

Medlemsblad for
Dansk UNIX-system Bruger Gruppe

DKUUG-Nyt

Nummer 23, 1. juni, 1989

Indhold

Redaktionelt	2
DKnet status maj 1989	3
Annoncer i DKUUG-Nyt	6
X Window System	7
System-administration – del 3	13
GNU projektet	19
Adresseliste – med nye telefonnumre	26
Lokale aktiviteter	27
Næste medlemsmøde	27
Oversigt over medlemsmøder i 1989	28

Redaktionelt

DKUUG-Nyts redaktion består af Søren O. Jensen og René Seindal (ansvarshavende). DKUUG-Nyt nummer 24 udkommer d. 1. august 1989. Deadline er d. 14. juli 1989. Deadline for nummer 25 er d. 19. august 1989.

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. kan sendes med elektronisk post til redaktionen på adressen:

`dkuugnyt@dkuug.dk`

eller, hvis man foretrækker almindelig sneglepost, til:

René Seindal
Datalogisk Institut
Universitetsparken 1-3
2100 København Ø
Telefon: 01-39 64 66, lokal 221.

DKnet status maj 1989

Af Kim Fabricius Storm
Texas Instruments A/S

Backbone status

Vores nye SUN-3 backbone er blevet døbt "*slyrf*", idet SUN-2 stadig bærer navnet *dkuug*. Maskinerne deler backboneopgaverne mellem sig, idet *dkuug* primært tager sig af post, mens *slyrf* tager sig af nyheder og navneserverfunktionerne. Disse tjenester fungerer nu fuldt tilfredsstillende, men *dkuug* ligger stadig på grænsen af dens ydeevne, så en yderligere udvidelse af kapaciteten ventes i nær fremtid for at imødekomme den stigende trafik.

Poul-Henning Kamp fratræder som ekstra netpasser efter blot to måneders aktiv tjeneste, idet han har fået job i udlandet pr 1/6. Han har fået sat *slyrf* i drift og har fået stabiliseret en række af vore vitale tjenester, især post og nyheder, samt navneserveren.

Post-tjenesten

Mange ting er blevet ændret (til det bedre) i den senere tid, men der kan da stadig opstå problemer, som der ikke kan ses nogen rimelig forklaring på. Mange af problemerne skyldes at man af vanvare har fundet en ny måde at skrue adresser sammen på. For det meste får den opfindsomme afsender blot sit brev retur med adressaten ubekendt, men undertiden ligner de gale adresser en rigtig adresse og efter en række omskrivninger begået af den berygtede *sendmail* kan adressen komme til at se meget mærkelig ud. Men i sjældne tilfælde kan selv fuldgyldige adresser blive tygget til ukendelighed af *sendmail*; ser man sådanne uregelmæssigheder bedes de rapporteret omgående til

`netpasser@dkuug.dk`.

Som tidligere fortalt er det kun post til Norden, USA og en del af Europa som går via faste linier, mens resten af verdenen primært

nås via X.25. Det er imidlertid lykkedes os at omlægge hovedparten af X.25 trafikken således at den anden part betaler for den, idet vi mener vi betaler tilstrækkeligt til den Europæiske infrastruktur via vore bidrag til de faste linier. Der har således endnu ikke været behov for at håndhæve princippet om at viderefakturere regningerne for den del af posten, der ikke kan sendes via faste linier, idet den resterende del af X.25 trafikken har været ret ubetydelig.

Nyheds-tjenesten

Vi gør endnu engang opmærksom på vores introduktionstilbud for nyhedstjenesten: få første kvartal gratis. De eneste forudsætninger er at man har eller tegner et postabonnement, samt betaler det sædvanlige oprettelsesgebyr.

Det meddeles i øvrigt fra pålidelig kilde at release 6.3 af nn programmet distribueres på nettet (eunet.sources) den 1. juni.

Login-tjenesten

Login-tjenesten fungerer efterhånden tilfredsstillende, så vi regner med at prøveperioden udløber pr. 1/7 (ikke som tidligere nævnt pr. 1/6). Vi vil dog ikke begynde at sende fakturaer til vore brugere uden først at have kontaktet dem med henblik på en tilmelding til den egentligt login-tjeneste.

Modemer

Vi vil endnu engang opfordre vore storbrugere, dvs. primært dem der har nyhedsabonnement, til at overveje anskaffelsen af et TrailBlazer modem, hvor DKUUG har indgået en aftale med 2M elektronik om en kraftig medlemsrabat, således at modem kan tjene sig selv hjem i sparede telefonudgifter på relativt kort tid.

Vi vil gerne have flyttet en større del af trafikken til TrailBlazere, da opkoblingstiderne pga. den større hastighed er langt mindre end for de almindelige modemer, hvorved vi udnytter terminalportene på vore maskiner mere optimalt (terminalporte er desværre en sparsom ressource på vore SUN maskiner).

Konsulenter technical writers oversættere

søges til freelance-arbejde

ARBEJDSOPGAVERNE:

Bearbejdning, nyskrivning, oversættelse, faglig korrektur.

DINE KVALIFIKATIONER:

Praktisk og teoretisk kendskab til UNIX og XENIX, herunder erfaring med Shell-programmering, system-administration, sikkerhed, installering af printere og terminaler samt kommunikation.

Vi lægger vægt på at du kan skrive et ligefremt og letforståeligt dansk og at du læser engelsk ubesværet.

VI TILBYDER :

Samarbejde i et inspirerende og uformelt miljø.

Løn efter kvalifikationer.

Send din ansøgning med et curriculum vitae til Borgen/Data, Valbygaardsvej 33, 2500 Valby.

Hvis du ønsker nærmere oplysninger er du velkommen til at ringe til os på 31 46 21 00 og spørge efter Niels Borgen eller Poul Henrik Westh.

Borgen/Data er Borgens Forlags afdeling for computerlitteratur og software. Vi har siden 1982 udgivet en lang række bøger om edb og computere. Vort mål er at udgive letlæselige computerbøger af høj faglig kvalitet beregnet for et professionelt publikum. Vi søger at nå dette mål gennem udstrakt brug af freelance-medarbejdere.

Borgen/Data

Annoncer i DKUUG-Nyt

Vi er begyndt at bringe annoncer i bladet. Vores mål er at dække bladets trykningsudgifter vha. annoncer. Dette svarer til, at ca. 10% af indholdet vil bestå af annoncer, hvilket er forholdsvis beskedent i forhold til de fleste andre computer-tidsskrifter.

Prisen for en halv side ligger på kr. 500,- og redaktionen forbeholder sig retten til at anbringe annoncerne hvor *den* har lyst—vi vil dog naturligvis såvidt muligt anbringe annoncerne på fremtrædende pladser, men såvel forsiden som bagsiden er annoncefrit område.

Vi kan oplyse, at bladet for tiden udkommer i et oplag på 550 eksemplarer, dvs. at man kun betaler lidt over 1 krone pr. eksemplar for en halvsides-annonce. Endvidere bliver hvert eksemplar af bladet normalt læst af mere end én person.

Annoncerne skal indleveres til den almindelige deadline i re-proklar tilstand (vi vil som sædvanligt påskønne, at få et praj på forhånd om, at der er materiale på vej)



X Window System

Af Steen Lindén
anubis@diku.dk

Efterhånden som arbejdsstationer er blevet mere og mere almindelige i Unix-verdenen, og skærmteknologien har udviklet sig i retning af bitmap orienterede højopløsnings-skærme, er det blevet mere og mere almindeligt at anvende vinduessystemer. SunView på Sun arbejdsstationer og X Window System på bla. Hp, IBM og Digital¹ arbejdsstationer er de to kendteste. Denne artikel omhandler designet af X vinduessystemet.

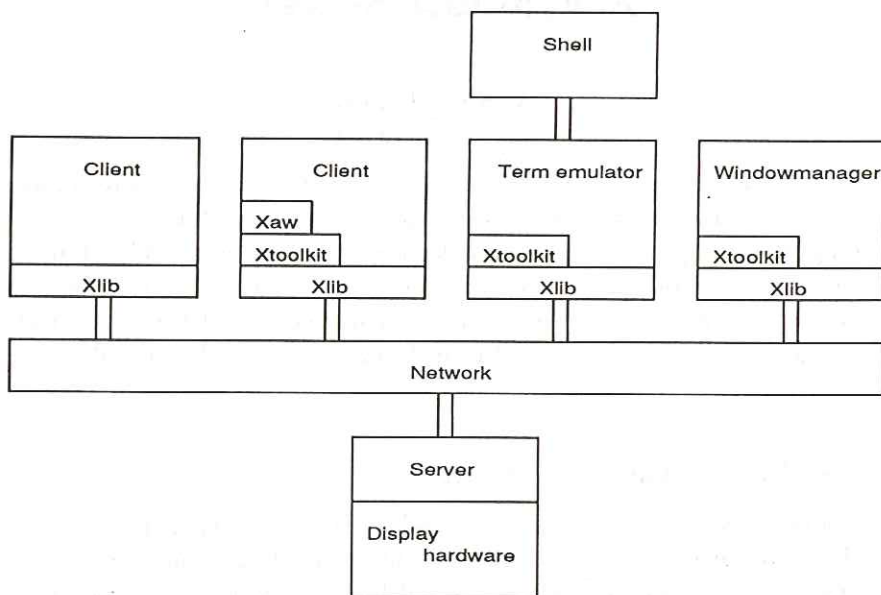
Hvad er X Window System

Fra brugersynspunkt opdeler X Window System arbejdsstationens fysiske skærm i et hieraki af vinduer, hvori der kan tegnes og skrives. Dette vindueshieraki manipuleres ved hjælp af en mus eller en anden anordning, der gør det muligt at bevæge en markør på skærmen. Med denne markør kan man gøre vinduerne større eller mindre, flytte dem rundt på skærmen og dirigere input fra tastaturet til det vindue man ønsker. Det er tilladt for vinduerne at overlappe hinanden, så nogle vinduer er helt eller delvist tildækkede af andre. Effekten ved alt dette er, at f.eks. et antal shells, et ur og et CAD-program kan anvende arbejdsstationens skærm simultant.

X Window System er udviklet på Massachusetts Institute of Technology (MIT) med støtte fra forskellige firmaer, bla. de ovennævnte, der tilsammen udgør X Consortium, som er ansvarlig for den videre udvikling af X standarden.

MIT distribuerer sin version af vinduessystemet *gratis* med kilde-tekster og dokumentation, og sammen med den designfilosofi, der ligger til grund for X, har det bevirket, at X er blevet meget populær overalt i Unix-verdenen. Bla. GNU projektet har adopteret X som det vinduessystem, der supporteres af GNU Emacs.

¹Digital lancerer X under navnet DECwindows.



Figur 1: Server/klient princippet i X

Det grundlæggende design

X er opbygget efter en netværksbaseret server/klient model. Serveren distribuerer input fra musen og tastaturet til klienterne og modtager output til skærmen fra disse. Klienterne er brugerapplikationer, der har forbundet sig til serveren og kan f.eks. være en terminalemulator, der gør det muligt at køre programmer, som er designet til konventionelle terminaler eller et grafisk repræsentation af et analogt ur. Server/klient princippet er illustreret i figur 1

Denne opdeling i klienter og en server, som kommunikerer via X protokollen, bevirker at X er netværkstransparent. Det vil sige, at man foruden at køre klienter på sin arbejdsstation samtidig kan køre klienter på andre datamater i et netværk, også på datamater, der har

anderledes maskinarkitektur og operativsystem i forhold til arbejdsstationen. Det eneste krav er, at serveren og klienten benytter samme netværksprotokol. Dette synes måske i første omgang at være en lidt overflødig egenskab, men den viser sig at være utrolig nyttig, når man har brug for regnekraft fra en datamat, der er større end ens arbejdsstation. Endvidere gør princippet det muligt at fabrikere X terminaler, dvs. intelligente terminaler, som implementerer serverdelen af vinduessystemet i hardware.

MIT distributionen af X implementerer protokollerne TCP/IP, DECnet og Unix domain sockets, men i princippet kan X køre over en hvilken som helst datastrøm, der er hurtig nok.

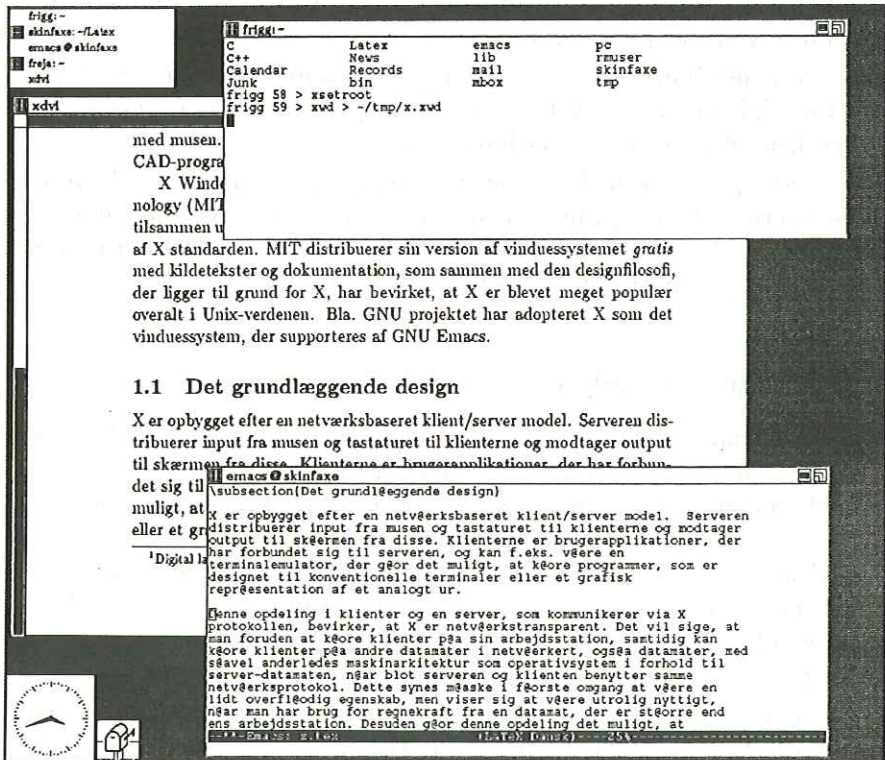
Server/klient princippet bevirker også, at de enkelte X klienter er hardware uafhængige, hvilket medvirker til at gøre dem portable. Netop portabiliteten var et krav, som MIT stillede sig selv ved designet af X og en medvirkende årsag til X's succes.

Brugergrænsefladen

Det grundlæggende princip i X er "*mechanism, not policy*". Systemet må ikke påtvinge brugeren en bestemt grænseflade, som det f.eks. er tilfældet i mange andre vinduessystemer. Under X findes en windowmanager, en speciel klient, som giver brugeren mulighed for at manipulere med sine vinduer i en grænseflade fastlagt af windowmanageren. På samme måde som man vælger shell i Unix, kan man så vælge og tilpasse sin windowmanager efter eget ønske og dermed opnå den grænseflade, der passer bedst til ens temperament.

Windowmanageren er ikke et privilegeret program, der har flere rettigheder end en almindelig klient. Den eneste måde at kommunikere med serveren på er via X protokollen, som er fælles for alle klienter.

Der findes flere forskellige windowmanagere. En populær en er twm. Et eksempel på skærmlayout kan ses på figur 2. Uwm er MIT's windowmanager, som i modsætning til twm ikke forsyner vinduerne med en såkaldt dragbar, men istedet kræver, at man skal bruge en kombination af musen og tastaturets meta-tast, når man skal flytte eller ændre størrelse på et vindue. Rtl er en højst utraditionel windowmanager, der



Figur 2: Eksempel på X med windowmanageren twm.

ikke tillader vinduer at overlape hinanden, men istedet omorganiserer skærmen ved at flytte vinduerne, når en ny klient har brug for plads.

Mens mange vinduessystemer er snævre, med forudfastlagt politik om hvordan f.eks. en scrollbar fungerer, er X bredt med mulighed for mange variationer. Ulempen ved dette er, at X for førstegangsbriugeren virker stort og uoverskueligt, præcis som det er tilfældet for mange, når de første gang stifter bekendtskab men Unix.

X programmering

Designfilosofien i X kan på mange områder sammenlignes med filosofien i Unix. Inderst ligger kernen, som tager sig af fordelingen af ressourcer samt fysisk in- og output. Unix systemkaldene svarer til X protokollen mellem server og klienter, som er implementeret i C i biblioteket Xlib.

I lighed med de mange biblioteksfunktioner under Unix, som supplerer systemkaldene, findes der over Xlib diverse toolkits. Toolkit'et Xt, der som Xlib hører med til X standarden, er et "objektorienteret" toolkit implementeret i C, hvoraf man bygger de såkaldte widgets, f.eks. scrollbars eller dialogboxe.

Med MIT distributionen følger et sæt widgets, Xaw, som er udviklet af Project Athena på MIT. Disse widgets er ikke en del af X standarden, men supporteres alligevel af MIT og ligger til grund for mange af de klienter, der følger med distributionen. Også andre end MIT har lavet widgets til Xt, f.eks. Sony og Hp.

Af andre toolkits kan nævnes InterViews, som er et C++ interface, der bygger på Xlib samt et Common Lisp interface, CLUE, der er en overbygning på CLX; en implementationen af X protokollen i Common Lisp.

MIT X distributionen

MIT's X distribution er delt op i to dele; en keredel, som indeholder de dele, der supporteres af MIT og en del bidraget af andre, som indeholder forskellige toolkits, windowmanagere og applikationer. Distributionen kan skaffes på bånd fra DKUUG eller via anonym ftp til freja@diku.dk (129.142.96.1), hvor der også findes patches.

Serveren, der følger med distributionen, er i følge X11R3 Release Notes blevet bygget under følgende operativsystemer.

- 4.3+tahoe
- Ultrix 3.0 og Ultrix 2.0
- SunOs 3.4
- HP-UX 6.01
- Apollo Domain/IX 9.7 (også 9.6 iflg. udviklerne)
- IBM AOS 4.3 (iflg. udviklerne)
- A/UX 1.0

På Datalogisk Institut har vi MIT's X11R3 installeret på VAXstation2000's, Sun3's, Sun4's og en Vax785, samt IBM's version af X på en IBM RT. Som følge af standardiseringen af X er der ingen problemer i at lade en kommerciel udgave som IBM's køre sammen med MIT distributionen, og det har vist sig, at performance over nettet er lige så god som lokalt på arbejdsstationen.

Litteratur

- [1] Robert W. Scheifler and Jim Gettys, "The X Window System" *ACM Transactions on Graphics* vol. 5, no. 2, 1986. En generel beskrivelse af X med udgangspunkt i version 10.
- [2] Oliver Jones, "Introduction to the X Window System", Prentice Hall 1989. Beskrivelse af Xlib med velvalgte eksempler og kommentarer.
- [3] Joel McCormack and Paul Asente, "Using the X Toolkit or How to write a widget" proceedings from *Summer USENIX '88*, 1988. Introduktion til toolkittet Xt og brugen af widgets.
- [4] David S. H. Rosenthal, "Going for Baroque" *UNIX REWIEW* June 1988. "Hello World" programmet udsat for Xlib og Xt.
- [5] Ed Lee, "Window of Opportunity" *UNIX REWIEW* June 1988. Diskussion af X vinduessystemets design og dets fremtid.

System-administration – del 3

Af Kim Chr. Madsen
UniTack A/S

Backup af Systemet

Som system-administrator er det dit ansvar at sørge for at der bliver taget jævnlig backup af systemet, således at tab af filer og data minimeres, ifald systemet skulle svigte. Som system-administrator er det dig, der har det fulde ansvar og får brugerne på nakken hvis backup's er utilstrækkelige i en kritisk situation. Du bliver også upopulær hvis denne backup tager flere timer hver dag, hvis det foregår i den almindelige arbejdstid eller det kunne være klaret hurtigere.

Selv om systemet aldrig fejler og kontoret ikke brænder, vil tilstrækkelige backup's kunne gøre dig til dagens helt, ved at være istand til at bringe andre brugeres filer, som de slettede ved et "uheld", tilbage på systemet.

Der findes to metoder du kan benytte ved backup, en total backup eller en partiel backup. Den totale backup er meget tidskrævende og kræver en del bånd eller disketter. Den partielle backup kan vælges som supplement til den totale backup, og fylder mindre og tager ikke nær så lang tid dagligt.

Total Backup

Det anbefales at tage en total backup en gang om ugen, dette kan normalt gøres næsten automatisk, idet de fleste Unix-systemer har indbyggede værktøjer til at klare dette, ofte lagt ind i et menusystem. Disse værktøjer er dog forskellige fra system til system. Den totale backup tager en del tid, idet hele indholdet af harddisken skal kopieres til det eksterne medie. Erfaring viser at det tager ca. 20-30 minutter at kopiere 60Mb data til streamer tape, og ca. 5-10 minutter at kopiere 0.7Mb data til floppy diskette.

Mange gange vil man kun tage total backup af brugerfiler og data, hvormed man undgår at tage backup af selve Unix-systemet og standard-

applikationer, der er installeret. Selve Unix-systemet fylder (når alt er installeret) ca. 10-13Mb. Dette er naturligvis kun en farbar vej, når man ikke ændrer i operativsystemet eller i standard applikationerne.

Inkremental Backup

Denne backup metode kræver en total backup som grundlag, idet man blot kan se denne som en a'jour-føring af den totale backup. Metoden går ud på at man hver dag tager en backup af de filer som er blevet dannet eller ændret siden den sidste total backup eller inkremental backup. Denne metode vil typisk ikke tage ligeså lang tid som den totale backup, med mindre de fleste filer på systemet bliver modificeret ofte, ellers ligger normalen på ca. 10-15Mb dagligt på et 20 brugers Unix-anlæg, dog betydeligt mere hvis der findes store databaser, der bliver ændret i dagligt.

Selektiv Backup

Denne metode kan ses som et supplement til de to ovennævnte metoder, og består i at man udvælger de filer eller kataloger man ønsker at tage backup af. Dette kan være databaser, meget vigtige filer, eller f.eks. alle filer tilhørende et bestemt projekt eller en bestemt gruppe af brugere eller filer tilhørende en bestemt bruger. Metoden benyttes ofte i forbindelse med at man fjerner en bruger fra systemet, men ønsker at bevare en kopi af denne brugers filer og data til senere brug eller genindlæggelse af brugeren.

Valg af Backup Metode

Som det ses kan man tage backup på flere forskellige måder, og der er fordele og ulemper ved alle måderne.

Hvis man vælger Inkremental Backup metoden, vil dette bevirke at normalt en gang om ugen tager en total backup, og resten af ugens dage tager en backup af nye og ændrede filer, vil man kunne komme ud for at når backup'en skal lægges ind på systemet igen at skulle igennem op til seks backup bånd i omvendt rækkefølge for at genetablere systemet. Dette har dog også en anden kedelig bivirkning, nemlig at filer der er slettet siden sidste total backup vil blive genoplivet på systemet. I

grelle tilfælde vil der slet ikke være plads på systemet til at geninstallere sine inkremental backup-bånd tilbage til total backup'en.

Hvis man derimod vælger en total backup, vil denne være tidskrævende i det daglige, men mere effektiv når backup'en skal køres ind på systemet igen. Men har man tiden og backup medier nok til denne form for backup, er det klart den sikreste.

Backup værktøjer

Der findes idag to forskellige programmer til at lave backup med, nemlig **tar** (Tape ARchiver) og **cpio** (CoPy In/Out). Grunden til dette er at der med introduktionen af System V, blev skabt en ny standard for hvorledes backup formatet skulle se ud. Denne nye standard er grundlaget for **cpio**, hvorimod **tar** er den gamle metode, som dog stadig bruges på mange Unix og Unix-lignende systemer.

Hvis **cpio** findes på den aktuelle maskine er denne at foretrække, idet alt tyder på at denne bliver den fremtidige standard, og at den oftest er hurtigere end **tar**.

Tar Backup

Tar, er en forkortelse for *Tape Archiver*, og bruges til at flytte filer fra harddisken til eksternt backup medie. **Tar** er mindre fleksibel end **cpio**, og er derfor nemmere at bruge, men har også visse andre fordele:

- Megen standard software kommer stadig i **tar** format.
- **Tar** dumps kan opdateres ved at tilføje nye data i slutningen af datamediet, givet at der er plads.

For at være fair har **tar** også en række ulemper, disse er:

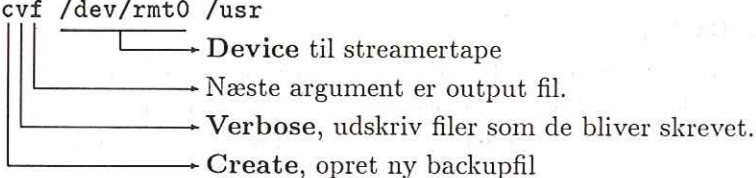
- **Tar** holder ikke rede på rettigheder på kataloger, hvilket vil sige at **tar** må gætte sig frem til de rettigheder katalogerne havde når backup'en køres ind igen. Dette gøres ved at kopiere rettighederne for den første fil i kataloget. Dette har dog kun betydning hvis kataloget skal oprettes.
- Det er ikke muligt at udvælge lister af filer, med samme nemhed som i **cpio**.

- Det er ikke muligt med `tar` at tage backup af kataloger, der indeholder specielle filer, dvs. devices, FIFO filer etc.
- `Tar` kan normalt ikke klare backup's, der strækker sig over flere medier.
- Forskellige Unix-leverandører har en tendens til at tilføje ikke-standard options til `tar` programmet. Dette betyder at den option sammensætning, som du hele tiden har brugt på dit eget system ikke gælder på et andet system. Et eksempel herpå er, at visse leverandører understøtter `tar` dumps, der strækker sig over flere medier (bånd eller disketter), medens andre systemer ikke gør dette.

Eksempler på brug af tar

Der ønskes taget backup af `/usr` filsystemet på streamertape. dette kan gøres med hjælp af følgende kommando:

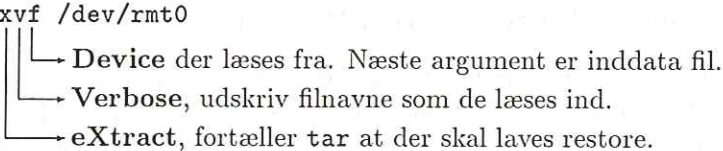
```
tar cvf /dev/rmt0 /usr
```



→ Device til streamertape
 → Næste argument er output fil.
 → **Verbose**, udskriv filer som de bliver skrevet.
 → **Create**, opret ny backupfil

Der ønskes lavet en `restore` af den backup, der findes på streamertape, dette gøres vha. følgende kommando:

```
tar xvf /dev/rmt0
```



→ Device der læses fra. Næste argument er inddata fil.
 → **Verbose**, udskriv filnavne som de læses ind.
 → **eXtract**, fortæller `tar` at der skal laves restore.

Cpio Backup

`Cpio`, er en forkortelse af *CoPy In/Out*, og bruges ligesom `tar` til at flytte data og filer fra harddisk til eksternt backup medie.

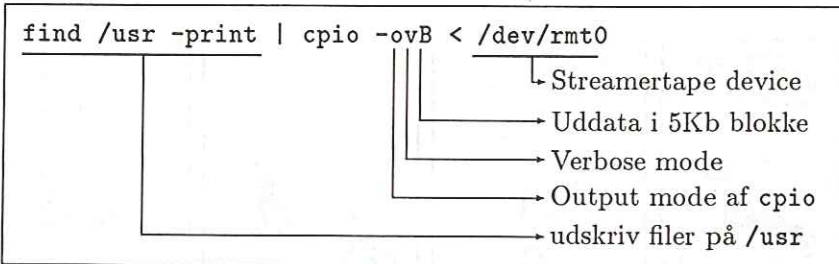
Cpio er en implementation af hvorledes backup-formatet skal se ud ifølge *SVID*² og fremtidige Unix-versioner.

Cpio kan det samme som tar, men også mere. Udvælgelsen af filer sker nemmere og mere præcist. Den større fleksibilitet har selvfølgelig også sine omkostninger, som mest giver sig udslag i kompleksiteten i brugen af programmet.

Det er naturligvis lidt bedrøveligt at der hersker forvirring omkring hvilket værktøj, der skal benyttes til at tage backup med, og min anbefaling må derfor være at man udfører backup på en måde så det for brugeren er ligegyldigt hvilken metode der benyttes. Dette kan nemt opnås ved hjælp af et menu system, som skjuler det reelle backup program for brugeren.

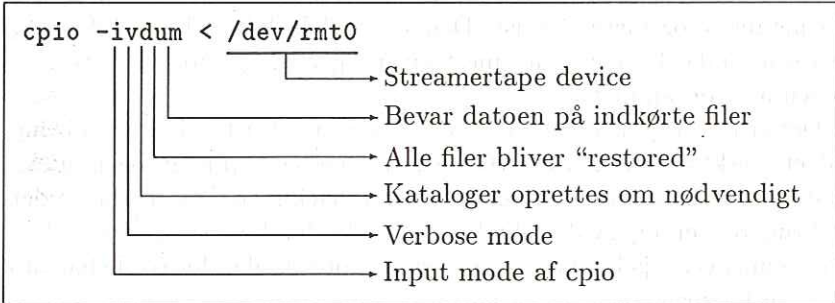
Eksempler på brug af cpio

Der ønskes taget backup af /usr filsystemet, dette kan gøres vha. følgende kommando:



²SVID er en forkortelse for System V Interface Definition AT&T's definition af Unix System V.

Der ønskes en restore fra streamertape, hvor alle filer ønskes kopieret ind.



GNU projektet

Af René Seindal
seindal@diku.dk

Vi bringer her på opfordring en artikel om hvad GNU projektet er for noget, da der tilsyneladende er mange, som ikke ved hvad det er. Artiklen vil dels tage udgangspunkt i mine egne erfaringer med GNU programmet, og dels i den seneste udgave af GNUs nyhedsblad, "GNU Bulletin," fra januar i år.

Indrykket tekst er ordrette uddrag af "GNU Bulletin." Da disse afsnit er på engelsk, skulle forveksling ikke være mulig.

Hvad er GNU

GNU er en projekt, igangsat af Free Software Foundation (FSF), med det formål at udvikle og distribuere gratis programmel. Med deres egne ord er ideen med Free Software Foundation:

The Free Software Foundation is dedicated to eliminating restrictions on copying, redistribution, understanding and modification of computer programs. We do this by promoting the development and use of free software in all areas of computer use. Specifically, we are putting together a complete integrated software system called "GNU" (GNU's Not Unix) that will be upward compatible with Unix. Some large parts of this system are already working and we are distributing them now.

The word "free" in our name refers to two specific freedoms: first, the freedom to copy a program and give it away to your friends and co-workers; second, the freedom to change a program as you wish, by having full access to source code. Furthermore, you can study the source and learn how such programs are written. You may then be able to port it, improve it, and share your changes with others.

Other organizations distribute whatever free software happens to be available. By contrast, FSF concentrates on development of new free software, building toward a GNU system complete enough to eliminate the need to purchase a proprietary system.

Selvom programmelt er gratis, og der derfor ikke er nogen service, er man dog ikke helt på herrens mark, idet:

After we create our programs, we continually update and improve them. We release between 2 and 20 updates a year, for various programs. Doing this while developing new programs takes a lot of work, so any donations of pertinent source code and documentation, machines, labor or money are always appreciated.

Service og support på programmelt bygger således på brugernes egne rettelser og tilføjelser. Det forventes at man, hvis man retter fejl eller forbedre programmerne, sender sine rettelser tilbage til Free Software Foundation, hvor de så måske finder sin vej ind i distributionen. Man skal ikke regne med at blive betalt for sit arbejde, idet det forventes at man, da man jo har haft glæde af det Free Software Foundation har givet en gratis, vil være villig til at givet noget tilbage.

Personerne

Der er efterhånden mange personer involveret i Free Software Foundation og GNU, men nogle er mere fremtrædende end andre.

Den utvivlsomt vigtigste person i Free Software Foundation er Richard Stallman, tidligere ved MIT, idet han har stiftet Free Software Foundation. Stallman har selv bidraget til GNU med programmel som GNU Emacs, som er en udvidbar editor, og GNU CC, som er en ANSI kompatibel C oversætter. Derudover arbejder han som koordinator for alle de personer, der skriver programmel og dokumentation til GNU. Til sin hjælp har han Leonard Tower, som bl.a. tager sig af meget af det administrative arbejde.

Af de mange personer, der bidrager med programmel og dokumentation, kan nævnes Michael Tiemann, fra Stanford University, som har

lavet 'G++,' der er GNUs C++ oversætter; Roland McGrath, som har lavet GNUs make og som arbejder på et ANSI kompatibelt C bibliotek; Doug Lea, som har skrevet 'libg++,' et C++ bibliotek; samt mange andre, som oftest arbejder på mindre separate projekter.

Filosofien

Drivkraften i hele projektet er idealisme, da der jo per definition ikke er penge i at udvikle gratis programmeler. De bagved liggende tanker er blevet udtrykt meget præcist af Doug Lea, i en artikel på USENET, som det følgende er et uddrag af:

I am, primarily, a teacher in a liberal arts college. As such, I stand for the 'free' dissemination of ideas. Historically, (please forgive any botching of historical facts to suit my needs, but that's what history is for!) the main tool by which intellectual property has been allowed to be widely disseminated (read 'taught') while at the same time both crediting originators, and protecting the works from corruption, mis-attribution, and so on, has been the notion of Copyright. For these reasons, the introduction of copyright laws is widely considered to have been an important step in accelerating intellectual and scientific progress.

Sadly, in the science of computing, this solution has not stood up well. While, in many disciplines, the price of a copyrighted work to be used for study is well within the reach of those who could best benefit from it (e.g., a copy of "War and Peace" might be \$5, or even \$50, but not \$50,000), the economics of computing have, for the most part, priced copyrighted software out of the reach of students (and most others). Most readers would agree that the study of high-quality existing programs is among the better methods for learning about the art of programming. These days, one cannot legally show, discuss, and teach from, say, Unix or Lotus source code.

I believe that Stallman's notion that the economics of copyright can be separated from its role in the protection

and propagation of intellectual property is as good a solution to this dilemma as we are likely to get. There are many of us, especially those of us in academe, who are actually very pleased to devote some time and effort to writing software without any direct monetary compensation. For all sorts of reasons. (For example, in my case, with 'libg++,' as a means to further investigate the pragmatics of object-oriented programming and so on. Or maybe it's just incorrigible hacking. Whatever.)

Now I, and many others, I suspect, are not terribly worried about maintaining proper authorship credit, etc., of such work. The reason that the GNU License Agreement is attractive is mainly that it keeps accessible the work that I intended to be accessible, but also generally offers all other benefits that Copyright engenders, but that the mere act of placing work in the 'public domain' would not.

GNUs copyleft

Free Software Foundation forsøger at beskytte deres programmel mod misbrug, herunder salg, med noget de kalder en "copyleft." Hovedformålet med en "copyleft" er at sikre at det programmel, som forfatteren ønskede skulle være gratis, vil vedblive at være gratis. Free Software Foundations egen forklaring følger:

The simplest way to make a program free is to put it in the public domain. Then people who get it from sharers can share it with others. But bad citizens can also do what they like to do: sell binary-only versions under typical don't-share-with-your-neighbor licenses. They would thus enjoy the benefits of the freeness of the original program while withholding these benefits from the users. It could easily come about that most users get the program this way, and our goal of making the program free for *all* users would have been undermined.

To prevent this from happening, we don't normally place GNU programs in the public domain. Instead, we protect

them by what we call "copylefts." A copyleft is a legal instrument that makes everybody free to copy a program as long as the person getting the copy gets with it the freedom to distribute further copies, and the freedom to modify their copy (which means that they must get access to the source code). Typical software companies use copyrights to take away these freedoms; now we software sharers use copylefts to preserve these freedoms.

Projektets status

Indtil nu har vi mest set på hvad Free Software Foundation og folkene bagved ønsker at lave. Hvad har de så opnået? Indtil nu har GNU produceret en meget stor del programmeler. Det meste af det er programmør-orienteret. Der er ingen spread-sheets og regne-programmer.

GNU Emacs er nok det mest kendte GNU produkt. Den er skrevet af Richard Stallman, og har i tidens løb (siden 1984) gennemgået en omfattende udvikling. Den er i dag nok den mest omfattende editor der findes. Den har specielle 'modes' til mange forskellige typer tekst, og kan bl.a. lave automatisk indrykning af C og lisp programmer. Stort set hele Emacs' funktionalitet er implementeret i Emacs' egen indbyggede lisp variant, og det er derfor fantastisk let at lave nye funktioner, eller modificere de allerede definerede (som man naturligvis har kildetekster til). Emacs kan stort set alt (ellers kan man få den til det), så nogle folk starter efter sigende en Emacs om morgenen, og forlader den ikke før de har fri. Emacs kan vist køre på stort set alle 32 bit maskiner, og den er i hvert fald portet til hundreder forskellige arkitekturer. Emacs kommer med en on-line manual, som udskrevet fylder 300 sider. Emacs Lisp er dokumenteret med et 400 siders dokument.

GNU har efterhånden et komplet sæt programmer til programudvikling i C. Det inkluderer en portabel, optimerende ANSI C oversætter, som er både hurtigere og laver mere effektiv kode end de fleste kommercielt tilgængelige oversættere. GNU CC kommer med en lænker, som bl.a. kan angive uløste reference med linie henvisning til kildeteksten. Der er bison, som er GNUs yacc (?) og flex, som er en "fast lex." Begge har udvidet funktionalitet i forhold til deres Unix modstykker,

og er samtidigt hurtigere. GNU har en version af make, som er den mest omfattende make jeg nogen sinde har set.³ GDB er GNUs debugger, som arbejder på kildetekst-niveau, og man behøver derfor stort set ikke kende til maskinsprogs-programmering for at kunne bruge den. Til sidst, så indeholder sættet programmer som size, nm, mm., fordi GNUs oversættere og lænkere ikke vil arbejde med objektfiler i COFF-format.

G++ er GNUs C++ oversætter, som i den nuværende udgave implementerer en større del af sproget end nogen anden C++ oversætter. G++ benytter GNU CCs kodegenereringsmodul, og kan derfor foretage mange af de samme optimeringer. Sammen med G++ kommer libg++, som er et meget omfattende C++ bibliotek, som indeholder klasser til alt hvad hjertet kan begære.

Af resten af GNUs programmer kan nævnes GNUs diff og grep, som efter sigende er verdens hurtigste. GNU leverer også programmer som ls, tar, sed, et interaktivt plot-program, en PostScript fortolker, flere netnews programmer, og meget mere.

Programmellens kvalitet

Programmernes kvalitet er til enhver tid varierende. Flere af de nyere programmer er af Free Software Foundation blevet mærket som beta-test distributioner, og programmerne kan være meget ustabile. På dette punkt adskiller Free Software Foundation sig fra de fleste kommercielle firmaer, idet de gerne frigiver programmet i *meget* tidlige faser i udviklingsforløbet. Det giver fremtidige brugere mulighed for at bidrage til den videre udvikling, og medvirker til at øge de færdige programmets kvalitet.

Generelt kan det siges at med tiden er de fleste af de ældre programmer blevet meget stabile, heriblandt Emacs, GNU CC, bison, flex, og mange af programmeringsværktøjerne. Jeg bruger selv mange af programmerne i det daglige, og oplever kun få problemer, som mest opstår på de områder, hvor GNUs programmet ikke er bagudkompatibelt med det tilsvarende Unix programmet.

³Der er vist ikke det den ikke kan, måske med undtagelse af at lave kaffe, og punktsvejs under vandet.

Det generelle mønster er, at de tidlige udgaver er praktisk taget ubrugelige til daglig brug. De sendes ud for at få foreslag, fejlreportinger og rettelser tilbage og man kan deltage i arbejdet, hvis man er interesseret. De senere udgaver er gerne ved at stabilisere sig, og kan bruges til almindelig brug, men man skal dog være forberedt på at støde ind i fejl. Efterhånden som fejlene bliver rettet, bliver programmerne dog mere og mere stabile.

Distribution af GNU programmel

GNU programmel distribueres på flere måder. Da programmerne ifølge deres copyleft må kopieres frit, kan man til enhver tid blot få et kopi af naboens eksemplar. Hvis man ikke kender nogen, som kan eller vil forsyne en med det man ønsker, kan man bestille det på bånd fra Free Software Foundation, EUUG eller i visse tilfælde fra DKUUG. I det sidste tilfælde skal henvendelse ske til DKUUGs netpassere (elektronisk post til netpasser@dkuug.dk eller tlf. 31 39 73 22).

Hvordan man bidrager til GNU

Hvis man har noget, som måske kan have interesse for Free Software Foundation, kan man skrive til gnu@prep.ai.mit.edu og give dem en beskrivelse af hvad det er man har at tilbyde. Hvis man blot ønsker at lave noget arbejde for dem, kan man også skrive og bede om deres "wish list" over de ting, som de godt kunne tænke sig, men endnu ikke har nogle til at lave.



Adresseliste – med nye telefonnumre

Bestyrelsen

Formand:	Keld Jørn Simonsen 33 13 00 23	Center for Anvendt Datalogi keld@dkuug.dk
Næstformand:	Kim Biel-Nielsen 42 89 49 99	UNIWARE danmark A/S kbn@kontiki.dk
Kasserer:	Tonny Andersen 42 11 08 16	Advanced Computer Software tonny@acs.dk
	Isak Korn 33 93 08 99	IT Center isak@itc.dk
	Kim Storm 42 91 74 00	Texas Instruments storm@texas.dk
	Poul-Henning Kamp 53 53 12 24	DKnet phk@dkuug.dk
	Steen Koefoed Larsen 42 80 75 22	Ambrasoft A/S steenkl@ambush.dk
	Hans Kierulff 42 63 01 55	Merkur Data-Service A/S hans@dkuug.dk

DKUUG og DKnet

Sekretariat:	Inge og Mogens Buhelt 31 60 66 80	Kabbelejevej 27 B, 2700 Brønshøj mogens@dkuug.dk, Giro: 1 37 86 00
Netpasser:	Claus Engdahl 31 39 73 22	Datalogisk Institut, Kbh. Univ. netpasser@dkuug.dk

EUUG

EUUG-repræsentant:	Keld Simonsen
Suppleant:	Steen Koefoed Larsen

Ansvarlige for bestyrelsens udvalg

Netudvalg:	Kim Storm
Medlemsmødeudvalg:	Steen Koefoed Larsen
Medlemsserviceudvalg:	Keld Simonsen
Blad- og PR-udvalg:	Keld Simonsen
Leverandørudvalg:	Hans Kierulff
Økonomiudvalg:	Tonny Andersen
Eksternt udvalg:	Keld Simonsen

Lokale aktiviteter

København, d. 11. april Næste Pub-crawling foregår på vores stamværtshus AW, Gl. Kongevej 41, fra kl. 20:00 til folk segner. De morgenfriske kan mødes kl. 18:00 til spisning på restaurant "Portofino," som er beliggende lige ved siden af AW. Pub-crawling er en fast begivenhed i København, som finder sted den anden tirsdag i hver måned (dvs. pubcrawlingen i juni måned foregår d. 13.).

Disse pubcrawls har vist sig at være en bragende succes (det overvejes om der skal bestilles flere billardborde). Der burde være basis for lignende arrangementer andre steder i Danmark. Hvis nogen skulle være interesserede i at arrangere noget sådant vil vi gerne høre om det.

Næste medlemsmøde

Som tidligere annonceret bliver temaet for det næste medlemsmøde *Migration til UNIX*. Da medlemsmøde-udvalget lider af mangel på manpower har de endnu ikke et færdigt program klart, når det foreligger vil det blive udsendt. Datoen burde dog ligge fast, nemlig d. 14.06.89.

Oversigt over medlemsmøder i 1989

Dato	Sted	Tema
14/6 †	København	Migration til UNIX.
28/9	København	Distribuerede systemer.
11/10 †	Provinsen	Industrielle systemer.
27-28/11	København	Årsmøde med netindslag.

De med † markerede møder er eftermiddagsarrangementer, som typisk er af ca. 2 timers varighed, placeret efter normal arbejdstid. Disse møder er gratis. De øvrige møder er heldagsarrangementer.

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