

Medlemsblad for  
Dansk UNIX-system Bruger Gruppe

# DKUUG-Nyt

Nummer 38, 1. marts 1991

## Indhold

Redaktionelt . . . . .	2
DKUUG Klubaften i København . . . . .	3
Flytning af medlemsmøder . . . . .	4
Den Tekniske Side . . . . .	5
AALBUG - Aalborg Unix bruger gruppe . . . . .	8
Source Code Control System . . . . .	10
Åbne Systemer — hvad er det? . . . . .	20
Why UNIX Will Win on the Desktop . . . . .	26
DKUUG - Meddelelser fra sekretariatet . . . . .	29
Oversigt over medlemsmøder i 1990-1991 . . . . .	32

## 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. 39 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. 39 er d. 22. marts  
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:

GNU — gratis software, lots of goodies

Foredragsholder:

Rene Seindal

Datalogisk Institut, Københavns Universitet

Tirsdag den 26. marts 1991

Kl. 19:00 – 22:30

Datalogisk Institut (DIKU)

Universitetsparken 1

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

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!

### **Flytning af medlemsmøder**

Medlemsmøderne i Odense om kommunikation og netværker samt om Systemudvikling og CASE er flyttet fra 29-30 maj til 3-4 juni. Møderne vil finde sted på H.C.Andersen Hotel.

Venlig hilsen  
Mogens Buhelt

## Den Tekniske Side

Af Poul-Henning Kamp

F. L. Schmidt

Flere og flere anskaffer sig Trailblazer modemmer. For at lette installationen af disse bringes her mit setup af en TB2500.

Det er et 'smart' setup paa en IBM RS/6000, som nok kræver en lille forklaring:

Ved at definere en masse 'Devices' opnår man at kunne vælge en hvilken som helst modulation, nogenlunde enkelt i 'Systems'

I Devices bruger jeg en udokumenteret feature ved HDB, nemlig flere Dialers pr linie, det gør at også Devices bliver nogenlunde over-skuelig.

I Dialers kan man så sætte de relevante 'sub-dialers'. Afhængig af hvad ens modem har som default konfiguration, min er til Dial-out only.

Læg mærke til S183=0 som default, men =1 ved V.32, dette er en option som P&T har fået slettet fra manualen, den vedrører brug af V.25bis svartone ved PEP mode, mod DKUUGs nuværende TB-modems skal man køre S183=0, mens man ikke kan køre V32 uden S183=1, V22 &c er vist ligeglade.

Håber at dette kan være en hjælp.

```
-----#####-----
# '/usr/lib/uucp/Systems'
#
# TB2500 Setup by phk@data.fl.s.dk 06feb91 -- BEGIN --
#
modem Any TB2500 Any 0
dkuug Any PEPCU Any DK_35375967 041 \r\c ogin:-\r\c-ogin: uufls rd: A1-Gua
dkuug Any S2400U Any DK_35375722 "" \r\c ogin:-\r\c-ogin: uufls rd: A1-Gua

hp-bbs Any V32T Any USA_503-750-3277
hp-bbs Any S2400T Any USA_503-750-4448
hp-bbs Any S1200BT Any USA_503-750-4448
nbs-time Any S300B Any USA_303-494-4774
#
# TB2500 Setup by phk@data.fl.s.dk 06feb91 -- END --
#
```



-----#####-----

# '/usr/lib/uucp/Devices'

#

# TB2500 Setup by phk@data.fl.s.dk 06feb91 -- BEGIN --

#

# For direct connection to the modem:

TB2500 tty1 - 19200 direct

S300	tty1 - 19200	S300	\D	TB	\D				
S300M	tty1 - 19200	S300	\D	MNP	\D	TB	\D		
S300T	tty1 - 19200	S300	\D	TRYMNP	\D	TB	\D		
S300B	tty1 - 19200	S300	\D	BELL	\D	TB	\D		
S300BM	tty1 - 19200	S300	\D	BELL	\D	MNP	\D	TB	\D
S300BT	tty1 - 19200	S300	\D	BELL	\D	TRYMNP	\D	TB	\D

S1200	tty1 - 19200	S1200	\D	TB	\D				
S1200M	tty1 - 19200	S1200	\D	MNP	\D	TB	\D		
S1200T	tty1 - 19200	S1200	\D	TRYMNP	\D	TB	\D		
S1200B	tty1 - 19200	S1200	\D	BELL	\D	TB	\D		
S1200BM	tty1 - 19200	S1200	\D	BELL	\D	MNP	\D	TB	\D
S1200BT	tty1 - 19200	S1200	\D	BELL	\D	TRYMNP	\D	TB	\D

S2400	tty1 - 19200	S2400	\D	TB	\D				
S2400M	tty1 - 19200	S2400	\D	MNP	\D	TB	\D		
S2400T	tty1 - 19200	S2400	\D	TRYMNP	\D	TB	\D		
S2400U	tty1 - 19200	S2400	\D	UUCP	\D	TB	\D		
S2400MU	tty1 - 19200	S2400	\D	UUCP	\D	MNP	\D	TB	\D
S2400TU	tty1 - 19200	S2400	\D	UUCP	\D	TRYMNP	\D	TB	\D

V32	tty1 - 19200	V32	\D	TB	\D				
V32M	tty1 - 19200	V32	\D	MNP	\D	TB	\D		
V32T	tty1 - 19200	V32	\D	TRYMNP	\D	TB	\D		
V32U	tty1 - 19200	V32	\D	UUCP	\D	TB	\D		
V32MU	tty1 - 19200	V32	\D	UUCP	\D	MNP	\D	TB	\D
V32TU	tty1 - 19200	V32	\D	UUCP	\D	TRYMNP	\D	TB	\D

PEP	tty1 - 19200	PEP	\D	TB	\D				
PEPC	tty1 - 19200	PEP	\D	CMPR	\D	TB	\D		
PEPCU	tty1 - 19200	PEP	\D	UUCP	\D	CMPR	\D	TB	\D

#

# TB2500 Setup by phk@data.fl.s.dk 06feb91 -- END --

#

-----#####-----

```
# '/usr/lib/uucp/Dialers'
#
# TB2500 Setup by phk@data.fls.dk 06feb91 -- BEGIN --
#
direct
TB      =W-,    "" ATDT\T\r\c CONNECT-\d-CONNECT
S300   =,-,    "" ATS50=1 OK
S1200  =,-,    "" ATS50=2 OK
S2400  =,-,    "" ATS50=3 OK
V32    =,-,    "" ATS93=20S94=3S50=6S183=1s7=255 OK
PEP    =,-,    "" ATS50=25S92=0 OK
BELL   =,-,    "" ATS90=0 OK
MNP    =,-,    "" ATS95=1 OK
TRYMNP =,-,    "" ATS95=2 OK
UUCP   =,-,    "" ATS111=30S58=0 OK
CMPR   =,-,    "" ATS110=1 OK
#
# TB2500 Setup by phk@data.fls.dk 06feb91 -- END --
#
```

-----#####-----

```
# TB2500 profile #1:
ATN?
E1 F1 MO Q0 T V1 W1 X3 YO &P0 &T4      Version HA2.00 LIB:DK
S00=001 S01=000 S02=043 S03=013 S04=010 S05=008 S06=006 S07:240 S08:001 S09=006
S10=007 S11=100 S12=050 S18=000 S25:001 S26=000 S38=000
S41:001 S45=000 S47:100 S48=000 S49=000
S50=000 S51:005 S52:002          S54:003 S55=000 S56=017 S57=019 S58=003 S59=000
S60=000 S61:025 S62=003 S63=001 S64=000 S65=000 S66:001 S67=000 S68=255 S69=000
S90=001 S91=000 S92=002 S93=008 S94:002 S95=000 S96=001
S100=000 S101=000 S104=000 S105=001
S110:000 S111:000 S112=001
S121=000 S130:003 S131:003
S150=000 S151=004 S152=001 S153=001 S154=000 S155=000
S160=010 S161=020 S162=002 S163=003 S164=007
S183:000 S255:001
OK
```

-----#####-----

## AALBUG - Aalborg Unix bruger gruppe

Emacs - Den programmerbare editor

Tirsdag den 19 marts 1991

Kl. 19:00 - 22:30

Aalborg Universitetscenter

Institut for elektroniske systemer

Fredrik Bajersvej 7, bygning D2

Lars Bo Nielsen fra Hugin vil fortælle om hvordan man får det fulde udbytte af EMACS editoren. EMACS bliver af mange betragtet som verdens bedste/største editor og der er stortset ikke det der ikke kan lade sig gøre inde fra EMACS. GNU EMACS er gratis software, derudover findes der et utal af EMACS cloner både til PC'ere og andre useriøse maskiner, stortset alle er gratis software.

Intentionen med AALBUG er at give UNIX brugere i det nordjyske mulighed for at mødes med ligesindede.

Der vil hvergang være et "aftenens emne", hvor en erfaren bruger eller programmør vil holde det indledende foredrag, 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. Så giver din UNIX kværn dig problemer eller har du lige fundet et nyt supersmart program, så trop op og måske kan du blive befriet for begge dele.

AALBUG vil have fast mødetid og sted nemlig den næstsidste tirsdag i hver måned kl 19:00 på ovenstående lokalitet. Døren til Instituttet er normalt låst udenfor normal arbejdstid, så kom rimeligt præcis. Der vil være kaffe og te adlibitum, mens øl og vand kan købes

Tidligere har der været afholdt følgende møder i Aalbug:

Tirsdag 20 november	Minix - en UNIX clone med kildetekst
Tirsdag 18 december	Pascal til C kompiler
Tirsdag 22 januar	brug af awk og sed
Tirsdag 19 februar	TeX - Verdens bedste Desktop Publishing system



Emnerne ligger ikke helt fast endnu for de efterfølgende møder, men foreløbig ser programmet således ud:

Tirsdag 19 marts Emacs - den programmerbare editor  
Tirsdag 23 april perl - hvad mon det er for noget ??  
Tirsdag 21 maj programmering under X-windows  
Tirsdag 18 juni gnu C compiler

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)  
alm. post: Peter L. Petersen  
Inst 8, AUC  
Fredrik Bajersvej 7C  
9220 Aalborg

---

Vel Mødt.

# Source Code Control System

Af *Kim Chr. Madsen*  
KIMCM CONSULT (UNIX Konsulent Firma)

## Introduktion

Source Code Control System, eller SCCS er et system designet til at vedligeholde flere versioner af samme dokument, programtekst eller andre tekstfiler under Unix.

SCCS-systemet er opbygget på traditionel Unix-facon, ved at lægge de enkelte delfunktioner ud i separate programmer. I de efterfølgende afsnit vil de enkelte programmer blive gennemgået, således at det vil være muligt at anvende disse systemer i langt de fleste tilfælde. Ønskes en mere gennemgribende og udtømmende forklaring henvises til systemets manualer og dokumentation om SCCS-systemet.

I resten af dette dokument vil der blive refereret til tekstfiler, som kan dække over både dokumenter, almindelige tekstfiler, kildetekster til programmer, eller kort sagt alt som ikke er binære filer.

Virkemåden hvorpå SCCS-systemet fungerer er at den kan gemme flere versioner af den samme tekstfil i den samme SCCS-fil. Måden det gemmer disse versioner på er ved at have en grundversion og derefter en række oplysninger om hvorledes senere versioner er fremkommet ud fra denne. Dette betyder at når der oprettes en ny version gemmes disse ikke i deres fulde udstrækning, men kun forskellene fra den tidligere version gemmes.

Gamle versioner kan hentes frem og der kan rettes i disse og derved dannes sidegrene (branches) på de enkelte versioner.

Normalt er det kun den person der har oprettet SCCS-filen der har tilladelse til at lave nye versioner af tekstfilen, men der kan løbende gives tilladelse til andre brugere og grupper til at oprette nye versioner, ligeledes kan disse rettigheder løbende fratages.

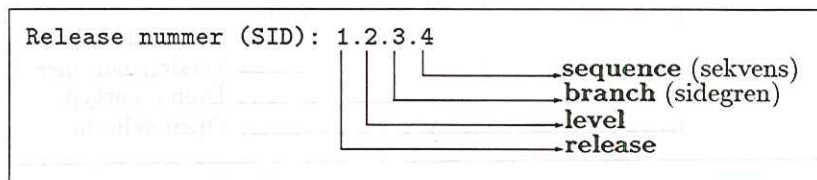
## Oprettelse af en SCCS-fil

Når man vil oprette en SCCS-fil, gøres dette ved at benytte kommandoen `admin` og specificere hvilken tekstfil der ønskes at bruge som grundlag for den første version. Når SCCS-filen er oprettet kan den oprindelige version slettes. Herefter kan den ønskede version hentes fra SCCS-filen enten der skal foretages ændringer af denne, eller man blot vil kigge på den.

### Options til `admin` ved oprettelse af SCCS-filer

`-inavn` Navnet på filen der indeholder den initielle version, hvorfra SCCS-filen skal dannes. Kan kun anvendes ved oprettelse af SCCS-filen.

`-rrelease` Release-nummeret på den initielle version. Hvis `-r` ikke anvendes bliver release-nummeret på den første version 1. Delta-nummeret på den initielle version vil altid være 1. (default release-nummer vil derfor blive 1.1). Kan kun benyttes hvis `-i` option også er anvendt.



`-fnummer` Højeste releasenummer der kan indeholdes i denne SCCS-fil (default er 9999).

`-ffnummer` Laveste positive releasenummer der kan hentes med `get`-kommandoen.

`-fqtekst` Brugerdefineret tekst der erstatter alle forekomster af makroen `%Q%`. !

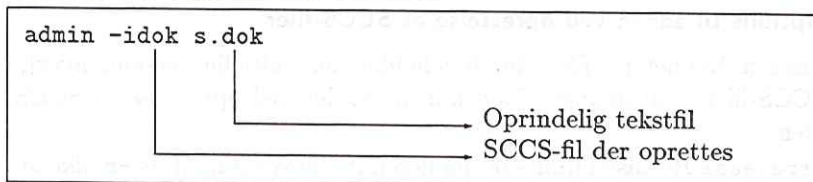
`-fttekst` Angiver typen af SCCS-filen, indsættes ved alle forekomster af makroen `%Y%`. !

`-anavn` Giver den navngivne bruger eller gruppe af brugere tilladelse til at danne nye versioner af tekstfilen i SCCS-filen. Denne option kan gentages flere gange i `admin`-kommandoen.

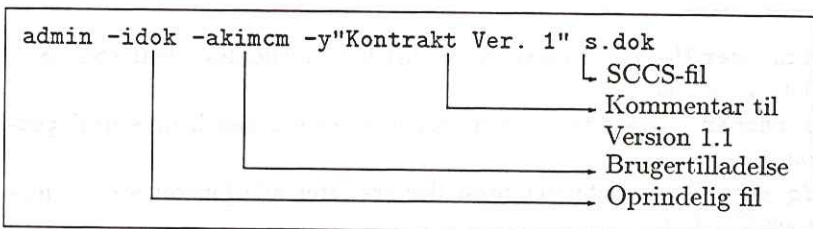
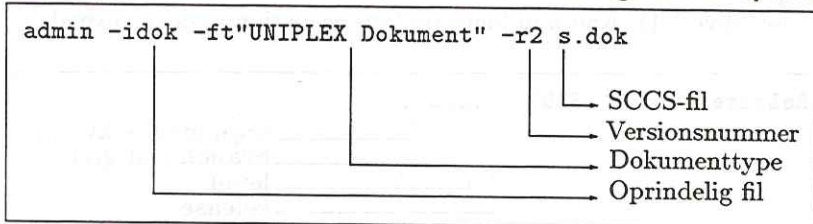
`-enavn` Fjerner den navngivne bruger eller gruppe af brugere fra listen over hvem der har lov til at lave ændringer i SCCS-filen. Denne option kan gentages flere gange i `admin`-kommandoen.

`-ytekst` Indsætter den angivne tekst som kommentar til den første version. Hvis `-y`-option ikke anvendes vil kommentaren til den første version være datoen og tidspunktet hvor den initiale version blev oprettet. Denne option kan kun benyttes når `-i`-option også benyttes.

### Eksempel 1. Oprettelse af SCCS-fil



### Eksempel 2 Oprettelse af SCCS-filer med forskellige admin options



### Udtrække en version fra en SCCS-fil

Når man ønsker at hente en version frem fra en SCCS-fil, sker dette vha. programmet `get`. Normalt vil `get` hente den version med det højeste versionsnummer, men andre versioner kan også hentes ved at

angive hvilket versionsnummer der skal hentes. Forskellige makroer ekspanderes med de aktuelle værdier der gælder når teksten hentes fra SCCS-filen (se afsnittet om Makroer). Hvis man ønsker at redigere i den fremkomne version, skal man gøre SCCS-systemet opmærksom herpå, se afsnittet Rettelser til SCCS-version nedenfor.

### Eksempel 3: Udtrækning af version fra SCCS-fil

```
get s.dok
```

Henter versionen med det højeste releasenummer fra SCCS-filen s.dok.

```
get -t s.dok
```

Henter senest redigeret version fra SCCS-filen

```
get -r1.4 s.dok
```

Henter version 1.4 fra SCCS-filen

### Rettelser til SCCS-version

Når der ønskes at lave rettelselser til en version af en tekstfil gemt i en SCCS-fil, sker dette ligeledes med programmet `get` dog skal man anvende option `-e` (edit/rediger) for at gøre SCCS-systemet opmærksom på at der redigeres i den hentede version, således at den nye version kan indarbejdes i SCCS-filen efter endt redigering. Når der redigeres i en version ekspanderes makroer ikke.

### Eksempel 4: Udtræk af version til redigering

```
get -e s.dok
```

Meddelelse om at  
der ønskes redigering  
SCCS-fil



### Opdatering af releasesnummer

Normalt vil SCCS-systemet kun opdatere levelnummeret når der laves nye versioner. Ønskes det at opdatere releasesnummeret gøres dette ved at kalde get-kommandoen og angive hvilken version man ønsker at benytte.

#### Eksempel 5: Opdatering fra release 1.7 til 2.1

```
$ get -e -r2 s.dok
1.7
new delta 2.1
612 lines
```

### Udskrift af version på terminal eller printer

I visse tilfælde er det ikke nødvendigt at gemme den fremkaldte version i en fil, men blot videregive den til terminalen eller til et andet Unix-program via pipe-mekanismen. Dette gøres ved at angive option `-p` (pass) til get-kommandoen.

#### Eksempel 6: Udskrift af version gennem programmet print

```
$ get -p -r1.4 s.dok _ print
1.4
578 lines
```

### Udskrift af historik på SCCS-fil

I visse tilfælde er det rart at kunne identificere de enkelte versioner i en SCCS-fil. Dette gøres ved at benytte `-l` option, hvilket giver de historiske oplysninger i en fil (hvis SCCS-filen hedder `s.strings.c` vil historik-filen hedde `l.strings.c`) eller ved at benytte `-lp` kommandoen der giver de samme oplysninger på standard output (terminalen).

#### Eksempel 7: Historik på SCCS-filen s.strings.c

```
$ get -lp s.strings.c
1.3
1.3 89/06/23 14:54:33 kimcm
Added minlength functionality into program
```

1.2 89/06/22 13:28:30 kimcm

Added SCCS Data Keywords

1.1 89/06/22 13:23:56 kimcm

UCB like strings program

79 lines

## Fortryd udtræk fra SCCS-fil

Hvis man har udtrukket en version til redigering fra SCCS-filen, vha. `get -e` kommandoen, og fortryder dette kan man sørge for at bringe dette forhold i orden igen ved hjælp af `unget`-kommandoen.

### Eksempel 8: Brug af `unget`-kommandoen

```
$ unget s.dok
```

1.3

Hvis man har flere udestående redigeringer fra samme SCCS-fil, er det nødvendigt at angive hvilken version man ønsker at fortryde.

### Eksempel 9: Brug af `unget` på specifik version

```
$ unget -r1.3 s.dok
```

1.3

## Indsættelse af rettede versioner i SCCS-filen

Efter at have frembragt en version fra SCCS-filen og tilrettet den efter behov, skal den nye version indpasses i SCCS-filen igen, dette gøres vha. kommandoen `delta`.

`delta`-kommandoen indfører den nye version i SCCS-filen, som tilrettelser til den eller de forgående versioner, og indsætter en kommentar til versionen, denne kommentar er normalt tidspunktet den nye version blev oprettet og af hvem, samt en brugerdefineret kommentar, som dog kan være tom.

**Eksempel 10: Brug af kommandoen delta**

```
$ delta s.tape.c
```

```
Type comments, terminated with EOF or a blank line:
```

```
Rettet mindre fejl i forbindelse med EOT (End-Of-Tape).
```

```
1.3
```

```
12 inserted
```

```
4 deleted
```

```
307 unchanged
```

Hvis der er flere udestående versioner fra den samme SCCS-fil er det nødvendigt at specificere hvilken version man ønsker at opdatere SCCS-filen med, dette gøres på følgende måde:

**Eksempel 11: Brug af kommandoen delta på specifik version**

```
$ delta -r1.3 s.tape.c
```

```
Type comments, terminated with EOF or a blank line:
```

```
Rettet mindre fejl i forbindelse med EOT (End-Of-Tape).
```

```
1.3
```

```
12 inserted
```

```
4 deleted
```

```
307 unchanged
```

**Fjernelse af en version fra SCCS-fil**

Hvis man ønsker at fjerne en specifik version fra SCCS-filen kan dette gøres vha. kommandoen rmdel. Følgende begrænsninger gælder dog:

- Kun den nyeste version i hovedgrenen eller nyeste version i en sidegren kan fjernes med kommandoen rmdel.
- Versioner der er udtrukket med redigering for øje, dvs. ved hjælp af get -e kommandoen kan ikke fjernes.

**Eksempel 12: Fjernelse af en version fra en SCCS-fil**

```
$ rmdel -r1.3 s.tape.c
```

## Låsning af releases fra SCCS-fil

Hvis man ønsker at gøre en release utilgængelig for redigering, men ikke til at kigge i, kan man låse alle versioner der tilhører denne release, dette gøres på følgende måde:

### Eksempel 13: Låsning af versioner fra angivet release

```
$ admin -f11 s.tape.c
```

Her låses alle versioner der tilhører release 1.

## Oplåsning af låste releases i SCCS-fil

Hvis man igen ønsker at kunne hente versioner fra en låst release frem kan dette gøres ved at låse op for denne release igen, dette gøres med kommandoen `admin`, på følgende måde:

### Eksempel 14: Oplåsning af låst release i SCCS-fil

```
$ admin -dl1 s.tape.c
```

Her låses igen op for redigering af versioner fra release 1.

Ønskes der en liste over forskelle mellem to versioner i samme SCCS-fil, kan en sådan genereres ved hjælp af programmet `sccsdiff`. Uddata fra dette program er på samme format som standard Unix-kommandoen `diff`, se manualen på denne for forklaring af uddata.

### Eksempel 15: List af forskelle mellem to versioner

```
$ sccsdiff -r1.1 -r1.2 s.strings.c
16a17,18
> static char *sccsid = "%Z%%M% %I% Date: %D% Time: %T%";
>
```

## Identifikation af SCCS-filer

Hvis en SCCS-fil indeholder tekststrengen "@(#)" bliver den efterfølgende tekst på linien udskrevet, når programmet `what` udføres. I afsnittet SCCS-makroer beskrives en række makroer der kan benyttes

i de enkelte versioner, som bliver ekspanderet af kommandoen `get`, når denne ikke benyttes til at udtrække en version til redigering.

### Eksempel 16: Identifikation af det oversatte program strings

```
$ what /usr/local/bin/strings
/usr/local/bin/strings:
strings.c 1.3 Date: 89/06/26 Time: 09:27:30
```

### SCCS-makroer

Når man redigerer en version, kan man indsætte forskellige makroer, som vil blive udskrevet når man anvender kommandoen `get` uden `-e` option. På næste side er der en liste af de mest brugte.



%I%	SCCS ID-Streng (SID)	%R%.%L%.%B%.%S%
%R%	Release-nummer	nnnn
%L%	Level-nummer	nnnn
%B%	Branch-nummer (gren)	nnnn
%S%	Sekvens-nummer	nnnn
%D%	Dato for delta	%Dy%/%Dm%/%Dd%
%Dy%	År - 1900 for delta	nn
%Dm%	Månednummer for delta	nn
%Dd%	Dag i måned for delta	nn
%T%	Tidspunkt for delta	%Th%:%Tm%:%Ts%
%Th%	Time for delta	nn
%Tm%	Minut for delta	nn
%Ts%	Sekund for delta	nn
%DS%	Delta sekvensnummer	nnnn
%C%	Kommentarer til delta	tekst
%UN%	Brugere der kan oprette deltaer	tekst
%Y%	Type af SCCS-fil	tekst
%Q%	Brugerdefineret makro	tekst
%M%	Modulnavn	tekst
%W%	En slags what tekst	%Z%.%M%.t%I%
%A%	En anden slags what tekst	%Z%.%Y% %M% %I% %Z%
%Z%	Indledning til what tekst	@(#)
%F%	Filnavn	tekst
%PN%	Stinavn	tekst

## Åbne Systemer — hvad er det?

Af *Steen Clausen*  
Dansk Data Elektronik A/S

Der har igennem de sidste par år været en stadig stigende anvendelse af udtrykket "åbent system" i mange forskellige sammenhænge. Da betegnelsen dækker over noget helt konkret, men af mange brugt meget løst og dækkende mange forskellige ting har vil jeg her forsøge at præcisere hvad der menes med udtrykket, og mere specifikt hvordan DDE i udviklingen af Supermax datamaten har benyttet principperne indeholdt i åben system-konceptet helt fra den tidlige begyndelse.

### Åbne Systemer — et forsøg på en definition

Et "åbent system" kan i datalogisk forstand defineres som et databehandlings system bestående af forskellige komponenter, der alle har det tilfælles, at de er baseret på standarder specificeret i enten en de jure eller de facto facon. Hvad menes der så med udtrykkene standard, de jure og de facto?

Det er vel sådan indenfor edb-området, som indenfor alle andre områder med en rivende udvikling, at standardisering er lidt af et mareridt. Alle taler om det, mange gør en masse ved det, standard-regler bliver vedtaget hele tiden, og alligevel er det ikke tilstrækkeligt. En **standard** er en metode, der anbefales til løsning af et bestemt problem. Indenfor edb-området findes der et utal af standarder, f.eks. inden for programmeringssprog og datakommunikation. Der findes mange internationale organisationer, som udarbejder standarder, blandt andre ANSI, CCITT, og ISO. Disse forkortelser betyder American National Standard Institute, Committe Consultif International Telephonique et Telegraphique, og International Standards Organisation.

De jure betyder et sæt normer (standarder) vedtaget i et internationalt anerkendt forum, medens de facto betyder faktisk eller reel. Det sidste udtryk benyttes som betegnelse for metoder, maskiner, programmer eller lignende, der fungerer som faktiske standarder uden at være blevet vedtaget i en standardiseringsorganisation. Her skal

det bemærkes, at der ofte sker det, at noget bliver en de facto standard på trods af en de jure anerkendt standard på området. Eksempelvis er netværks-overvågning idag baseret på de facto standarder i modsætning til allerede vedtagne normer; det samme kan man til en vis grad sige om datakommunikation/systemsammenkobling, hvor TCP/IP idag endnu er mere brugt end OSI-modellen for datakommunikation.

Det er kendetegnende for alle standarder, skulle de være de facto eller de jure, at de ikke er under kontrol af nogen leverandør, og tilgængelig for alle.

Efter at have været bundet i mange år af forskellige leverandørers forskellige særegne implementeringer af specifikke systemarkitekturer på både hardware- og software-niveau er flere og flere brugere gået over til anskaffelse af nye edb-systemer baseret på åbne system principper. Det er endda således, at offentlige indkøb af edb udstyr i meget høj grad er underlagt regler om anvendelse af disse principper, som fastlagt i en beslutning vedtaget i EF's Ministerråd i 1986. (Council Decision 87/95 December 1986).

Det er i alle brugeres interesse at forblive frie til at vælge den bedste måde at integrere ny teknologi på, uafhængig af leverandør interesser og specielle kommercielle/tekniske hensyn/krav disse måtte have.

Flere og flere brugere er derfor gået over til at specificere nye system-løsninger udelukkende baseret principperne fundet i "åbne systemer", idet det muliggør en leverandør-uafhængig edb-strategi, som både er økonomisk såvel som teknisk mulig idag. At en sådan strategier også er fornuftig når man vurderer brugeres edb-behov og værdien af investeringer i edb-teknologi er en helt anden side af samme sag.

## Eksempler på standarder

Et af de bedst kendte eksempler på åben system standarder er vel nok ISO's Open System Interconnection, OSI Men dette er blot en af mange forskellige standarder, der eksisterer idag. Mange ressourcer er brugt i de sidste par år i udvikling af forskellige standarder, der alle ligger til grund for åbne systemer. Hovedvægten har været på aspekterne af OSI vedrørende "system connectivity", men også aspekter som



operativsystem-snitflader, og senest grafiske mand-maskine snitflader har været genstand for standardisering. Alt dette arbejde med standarder er blevet udført i den faste tro, at åbne systemer er godt for både brugere og leverandører.

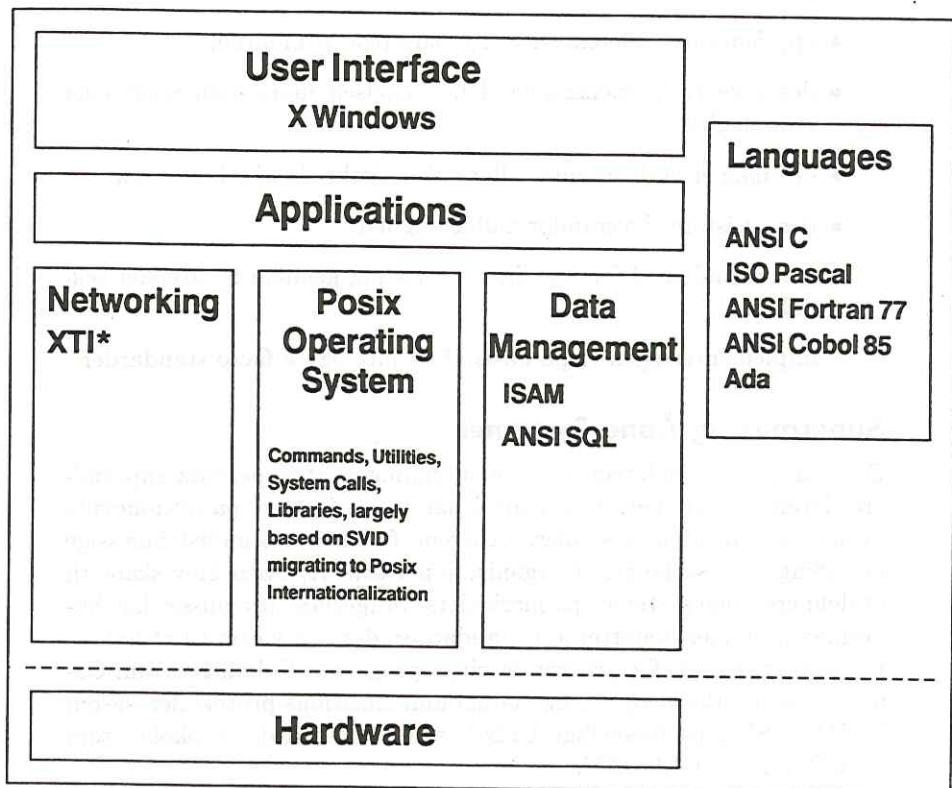
OSI er et af de mest standardiserede områder indenfor åbne systemer. Næsten alle større leverandører har fastlagt en OSI-politik, og mange offentlige indkøb i Europa af edb-udstyr er forudsat, at leverandørerne opfylder basale systemmæssige krav med hensyn til overholdelse af OSI-standarder.

Det skal bemærkes, at selvom nogle OSI-standarder er endeligt vedtaget, såkaldte de jure standarder (jvnf. ovenfor) er andre stadig under udvikling. Men de forskellige standarder, uafhængig af hvor de er i vedtagelsesprocessen er alle struktureret på en sådan måde, at det er rimelig nemt, at inkludere ny teknologi, når og hvis den kommer. Derfor kan brugere, der idag vælger at implementere systemløsninger baseret på OSI-standarder sikre sig, at de også fremover kan have samspil mellem applikationer endnu ikke udviklet, og på trods af der forventes udviklet af mange forskellige leverandører, idet alle vil overholde standarderne.

## Applikations-arkitekturer

Et andet område, som har været udsat for megen standardisering, er egentligt et helt overordnet område i forhold til individuelle standarder. Der er her tale om begrebet applikations-arkitektur, som af mange forskellige leverandører er defineret forskelligt. For eksempel kendes fra firmaet International Business Machines udtrykket SAA for System Application Architecture, medens man fra Digital Equipment Corporation har udtrykket DIA for Digital Integration Architecture. En del leverandører er gået sammen om udviklingen af en fælles applikationsarkitektur gennem deres arbejde i forskellige X/Open arbejdsgrupper. Den arkitektur man er nået frem til er kendt under navnet X/Open Common Applications Environment (CAE). Illustrationen nedenfor viser hvilke elementer denne arkitektur består af, og nogle af de standard-anbefalinger der er indeholdt i X/Opens Portabi-

lity Guides, som sikrer en bærbarhed af applikationer som efterspurgt af markedet.



Dette diagram viser de forskellige områder, som er standardiseret, og som ialt giver en platform for et åbent system. Indenfor hvert delområde er der stadig udvikling, og således forventer tiltag i fremtiden fra X/Open der gør, at aspekter som realtids transaktionsbehandling vil blive behandlet på en ensartet måde af mange forskellige leverandører, og derfor resultere i at brugere vil have adgang til mange forskellige applikationer, ligegyldigt hvilken hardware platform de benytter sig af.



For at resumere kan vi sige, at et åbent system, selvom det betyder mange forskellige ting til mange forskellige brugere og leverandører principielt er et system, hvor

- applikations kildetekster er flytbare pga. standarder
- der ikke er begrænsninger i udbredelsen markeds-mæssigt eller teknologisk
- der ikke er en begrænset eller enkel ejerkreds af teknologien
- der eksisterer leverandør uafhængighed
- der er mulighed for yderligere udvikling gennem nyudviklet teknologi
- implementeringen er på basis af de jure og de facto standarder

## Supermax og Åbne Systemer

Det har været kendetegnende for udviklingen af Supermax supermicro datamat systemet, at det altid har været baseret på ovennævnte principper om åbne systemer. Allerede før den edb-industrimæssige udvikling kom så langt, at organisationer som X/Open blev skabt til at definere åbne systemer på markedets (brugernes) præmisser har Supermax systemet benyttet alle standarder, der har været til rådighed. Eksempler er indenfor programmeringssprog, som Cobol, Fortran, Comal, Pascal, Basic og C, og datakommunikations-protokoller såsom TCP/IP, SNA og forskellige Unix-baserede standard-protokoller som UUCP, CU, og UXCONV.

Som leverandør af total-løsninger i et meget konkurrencepræget marked, hvor mange af DDE's kollegaer indenfor industrien har været store internationale selskaber med filialer over hele verden har det været nødvendigt for DDE, at være istand til at give vore kunder en yderligere værdi, udover et godt arbejdende system, med en løsning for brugerens databehandlingsmæssige behov. En del af denne yderligere værdi har altid i DDE været defineret som værende det, at vort basale produkt-program (det vil sige vor hardware platform og operativ system med dertil hørende hjælpeprogrammel) har været baseret på en modularitet

og åbenhed, der har beskyttet vore kunders investering i edb-teknologi fra begyndelsen. Vor første kunde benytter stadig, efter otte år, sit første system, selvom vi gennem modulær udbygning har formået at mangedoble datakraften i samme system. Få, hvis nogen, af vore kollegaer indenfor edb-industrien kan udtale sig på lignende vis, uden at skulle ud i forklaringer om skift i teknologi og deraf følgende udskiftning af systemer, for at kunne følge med i udviklingen indenfor både teknologi såvel som marked.

## Why UNIX Will Win on the Desktop

Af Dr. Eric E. Schmidt  
Sun Microsystems, Inc.

It's trendy to say that the line between PCs and workstations is blurring. The real truth is that UNIX-based systems are taking over, and by the end of the 1990s, the market will clearly distinguish between two types of desktop computers: machines that run DOS and machines that run UNIX. DOS machines will be purchased primarily as single-user systems for stand-alone applications. And in the long run, the only single-user machines the only DOS machines will be in people's homes.

Why is this? Because the power of distributed, networked computing is effecting a profound change on corporations. The fact is that humans are inherently networked: we inherently need to communicate with each other. Anyone that has networked application needs will turn to UNIX-based solutions, because UNIX is the only multi-tasking, multi-user, multi-vendor operating system available today. That's what makes it the only choice for the next generation of distributed applications. That's why UNIX will win. It's the closest thing the industry has to a pure open standard. Because AT&T Bell Labs made it available in source form in the 1970s, nearly all systems companies either base themselves on UNIX or offer a form of it, including major players such as IBM, Apple, Intel, HP, Sun, DEC, Olivetti, Fujitsu and Unisys.

UNIX is available on all PCs today in one form or another. Within five years, all high-end '386 and '486 PCs will be running UNIX with DOS emulation. Gradually, the final "dumb terminal" applications will disappear when this occurs the death knell of the minicomputer. And mainframes? They aren't going away: they'll become servers. We'll just stop using them with dumb terminals.

The other operating system in contention (more or less) is OS/2. While OS/2 provides multi-tasking, it's still a single-user operating system. It will have some following because of the existing installed base of DOS applications. However, users are looking for open solutions and

OS/2 isn't open. Particularly since IBM recently took back primary control over it. The resulting perception of OS/2 as a proprietary solution will stunt its growth and limit it to the derivative market of DOS applications.

Besides pointing to the "threat" of OS/2, UNIX foes have brought up other arguments. But they just aren't valid anymore. For example:

**Myth:** The UNIX market is splintered.

**Fact:** We're down to two choices: UNIX System V Release 4 and OSF/1. All the major computer companies have endorsed one or the other. OSF/1's sponsors say that it will be interoperable with SVR4, giving us as close to a single standard as possible without having a monopoly.

**Myth:** There aren't any UNIX applications.

**Fact:** The leading PC software applications Lotus 1-2-3, WordPerfect, dBase, Ventura Publisher are ported or are in the process of being ported to UNIX. The combination of these applications and new distribution mechanisms based on CD-ROM will initiate an explosion in UNIX use by former DOS users. CD-ROM is inexpensive, easy to use and stores up to 644 megabytes of information, making it a much more effective software distribution medium than floppy disks.

**Myth:** UNIX is hard to use.

**Fact:** Sun and other companies have created graphical user interfaces like OPEN LOOK that conceal the complexity of UNIX, making it easier to install and use. DOS, of course, has been easier to use because it doesn't do much. There's no functionality, so it's not complicated.

**Myth:** UNIX will always be outdistanced by Macintosh OS and DOS.

**Fact:** Although they currently have a larger market share, UNIX is catching up. According to UNIX International, UNIX has more than 10 million users worldwide as well as the highest growth rate.



There will surely be more than a million SPARC/UNIX seats alone within the next few years, translating to more market opportunity for software developers and more innovative solutions for end users.

Myth: UNIX systems are too expensive.

Fact: A comparably equipped high-end PC now costs more than a Sun SPARCstation, which unlike a PC comes standard with such useful features as Ethernet connection, large memory and a monitor. And the cost of UNIX systems will continue to drop. UNIX is synonymous with "open," which is one reason why UNIX and the accompanying open systems movement are revolutionizing the computer industry. The amazing events of the past year, with long-time proprietary vendors being pushed onto the open systems bandwagon, will only ensure UNIX's ascendancy.

Ovenstående artikel er hentet fra SunFlash-DK-mailinglisten <sup>1</sup>, med tilladelse fra NOKIA Data.

---

<sup>1</sup>SunFlash-DK er en elektronisk nyhedsservice fra Nokia Data A/S rettet mod SUN brugere og kunder.



## DKUUG – Meddelelser fra sekretariatet

Supplerende medlemsliste for perioden 1991-01-11 – 1991-02-26

Nr.	Navn Afdeling/navn fortsat	Adresse Postdistrikt	Tlfnr./lokalur. Kontaktperson	Netadresse
560	FLS Data A/S	Ramsingsvej 7 2500 Valby	36 18 12 35 Poul-Henning Kamp	36 18 12 18
561	Dansk Teknologisk Institut Produktionsteknologi	Gregersensvej, Postbox 141 2630 Tåstrup	42 99 66 11 Henrik Kullmann	43 71 20 24
562	DIAX Telecommunications A/S	Fælledvej 17 7600 Struer	97 86 90 22 Peter S. Christensen	97 85 44 22
563	Mærsk Data A/S	Postbox 176 5000 Odense C	65 97 62 11 Troels Rossen	65 97 68 73
564	Nick Sandru	Høje Tøpholm 37 3390 Hundested	47 98 06 27	
565	Poul Henrik Westh	Valbygårdsvej 33 2500 Valby	31 46 21 00	36 44 14 88
566	Storstrøms Amt	Parkvej 37 4800 Nykøbing F	54 82 32 32 Aage Svanholm	

Status: I alt 0 stormedlemmer, 5 organisationsmedlemmer og 2 individuelle medlemmer.

Rettelser i perioden 1991-01-11 – 1991-02-26, på uformel form, i den rækkefølge, medlemmerne findes i den trykte medlemsliste. Kun ændringer af adresse, tlfnr og lignende er medtaget, ikke konverteringer org. medlemsskab/individ. medlemsskab vice-versa. Udmeldelser er heller ikke medtaget.

58 Aalborg Universitetscenter, Datacentret.

Nyt tlfnr: 98 15 85 22.

396 Aalborg Universitetscenter, Lab for Billedanalyse.

Ny adresse: Fredrik Bajersvej 7 D. 9220 Aalborg Ø

Nyt tlfnr: 98 15 42 11.

446 Danagraf A/S

Ny adresse: Industrivej 11-13, 4180 Sorø (tlfnr uændret)

525 DANOSI

Ny netadr: plh@danosi.dk

337 ERHVERVS-DATA

Nyt navn: ERHVERVS-DATA Århus A/S

548 Gartner Group Skandinavia

korrekt navn: Gartner Group Scandinavia A/S

- Ny adr: Skodsborg Strandvej 156, 2942 Skodsborg  
nyt tlfnr: 45 50 55 56, nyt faxnr: 45 50 55 66
- 538 I.T.I.D. A/S  
Nyt tlfnr: 33 97 03 90
- 473 Kræftens Bekæmpelse  
Netadr tilføjes: geert@crg.dk  
faxnr tilføjes: 31 26 00 90
- 176 Købmandsskolen  
Nyt navn: Niels Brock Købehavns Handelsgymnasium  
Ny kontaktperson: Jørn Folmer Larsen
- 349 Mjølner Informatik  
Nyt tlfnr: 86 20 20 00  
Rettelse kontaktperson navn: Petersen stryges.
- 531 Jørn Møller-Nielsen  
rettes til: Dansk Data Support A/S, Gl Mønt 14, 1117 Kbh K  
tlf 33 15 12 70, kontaktperson Jørn Møller-Nielsen
- 528 Prolog Development Center  
Netadr tilføjes: postmaster@pdc.dk
- 502 SAS, Line Maintenance, RESAM Project Office  
Ny kontaktperson: Nils Erik Olsen  
Nyt tlfnr: 32 32 22 81
- 128 Siemens Nixdorf Info.systemer A/S  
Ny kontaktperson: Preben Alsing
- 316 Skako A/S  
Ny kontaktperson: Morten Hansen
- 314 Niels Svennekjær  
Ny adresse: Vilsundvej 116, 9220 Aalborg Ø
- 162 Texas Instruments A/S  
Tlfnr rettes til: 44 68 74 00
- 270 WordPerfect Danmark  
Tlfnr rettes til: 42 25 11 99, Faxnr tilføjes: 42 25 00 58

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

Listen sendes til:

DKUUG  
Sekretariatet  
Kabellejevej 27 B  
2700 Brønshøj

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.

(telefon og telefax: 31 60 66 80)

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) ..	
- Medlemskab af UniForum (tidligere /usr/group) (ca. ??,-/??,- USD pr år for associeret/generelt medlemskab) ...	
- 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.

Medtilb 90-10-22

## Oversigt over medlemsmøder i 1990-1991

Dato	Sted	Emne
21/03 †	København	Administrative systemer — siden sidst
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.