med dens flade tabeller.

Modellen kommer derfor til at indeholde et nyt objekt, ORDRELINIE.

Fysisk kommer databasen til at bestå af 4 tabeller, selv om man logisk opfatter en kundeordre som en logisk enhed, svarende til det fysiske dokument (kundeordre).

> ***"Den moderne Relations Database overholder ikke Første Normal Form"***

For at kunne arbejde med en kundeordre, kræves tilgang til 4 tabeller med joins mellem disse.

Manglende mulighed for flerværdi-felter

Lad os udvide datamodellen.

Kunden har flere telefonnr. Dette er ikke noget problem. Vi opretter blot en ny tabel, KUNDETLF. Eller vi afsætter plads til f.eks. maksimalt tre telefonnr. Dette kan spare en tabel, men det går ud over fleksibiliteten og medfører også spildplads. Muligheden for at indeksere ved denne implementering er heller ikke til stede.

Performance

Implementeringen i flade tabeller giver redundans og de mange tabeller kræver i sagens natur mange opslag. Selv om man i nogle implementeringer kan styre den fysiske placering af data, og dermed reducere disk I/O, ændrer det ikke noget ved SQL-koden.

SQL

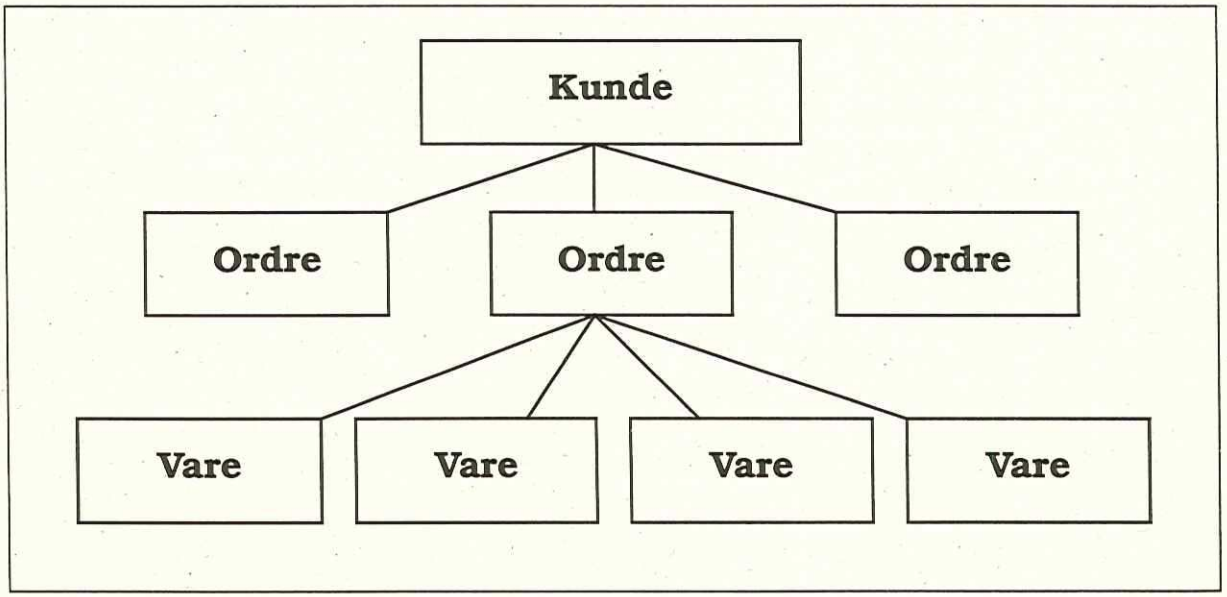
SQL-koden til at give den komplette kundeordre med kundens navn, adresse, telefonnr., ordrehoved, ordrelinier og varetekst skal jeg ikke trætte læseren med. Prøv heller selv at skrive SQL-koden.

"Variabel record er ikke længer tabu"

Selv om dette eksempel er simpelt, kræver det alligevel læsning i 5 tabeller og joins mellem disse.

Den nestede relationsmodel

E.F. Codd har brugt tiden siden sin lancering af relationsdatabase-teorien til at bevæge sig mod en mere



Figur 1. Eksempelmodellen

Customer Table

Customer #	Customer Name
AA2340987	Zedco, Inc.
GV1203948	Alphabravo
MT1238979	Trisoar

Order Customer

Order #	Customer #
93-1123	AA2340987
93-1154	AA2340987
93-2321	GV1203948
93-2342	MT1238979
93-4596	MT1238979

Order Table

Order #	Part #	Qty.
93-1123	063364	39
93-1123	087905	32
93-1154	037617	81
93-1154	053135	36
93-2321	006776	72
93-2321	055622	81
93-2321	067587	93
93-2321	067479	29
93-2342	036893	52
93-2342	065625	28
93-2342	090634	33
93-4596	005449	33

Figur 2. Modellen implementeret i en traditionel relationsdatabase

praktisk, logisk og objektorienteret model.

I sine "12 nye regler" beskriver E.F. Codd bl.a. relationsmodellen som værende ikke-Første Normalform model. Eller i daglig tale: Den nestede model.

Muligheden for at anvende flerværdi-felter og nestede tabeller giver helt nye muligheder.

I det nævnte eksempel kan vi nu implementere en tabel ORDRE, der indeholder både ORDREHOVED og et variabelt antal ORDRELINIE

Hver kunde kan desuden have et variabelt antal telefonnr. Vi har nu kunnet reducere antallet af tabeller fra de oprindelige 5 til blot 3: KUNDE. ORDRE og VARE.

"Den nestede model reducerer antal tabeller"

Enkelheden i implementeringen med færre tabeller, færre joins og enklere SQL er indlysende. Læg hertil den performancemæssige forbedring dette giver.

ANSI SQL 3 understøtter denne model med sine nest og unnest kommandoer.

Den nestede model er baseret på et ligeså solidt matematisk grundlag som den oprindelige relationsmodel og er en anerkendt model.

UniData Relations

Databasen, der ikke overholder første Normal Form

UniData er en af de yngre relationsdatabasesystemer på markedet, og det har derfor været naturligt at arkitekturen var baseret på de seneste relationsdatabaseteorier.

Nested Table				
Customer #	Customer Name	Order #	Part #	Qty.
AA2340987	Zedco, Inc.	93-1123	037617	81
			053135	36
		93-1154	063364	39
			087905	32
GV1203948	Alphabravo	93-2321	006776	72
			055622	81
			067479	29
			067587	93
MT1238979	Trisoar	93-2342	005449	33
			036893	52
			065625	28
		93-4596	090634	33

Figur 3. En nested relationsdatabase for samme model

Med den nestede relationsmodel reduceres antal tabeller, og den fysiske implementering afspejler den logiske datamodel — og ikke den traditionelle datamodel, der bygger på første normalform med overflødige objekter.

Der er ikke tale om at man mister de traditionelle fordele. Der er stadig mulighed for at anvende SQL, og der er selvfølgelig både nest og unnest kommandoer.

På samme måde som i andre relationsdatabasesystemer, kan vilkårlige felter indekseres — også flerværdifelter.

Erfaringerne har vist at performanceforbedringerne er store. Den største UniData implementering har 1.500 samtidige brugere på en, central database!

Objektorienteret database?

Objektorienteret systemudvikling, som er meget oppe i tiden, kræver en objektorienteret database. Jeg vil ikke påstå at UniData er en objektorienteret database. Dette ville kræve en definition af begrebet, hvilket jeg nødigt vil indlade mig på.

Men UniData er et skridt i den rigtige retning. Foruden muligheden for at lagre store datamængder (images, voice etc), som mange relationsdatabaser kan, findes der en række særdeles brugbare funktioner, f.eks.:

Virtuelle felter, der er funktion af andre data, kan defineres i data dictionary. Det virtuelle felt anvendes af SQL som ethvert andet felt.

En eksempel på dette kunne være ordresum i ordretabellen, beregnet ud

fra summen af de enkelte beløb i ordrelinierne.

Et andet eksempel kan være varetekst, der hentes i varetabellen med varenr. fra ordretabellen.

Resultatet af en nestet model med virtuelle felter er, at både programmører og brugere ser databasen på samme måde som virkelighedens objekter.

Det er muligt at generere en kundeordre med al relevant informationer med kun et SELECT-statement uden en eneste join !

Epilog

Den nestede model giver mulighed for at implementere komplekse datastrukturer, der afspejler virkeligheden.

Men det ændrer ikke ved det forhold, at data skal analyseres og at databasen skal designes på en fornuftig måde.

Også i den nestede model kan man lave et dårligt design, men databasedesigneren har nye muligheder og er sluppet af med nogle af tidligere tiders begrænsninger.

❑

Forbytning

Selv i et så omhyggeligt korrekturlæst blad som DKUUG-Nyt kan der faktisk snige sig en fejl ind. Især når redaktøren er af den gamle tekst-orienterede skole: "hvis et billede kan sige lige så meget som 1000 ord og jeg bliver betalt per side, så skal der ikke noget stort geni til at regne ud hvad jeg bringer!", er billederne tit det sidste, der bliver sat ind i bladet.

Lad det være ufortalt hvordan fejlen er sket, men i sidste nummer af DKUUG-Nyt bragte vi en præsentation af DKnets medarbejdere, hvor billederne af Stig Jacobsen og Frank Neergaard var blevet byttet om. Det forsøger vi hermed at råde bod på ved at bringe de to herrers protrætter igen, denne gang med de rigtige tekster under billederne:

❑

Stig Jacobsen

Frank Neergaard

Ny medarbejder hos DKnet

Frank Neergaard
DKnet

Med henblik på at styrke den salgsmæssige side i firmaet, har vi pr. 1. juni 1994 udvidet medarbejderstaben hos DKnet med Anne Dorthe Larsen. Anne Dorthe vil således aflaste Frank Neergaard med den daglige kontakt til kunder og emner og ligeledes fungere som sparringspartner ved udarbejdelse af diverse marketingstiltag m.v.

Anne Dorthe er uddannet som Cand. Merc. fra Handelshøjskolen i Århus, hvor hun dimiterede i 1987. Derefter flyttede hun til København for at starte som trainee i det tidligere Sparekassen SDS, nu Unibank. Som følge af fusionen og den deraf kaotiske jobsituation i banken valgte hun i 1990 at forlade Unibank til fordel for Multiinform A/S, som er autoriseret IBM-forhandler. Her arbejdede hun som salgskonsulent i løsningscenteret med fokus på mindre og mellemstore virksomheder. Senest har hun arbejdet som produktkonsulent i UNIWARE danmark a/s, hvor arbejdsområdet har været salg og markedsføring af kontorsystemet Uniplex og Uniplex onGO.

Vi ønsker selvfølgelig Anne Dorte Larsen velkommen i vores (verdens bedste!) firma, DKnet.

❑

USL og UC Berkeley når til enighed

Striden mellem USL og UC Berkeley er nu bilagt. Vi bringer her pressemeddelelsen fra marts-nummeret af Usenixs blad ";login"

UNIX System Laboratories, Inc. and the University of California, Berkeley have announced they have reached an agreement resolving their disputes. The settlement clears the way for the University to release a new, unencumbered version of the Berkeley 4.4 BSD operating system software, to be called 4.4 BSD-Lite.

Ray Noorda, Chairman of Novell, Inc., which recently acquired USL, called the settlement an "excellent example of what can be accomplished by cooperation between the business and academic communities." Mr. Noorda stated that "the settlement permits the University to accomplish its goals but preserves USL's legitimate interest in protecting its intellectual property."

David Hodges, Dean of the College of Engineering at the University of California, Berkeley, said that the settlement "once again allows the University to resume its leading rôle of providing computer software technology transfer to industry. By providing wide distribution of 4.4 BSD-Lite with minimal restrictions om its use, the University will continue to be the focal point for both software research in and commercial development of truly open systems."

The University of California was one of the earliest licensees of the UNIX operating system software, originally developed at AT&T's Bell Laboratories. In the 1980s. Berkeley's Computer Systems Research Group issued a series of "Berkeley Software Distributions" containing modifications to the UNIX software. However, because of licensing restrictions, public access to the source code for many of those modifications has been limited to firms holding licenses from USL, which acquired the rights to the UNIX system from AT&T.

> ***"The settlement clears the way for Berkeley to release a new version of 4.4 BSD"***

In July 1991, the University issued the "Second Networking Release" also known as Net2, which was intended to make available to the public those portions of the Berkeley Software Distributions which were not sub-

ject to license restrictions. However, USL brought a lawsuit against the University, claiming that portions of the release contained restricted material. The University denied USL´s claims. It also brought a separate action against USL alleging that USL had violated the terms of its Berkeley Software Distribution, also known as BSD, license agreement by failing to give the University credit for certain material in the UNIX release.

Over the past several months, attorneys and computer scientists representing the University and USL have worked together in an effort to reach a compromise on their disputes. The result of these efforts will be a new, unencumbered version of the latest Berkeley Software Distribution called 4.4 BSD-Lite which will retain virtually all of the functionality of the Second Networking Release along with a number of enhancements from the University´s latest 4.4 BSD release.

The settlement restricts further use and distribution of certain files in the Second Networking Release and requires that certain files in 4.4 BSD-Lite include a USL copyright notice. In addition to providing several enhancements, the new 4.4 BSD-Lite Release will replace most og the restricted files and incorporate all of the agreed-upon modifications and notices. Thus, 4.4 BSD-Lite will not require a license from nor payment of royalties to USL. The University strongly recommends that 4.4 BSD-Lite be substituted for Net2.

"4.4 BSD-Lite will retain virtually all of the functionality of the Net2 Release along with enhancements"

Although it has denied the University´s claims, USL has also agreed to affix the University´s copyright notice to certain files distributed with future releases of the UNIX system and to give credit to the University for material derived from BSD releases which have been included in the UNIX system.

"USL has also agreed to affix the University´s copyright notice to certain files in its UNIX distribution"

Copies of the source code for 4.4 BSD-Lite may be obtained from the University at nominal cost. Source code copies and further information on 4.4 BSD-Lite and the restrictions on Net2 may be obtained from the Computer Systems Research Group or from USL´s licensing offices.

❑

"Vis mig din source, og jeg skal vise dig dit rod..."

Jeg modtog fornylig et brev, addressen var skrevet med en dannet og pertentlig hånd og indeni lå et ark af det fineste håndlavede bøttepapir penge kan skaffe. I et dannet og diplomatisk sprog lod afsenderen mig vide at der fordredes visse ydelser fra min side og anmodede mig høfligt om at koncentrere mine evner og energier mod opgaven så hurtigt det ville forenligt med mine andre gøremål.

(Den Gl. Redacteur har forlangt at jeg ikke ustandselig rakker ned på hans dynamiske management, så jeg har på enkelte steder taget visse friheder i min beskrivelse af de faktiske hændelser. Faktisk var der tale om et billigt postkort, for hvilket jeg måtte betale strafporto, min kone rødmede beskæmmet da hun læste de gloser, der blev brugt for at beskrive afsenderens følelser og planer og min sagfører, som nu er i besidelse af bemeldte postkort, mener at det åbner "interessante juridiske muligheder".)

Til sagen

Nå til sagen. Hvis man har nogle lokale programmer, har man med garanti også problemer med at holde styr på sourcen til dem. Denne gang vil jeg derfor holde for, som de siger hinsidan, om cvs.

Cvs er et program der bruges til at lave rcs(1) om til et brugbart kilde-kontrol-system. Dvs, man skal have kompileret både rcs og cvs før man kan bruge det. Skaf den nyeste version af begge, selvom dit system evt. har rcs allerede. Nogle af de versioner der sendes med ud, er så archaiske at man undres over seriøsiteten af dettte (dette var et vink til HP, for de der ikke vidste det). Begge programmer kan findes på enhver nogenlunde velassorteret GNU-ftp-site. Compilering er lige ud af pakken, så det gider vi ikke dvæle ved, læs den medfølgende dokumentation om den slags.

Kildekontrolsystem?

Hvad er et kildekontrol system vil nogen nu uvægerligt spørge? Kort sagt et det et versions-baseret arkiv af filer. Man lægger filer ind og kan hente dem ud igen. Fidusen er at man kan også bede om at få den version der

kørte før fredag d. 13 september, eller bede om at se forskellen mellem de to versioner der hed <1.3> og <1.4>. Men kan give en version et symbolsk navn (FINAL) og man kan skrive en kommentar som bliver gemt sammen med de ændringer man laver.

Nu er der mange der siger "Ha, den slags behøver jeg ikke!" og det har de ret i. Det er en kendt sag at EDB bliv· ·brugt forbavsende lidt til at gøre livet nemmere for programmører og andre edb-folk. Jeg tror selv det er konservatisme der er skylden, men det kan man jo ikke sige til så "progressive" folk. Spørg en budget-triller om han har brug for sit spread-sheet. Han **kan** godt undvære det, men han vil satans nødigt.

Fordelen ved at bruge f.eks. cvs til at styre sine sourcefiler er, at der et veldefineret interface til arkivet over disse filer, og at det faktisk giver grundlæggende sporbarhed!

(For de der endnu ikke har opdaget det, består ISO900[0123] kun af to krav: dokumentation og sporbarhed. Af disse er sporbarheden det vigtigste fordi man kan udlede dokumentationen fra det, men ikke omvendt. Hvis man har sporbarhed betyder det at man kan dokumentere den række af ændringer der tog et produkt fra et niveau til et andet medsamt hvem der gjorde det og hvorfor.)

Man får derudover også muligheden for at have flere afgreninger af en source i samme træ i stedet for separate kopier. Således kan man f.eks have en "gren" til Solaris, en til AIX og en til HPUX. Man kan diff'e flere versioner, og man sparer diskplads fordi kun ændringerne bliver gemt.

Så tag lige og få fingeren ud og få ryddet op i alle de filer du har til at ligge i dit hjemme katalog, nu hvor sommeren aligevel ser ud til at drukne i regn.

Praksis

Cvs har brug for et subdirectory hvor alle de sources man checker ind kan opbevares. Udpeg et afsides hjørne af en stor disk. Sæt environment variablen CVSROOT til at pege på dette katalog.

> ***"EDB bliver brugt forbavsende lidt til at gøre livet nemmere for programmører og andre edb-folk"***

Jeg siger kun dette en gang: rør aldrig ved de filer der ligger under dette katalog med andet end cvs. Nu er du advaret! Der er ikke noget specielt ved disse filer, det er bare almindelige rcsfile(5)'er, som man kan læse med more(1) hvis man har brug for at checke et eller andet hurtigt, men CVS bliver utroligt forvirret hvis man laver om på noget uden at fortælle om det.

Nu skal der køres et script der laver nogle administrative filer i CVSROOT, dette script hedder `cvsinit` og bliver ikke installeret normalt, så det skal findes i source-træet for

cvs.

Nu er vi ellers klar. Lad os sige at vi har et sourcen til et program liggende et sted, cd ind i kataloget og gør følgende:

```
cvs import local/src/futtog LOKAL
V1_1
```

der dukker en editor op, skriv en forklarende meddelelse i den, "Første import af lego-kontroller" eller lignende. Herefter er sourcen checket ind. Det path man giver til "`cvs import`" er det sted i træet hvor man vil have indholdet af "`pwd`" lagt ind. Den kopi der blev importeret, dvs "`pwd`" kan nu slettes. (når man har taget backup-kopi!)

Hvis man nu skal bruge sin source igen siger man

```
(cd /scratch)

cvs co local/src/futtog

(cd local/src/futtog)

(make all install)
```

og det hele havner i "`./local/src/futtog`". Hvis man retter noget i sourcen skal man checke rettelsen ind. Stil dig i det øverste katalog hvor noget er ændret og:

```
(vi damp.c)

cvs commit
```

Editoren dukker op igen for at få at vide hvad du foretog dig. Hvis du lave en ny fil skal du fortælle cvs om det:

```
(vi diesel.c)

cvs add diesel.c
```

så bliver den lagt ind når du laver en commit. Tilsvarende når du har fjernet en fil skal du fortælle cvs om det:

```
cvs remove sporvogn.c
```

Tillykke du er nu beviseligt cvs-bruger. Her kommer så en masse løse ender og fif:

Hvis du bare taster "`cvs`" får du en kort oversigt over cvs's subkommandoer. Alle subkommandoerne tager argumentet "`-H`", som betyder hjælp, så hvis du ikke kan huske hvordan man bruger"`cvs diff`" siger du "`cvs diff -H`".

Bemærke at cvs har en liste af black-listede navne, som den ikke vil importere: `*.o`, `core` osv. Man kan sætte dette ud af kraft med option `-I`, læs manualen.

Det er muligt at lave et kælenavn for et subkatalog ved at lave en linie i `modules`-filen.

Denne og alle andre cvs administrative filer er under cvs kontrol så du får fingre i den ved at checke den ud med cvs selv:

```
cvs co CVSROOT
```

når du har rettet dem, skal de selvfølgelig checkes ind igen. Der er forskellige andre muligheder med de filer der ligger under CVSROOT. Man kan få ting til at ske i forbindelse med en commit, f.eks. maile en besked til en eller anden, eller kør et program som gør et eller andet praktisk.

Der er ikke noget i vejen for at checke hele gcc, X11 release 6, cvs selv og alt det andet software man bruger ind under cvs, jeg vil faktisk anbefale at man gør det, selvom det kræver en del diskplads.

Når en ny version af f.eks. gcc kommer ud, importerer man den bare oveni den forrige, cvs laver alt det benarbejde den kan. Hvis der er nogen konflikter må man rede dem ud i hånden

Da cvs er baseret på rcs, kan man bruge alle Foo stregene til at få lagt identifikation ind i sine filer. Prøv at lege lidt med det, indsæt f.eks:

```
/* $Id$ */
```

i sourcefilerne. Kig i co(1) for mere information.

Hvis man skal lave noget farligt, kan man lave sig et lille CVSROOT, ændre sin CVSROOT environmentvariabel, og lave et skalaforsøg først.

Det kan være en meget god ide at lade $CVSROOT pege på et symbolsk link til det faktiske katalog, specielt hvis man risikerer at skulle flytte det.

Der er ikke meget mere end rudimentær support for sikkerhed i cvs, men der kan laves meget takket være de scripts der ligger i $CVSROOT/CVSROOT.

Hjemmefronten

Jeg kan ikke dy mig for lige at berette lidt fra hjemmefronten:

Jeg har fået en lille.

[hva'ba? ... marts, april, maj, juni?? - red]

En velskabt lille fyr på 25 centimeter, og omkring 3 pund. Det er en "handbook 486" fra Gateway2000 computers, dvs 8Mb ram, 130Mb disk, og en 486DX2/40. Mere end rigeligt til at køre en UNIX på. Faktisk sidder jeg pt. i et af etatens regional-tog og skriver dette i vi(1). Det er en rar følelse: Pluselig har man alle sine gamle venner ved hånden når man rejser: awk, sed, tcl, bc og gcc.

Indeni de fleste bærbare sidder Intels 'SL' kredse, der er en familie af x86 cpu'er med støttekredse. De er lavet i CMOS, kan køre med variabel clock-frekvens og har indbygget mulighed for power-save funktioner. Faktisk kan man med 2 chips lave det meste af et motherboard. Lægger man dertil en PCMCIA, en VGA og muligvis en floppy controller, er man færdig med at bygge en PC.

Power-save er mange ting, der er de gammelkendte tricks: sluk for lyset, sluk for hard-disken osv. Med CMOS-kredse er

strømforbruget proportionalt med clock-frekvensen, så den kan man også sænke. Det mest drastiske er "suspend", hvor alle registre gemmes, og CPU'en går i stå. Det er temmeligt smart med UNIX: man kan "slukke" for sin PC, og senere "tænde" og fortsætte hvor man kom til. For det første skal man ikke vente på en reboot, og for det andet sparer man en **masse** strøm ved ikke at skulle lave en fsck. Det virker på et niveau over Protected mode, så selvom UNIX tror den har kontrol med hele maskinen, er der faktisk nogen der har endnu mere kontrol. (lidt ligesom politiet og pressen).

"Det er også ret rart at kunne sidde under møder og kode"

På kontoret kobler jeg mig på min stationære UNIX-maskine med RS-232 porten og SLIP (115200 bps), hjemme bruger jeg et modem. Jeg har lavet nogle strategiske shell-scripts, således at jeg f.eks. om morgenen, medens jeg vågner under bruseren og fortærer min kones hjemmebag, downloader `$MAIL` fra arbejdet. Så kan jeg i ro og mag læse og svare på mails på vej i toget. Når jeg ankommer på jobbet kobler jeg op med SLIP igen og kører `runq`, således at smail afleverer alle de mails jeg har sendt. Jeg overvejer at udvide dette med at downloade news på arbejdet og læse det på vejen hjem igen.

Det er også ret rart at kunne sidde under møder og kode, jeg har gcc og g++ lige indenfor rækkevidde, så når den abstraherende direktør går igang med noget alt for strategisk, kan man jo passende finde et par fejl i sin kode med `"cc -Wall"`. Eller bare det at kunne bruge `"cal 1997"` medens de andre stadig bladrer frem og tilbage i deres underlige ringbinds-kalendrer for at finde ud af at vi har en deadline på en mandag...

Jeg bruger normalt ikke X11. På en skærm med 640x480 er det lidt overkill, man har mere fornøjelse ud af at have "virtuelle consoler", de fleste x86 UNIXoider har dem på en eller anden måde. Men man kan godt køre X11, hvis man har brug for det.

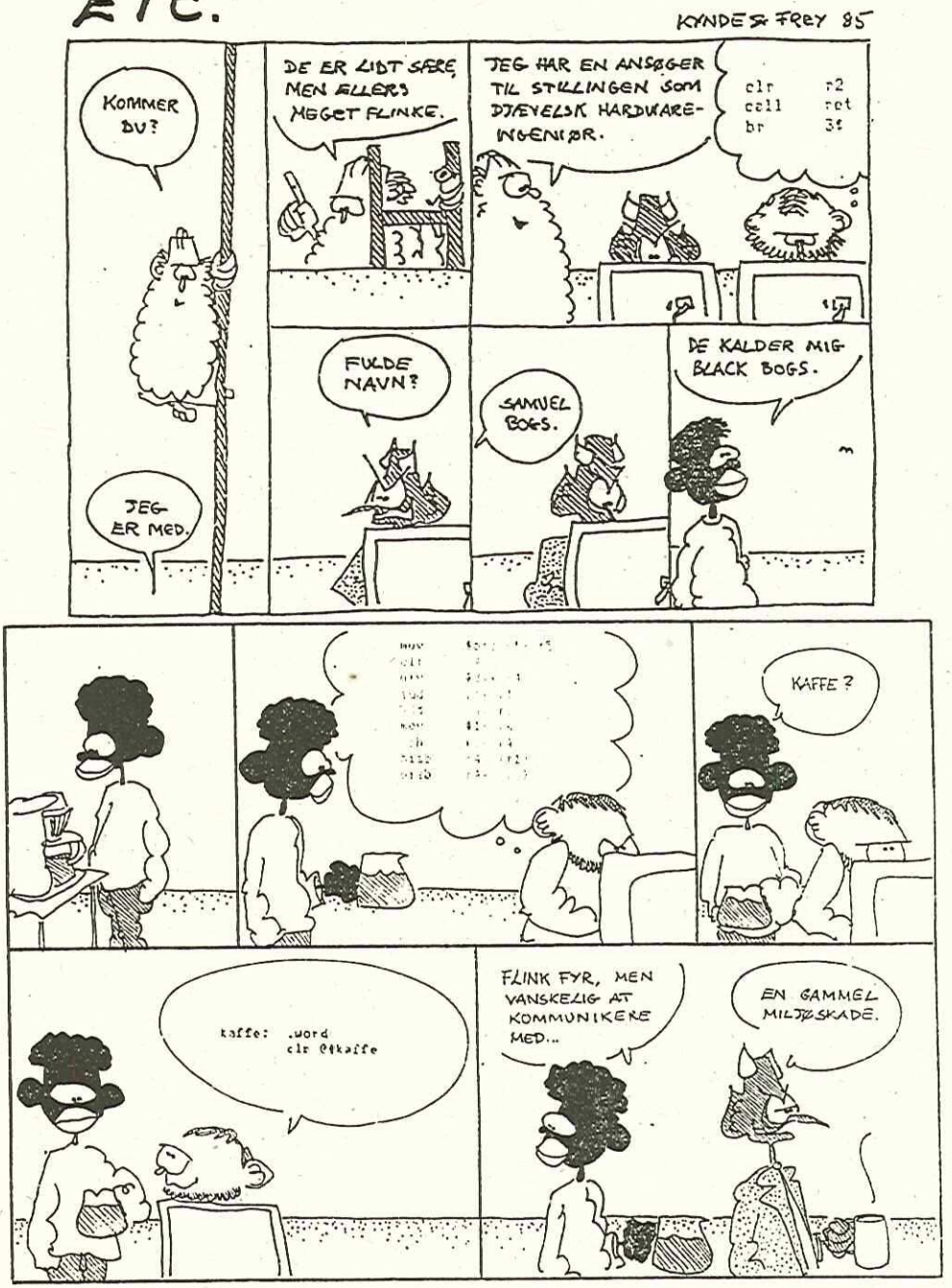
Efter nogen tid bemærker man dog at UNIX ikke netop er bygget til denne slags miljø. Der er ikke nogen faciliteter for energi-besparende foranstaltninger. Suspend virker f.eks. fint med UNIX, bortset fra at uret står stille mens maskinen er suspend'et. (Det er således klart for enhver hvorfor denne artikel kom for sent.)

[HA, skulle det vare en undskyldning ?? - red]

```
$Id: tekst,v 1.71.1.1 1994/06/21
21:01:23 jpp Exp $
```

❑

ETC.


KYNDE & FREY 85
KOMMER DU?
JEG ER MED.
DE ER LIDT SÆRE, MEN ELLERS MEGET FLINKE.
FULDE NAVN?
JEG HAR EN ANSØGER TIL STILLINGEN SOM DJÆVELSK HARDWARE-INGENIØR.
clr r2
call ret
br 3f
SAMUEL BOGS.
DE KALDER MIG BLACK BOGS.
KAFFE?
kaffe: .word
clr @tkaffe
FLINK FYR, MEN VANSKELIG AT KOMMUNIKERE MED...
EN GAMMEL MILJØSKADE.

Klubaften i København

Tirsdag den 30. august kl. 19:00 - 22:30
Unidata (Scandinavia) Ltd
International House
Center Boulevard
2300 København S

Den nestede Relations Database

Foredrag, debat og hands-on

E.F Codd og C.J Date har været foregangsmænd for den relationelle model, hvilket har dannet basis for de traditionelle RDBMS-implemeteringer. Disse bygger alle på 25 år gamle teorier.

Flade tabeller, hvor der ikke er mulighed for at lagre flerværdi-felter giver et stort antal tabeller og redundans. Dette gør det vanskeligt at overskue databasen, da denne ikke er en logisk afspejling af virkeligheden.

Ydeevnen lider også under den gamle RDBMS-model. Hvem kender f.eks. et flyselskab, der har baseret svartids-kritiske systemer på en relationsdatabase?

De gamle teorier lever heller ikke op til tidens krav om en objektorienteret modellering og lagring af data.

E.F. Codd har da også revideret den oprindelige relationsmodel med sine 12 nye regler for On-Line Analytical Processing, hvor han bl.a. beskriver Non-First Normal Form (NF2). C.J. Date har fulgt op på dette i sine bøger, hvor han også beskriver den nestede relationsmodel.

Hvordan dette er implementeret — uden at gå på kompromis med SQL, fleksibilitet og fortsat anvendelse af eksisterende applikationer — kan du se og prøve i praksis denne aften.

Arrangementet er gratis, men af hensyn til det praktiske, bedes du tilmelde dig til DKUUG-sekretariatet senest to dage inden arrangementet.

NB: Læg mærke til at denne klubaften afholdes i Bella Center, da der vil blive mulighed for selv at opleve den nestede relationsdatabase.

Vel Mødt!

UNIX-markedet 94/95

HUSK

UNIX-markedet:

Hvordan er det gået i det forløbne år — hvad vil der ske i det kommende?

Medlemsmøde d. 25.8.94

Scanticon i Snekkersten

Yderligere oplysninger hos DKUUGs sekretariat:

Tlf. 39 17 99 44
Fax 31 20 89 48

DKUUG-Nyt udgives af:
Dansk UNIX-system Bruger Gruppe

DKUUG, sekretariatet
Symbion
Fruebjergvej 3
2100 Kbh. Ø
Tlf. 39 17 99 44
Fax 31 20 89 48
Giro: 137-8600
Email: sek@dkuug.dk
Man - tors kl 9 -16.30
Fredag kl 9 - 15.30

DKnet
Tlf. 39 17 99 00
Fax 39 17 98 97

Redaktion

Søren Oskar Jensen (ansv.)

DKUUG-Nyt
C/O Søren O. Jensen
Vesterbrogade 65, 2.th.
1620 Kbh. V
Tlf. 31 22 84 43
Fax 31 22 84 43
Email: dkuugnyt@dkuug.dk

Deadline

Deadline for næste nummer, nr. 72, er fredag d. 22.7.94

❑

Medlemsmøder i 1994

25/8	UNIX-markedet	Scanticon
27-28/9	X-event	Odense
27/10	Drift, network managemnet	Symbion
24/11	Posix i praksis / Generalforsamling	Symbion