

Medlemsblad for Dansk UNIX-system Bruger Gruppe

DKUUG NYT

Nr 12, april 1987

Redaktion:Carsten Reimers, BKI (ansvarshavende)
Keld Simonsen, Center for Anvendt Datalogi**Adresse:**DKUUG NYT ved Carsten Reimers,
Beton- og konstruktionsinstituttet
Postbox 82, 2970 Hørsholm
Tlf: 02 86 63 11

DKUUGs bestyrelse kan træffes:

	Telefon:	Netadresse:
Formand: Keld Simonsen, Center for Anvendt Datalogi	01 12 01 15	keld@diku.uucp
Kasserer: Hanne Andersen, SC Metric A/S	02 89 35 00	
EUUG: Kim Biel-Nielsen, SC Metric A/S	02 89 35 00	
DKUUG NYT: Carsten Reimers, BKI	02 86 63 11	
DKnet: Keld Simonsen, Center for Anvendt Datalogi	01 12 01 15	keld@diku.uucp
Leverandører: Kim-Biel-Nielsen, SC Metric A/S	01 83 95 11	UPS!
Leverandører: Hanne Andersen, SC Metric A/S	02 89 35 00	
Øvrigt: Isak Korn, Ericsson A/S	02 84 33 66	
Øvrigt: Erik Wismann, Datacentralen	01 46 81 22	
Øvrigt: Lars Povlsen, Datalogisk Institut		krus@diku.uucp
Administrator: Mogens Buhelt, SBI	02 86 55 33	privat: 01 60 65 80

Indhold

Det tog sin tid, men endelig har vi fået fastlagt DKUUGs forårsmøde. Det blev til

4GL møde, torsdag den 7. maj kl. 13 - 17 hos IBM i Lyngby.

Mødet handler om 4. generations værktøjer til databasehåndtering mv. samt om nogle produktannonceringer fra IBM.

Vi har også fået fastlagt mødekalender for resten af 1987, samt emner for møderne, så nu er det tid at reservere datoerne i kalenderen. Det er UNIX-introduktion 26. maj, grafik og vinduer mm. 10. september i Århus og general-forsamling og årsmøde den 26. november.

Endelig annoncerer vi en ny spændende mødeform i DKUUG.

Vi har lidt nyt fra rygtesmedene om AT&T og IBM og nyt om leverandørgruppen og udstillingen i Bella Centret. Der er en revideret medlemsliste vedlagt samt lidt om hvordan den bruges. Vi videregiver nyheder om EUUGs Helsinki konference samt X/OPEN portability guide.

Der er sket en del ting omkring nettet på det sidste. Datalogisk Institut har frabedt sig ansvaret for at koble nye brugere på nettet og anden telefonisk kontakt med DKUUGs medlemmer iøvrigt. Men der er vist ikke så slemt som det lyder. Kort fortalt ser det ud til at DKUUG overtager hele ansvaret for nettet incl. det økonomiske, Datalogisk Institut yder fortsat teknisk support, DKUUG ansætter en person til brugersupport og priserne bliver sat lidt op. Se artiklerne om Datalogisk Institut og nettet, samt omtale af mødet om det danske UNIX-nets fremtid.

Læs desuden om DKUUGs nye maskine, det nye EUUG konferencebånd og hvordan man får fat i udvalgte dele via nettet, samt andre nyheder.

Lars Povlsen har iøvrigt orlov fra sit job som network manager på Datalogisk Institut, da han skal aftjene værnepligt i flyvevåbenet. Vi ønsker Lars god vind hvor han kommer frem!

Vi har taget en større artikel fra nettet af Andy Tanenbaum om MINIX.

Endnu en gang har vi fået revideret DKUUG-pjecen - 1987 udgaven! Et eksemplar er vedlagt, og flere kan bestilles hos Mogens Buhelt. Den er trykt i 4000 ex. og især leverandører, der vil gøre lidt reklame for DKUUG til deres kunder, bedes rekvirere nogle stabler.

Medlemsmøde om 4. generations (4GL) værktøjer samt IBM Unix

Det første DKUUG medlemsmøde i år får som tema "4. generations databaseværktøjer under UNIX" - ja, vi har diskuteret en del hvad 4GL egentlig var for en størrelse og mener dette er den mest dækkende betegnelse for hvad der omtales på mødet. Desuden byder værten, IBM Danmark A/S, på nogle nyheder mht. deres IBM 6150 (RT) UNIX maskine samt en demonstration af den. Mødet holdes:

Torsdag den 7. maj 1987 kl. 13.00 - 17.00
IBM Danmark A/S, Nymøllevej 85, 2800 Lyngby

Vi beder jer forhåndstilmelde med angivelse af firma og navne til Mogens Buhelt, tlf 02 86 55 33 eller på nedenstående tilmeldingsformular. Højest tre pr organisationsmedlem og én pr individuelt medlem.

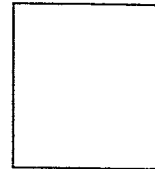
Tilmelding til DKUUG medlemsmøde torsdag d. 7. maj 1987
hos IBM, Nymøllevej 85, 2800 Lyngby

Følgende personer tilmeldes (organisationsmedlemmer kan tilmelde op til tre personer, individuelle medlemmer kun én):

Medlem (firma): _____

Navne: _____

Brev



DKUUG
C/O: Mogens Buhelt
Kabbelejevej 27 B
2700 Brønshøj

Program for DKUUG-medlemsmøde 7. maj 1987:

- 13.00 - 13.30: **Tendenser inden for applikationsudvikling og 4GL**
Tue Bertelsen, Ambrasoft A/S
Har udviklere virkelig brug for 4GL? Er det ikke således, at det er lige så kedeligt at programmere i 4GL som i Fortran? Nogle mener at 4GL giver dårligere og mindre pålidelige applikationer. Er dette rigtigt? Er det korrekt at der kun vil være 3 leverandører af 4GL tilbage på markedet om 3 år? Skal vi hellere skrive programmer med andre moderne værktøjer som fx C++? Er anvendelsen af 4GL den eneste chance for at UNIX-markedet kan overleve i forhold til PC-markedet? Tue Bertelsen fremlægger en række synspunkter, observationer og erfaringer, samt orienterer om tendenserne set ud fra en beslutningstagers synspunkt.
- 13.30 - 14.00: **Om valg af 4GL værktøjer**
Hans Kierulff, Merkur Data A/S
Indlægget giver et resumé af 4GL teknologien samt en metode til valg af 4GL i en given situation. Der omtales: 4GL's faciliteter, eksperimentel systemudvikling, baggrunden for 4GL, opgaver for 4GL, primære krav til 4GL samt hvordan 4GL kan vælges.
- 14.00 - 14.15: **Informix præsentation**
Kim Biel-Nielsen, SC Metric A/S
- 14.15 - 14.30: **Gennemgang af TODAY**
Claus Bytzau, Merkur Data A/S
TODAY er en moderne applikationsgenerator med indbygget data dictionary, non-procedural programmering og automatisk dannelse af funktioner, velegnet til eksperimentel udvikling af komplekse administrative edb-systemer.
- 14.30 - 14.45: **Dataflex præsentation**
Lars Balle Andersen, Eurosoft Danmark Aps
- 14.45 - 15.00: **Accell præsentation**
John Brun, Ambrasoft A/S
Accell er ikke et 4GL, men et applikationsudviklingssystem, der integrerer en række 4GL værktøjer: database, applikationsgenerator, windowing environment, integration tools - og et non-proceduralt 4GL. Dette skyldes en erkendelse af, at ingen af disse værktøjer er nok i sig selv til professionel udvikling, men ved at kombinere dem i et integreret system opnås en betydelig synergieffekt. Idéen bag applikationsudvikling i Accell er: Brug kun 4GL når det undtagelsesvist er nødvendigt! Denne filosofi gør det muligt at udarbejde fungerende prototyper af en applikation meget hurtigt sammen med en kunde og siden færdiggøre applikationen ved at tilføje 4GL-kode, hvor Accells applikationsgenerator ikke er dækkende. Accell er for nylig valgt af General Motors som primært UNIX-udviklingsværktøj til over 10.000 UNIX-installationer.
- 15.00 - 15.30: **Pause: kaffe, the mm.**
- 15.30 - 16.00: **Paneldiskussion om UNIX og 4GL**
Med deltagelse af alle ovennævnte foredragsholdere under ledelse af Tue Bertelsen.
- 16.00 - 16.30: **IBM 6150 (PC/RT) - nyheder**
Jens Allerslev Jensen, IBM Danmark A/S
De seneste nyheder på IBMs UNIX-maskine serie 6150 (i USA kaldet PC/RT) vil blive omtalt. 6150 har været omtalt på et par tidligere DKUUG møder.
- 16.30 - 17.00: **IBM 6150 demonstration**
Lars Thorsen, IBM Danmark A/S

Medlemsmøde den 26. maj - Hvorfor UNIX?

Bestyrelsen har fastlagt endnu et møde, hele dagen den 26. maj i København (somewhere). Denne gang bliver det et bredt anlagt møde om UNIX - og alle er velkomne, også ikke-medlemmer. Der vil blive annonceret i fagpressen om dette "dyt i bamsen". De planlagte emner er: Hvad er UNIX? UNIX og MS-DOS; C og Pascal, Brugerforeningen DKUUG samt UNIX-nettet.

Lokale UNIX møder

I lighed med englænderne og tyskerne vil nogle UNIX-brugere i København forsøge at lave en Copenhagen UNIX Pub / Stammtisch Kopenhagen hvor vi over en stille øl og en pot billiard vil mødes og sludre om UNIX, fodbold, kvinder, mænd samt hvad hjertet ellers kan begære. Det sker den anden tirsdag i hver måned på værtshuset "Ambrosius" på Gråbrødre Torv i Middelalderbyen. Første gang er tirsdag d. 9. juni kl 20:00.

Ja, hvis I har lyst til at lave det samme arrangement i andre storbyer, så fortæl redaktionen det, og det brede sig som en løbeild...

Nyheder og rygter

AT&T har frigivet UNIX System V release 3.1. Nyt i denne udgave er bl.a. support for 8-bits nationale tegnsæt.

IBM har frigivet en ny serie af PC-er, Personal System/2. De bygger på en ny, hurtigere bus, og der er en Intel 80386 baseret model som toppen af serien. Samtidigt er PC-DOS 3.3 frigivet. IBM har annonceret et nyt operativsystem til PS/2 serien, nemlig OS/2 til frigivelse i 1988. Gad vide hvad OS/2 ligner?

Leverandørgruppen

Leverandørgruppen har holdt 2 møder siden sidste nummer. Det væsentligste punkt på dagsordenen var Mikrodata 87 udstillingen i Bella Centret, som blev arrangeret, og det med stor succes. Standen var velbesøgt, bortset fra de tidlige formiddagstimer, og der blev etableret mange nye kontakter, og gamle kontakter blev styrket. Firmaerne ICL, Ericsson, Altos, Bigum og Stenfors, DDD - Danmarks Ingeniørakademi, Data Reforming, NCR samt Metric bidrog med et tilskud til standen. Tak for det!

Der skal også lyde en stor tak til dem, der ville stille op og tale foreningens sag: René Seindal, Kim R. Pedersen, Lars Povlsen, Torben Nielsen, Lars Lassen, Niels Schachtschabel (alle fra Datalogisk Institut, Københavns Universitet), Keld Simonsen (sagde hunden), Thomas Nicolaisen, Arne Facius, Jesper Lauritsen (alle fra Center for Anvendt Datalogi, Københavns Universitet) Hanne Andersen, Kim Biel-Nielsen (begge Metric), Isak Korn (Ericsson) og Niels Svenningsen (Niels Svenningsen Aps). Ja det ser ud til at Københavns Universitet er dem der er mest på barrikaderne for at sælge DKUUG!

En anden ting der blev besluttet på leverandørmøderne var at lave et produktkatalog, efter skabelon fra AT&T. Dette er bestyrelsen således i fuld sving med udarbejdelsen af spørgeskemaet.

Medlemslisten

Foreningen har haft en kraftig medlemstilgang i den seneste tid, blandt andet som følge af vores stand på Mikrodata '87 i Bella Centret. Pr. 1. april var vi 160 organisations- og 19 individuelle medlemmer. Vi har altså haft en nettotilvækst på 22 medlemmer siden udsendelsen af sidste totale medlemsliste pr. 1986-12-14.

Med dette nummer af DKUUG Nyt følger derfor et supplement pr. 1987-04-01 til medlemslisten. Det erstatter alle tidligere supplement, hvis du skulle være i besiddelse af sådanne. På supplementets bagside findes "Nøgle til medlemslisten, sorteret efter navn", som erstatter alle tidligere udgaver af nøglen.

Hvis du mangler en medlemsliste pr. 1986-12-14, kan du rekvirere den hos undertegnede.

Hvis du har ændringer til medlemslisten/supplementet, bedes du ligeledes henvende dig til undertegnede, gerne skriftligt.

Medlemslisten må ikke bruges kommercielt, altså til udsendelser af reklamemateriale. I stedet kan reklamemateriale

vedlægges DKUUG Nyt mod en beskedens betaling. Hvis næste nummer af DKUUG Nyt ikke kan afventes, kan bestyrelsen tillade, at man køber et sæt adresseetiketter. I et sådant sæt mangler de medlemmer, der har meddelt, at oplysningerne fra indmeldingsblanketten ikke må udleveres til brug kommercielt.

Mogens Buhelt Kabbelejevej 27 B, 2700 Brønshøj. (evt. tel 02 86 55 33 i dagtimerne).

Helsinki-konferencen

Som bekendt (?) afholder EUUG sin forårskonference i perioden 11. - 15. maj 1987 i Helsinki og på den Botniske Bugt. Brochure med tilmeldingsblanket er udsendt til "gamle" medlemmer (dog ikke ekstra navne på den danske adresseliste) med EUUG Newsletter Vol. 6 No. 3. Brochuren kan iøvrigt rekvireres hos undertegnede.

Mogens Buhelt

X/OPEN

2. udgave af X/OPEN Portability Guide er udsendt i januar 1987. Det drejer sig om 5 bind til en samlet pris af USD 150,-. Bestillingseddelen kan rekvireres hos undertegnede, og mit eksemplar af brochuren om guiden kan evt. lånes mod løfte om hurtig tilbagelevering.

Mogens Buhelt

PS: Der arbejdes på at få X/OPEN Portability Guide som medlemstilbud. - *Keld.*

Lars Povlsen: Diku (Datalogisk Institut) og nettet

Der er ved at ske en del ting med nettet, hvorfor det nok er påkrævet at jeg klarificerer nogle punkter.

Til reference vil jeg citere referatet fra seneste bestyrelsesmøde i dkuug. Så kan folk selv gøre deres overvejelser. (Jeg var ikke til stede.)

Uddrag af "Referat af DKUUG bestyrelsesmøde nr. 34 af 17/3-87"

Punkt 6., Netstatus.

Fra 01/05-87 vil DIKU ikke supportere nettet.

KS (Keld Simonsen) vil indkalde til møde med alle deltagere som kører på nettet.

-" Eventuel oprettelse af Net-gruppe.

(Uddrag slut.)

Forholdene på diku er følgende:

---- o ----

1) Diku's (Datalogisk institut's vax11/785) dropper nettet.

I stedet flyttes nettet over på en SUN workstation doneret af Eriksson. Dette er planeret til at ske torsdag d. 26/4, og vil forhåbentlig foregå uden at folk bør vide det, i hvertifælde vil det ske uden at folk bør ændre hverken adresser eller ting vedrørende opkald.

Nuværende host diku vil blive til freja, men al post til maskinen diku (som nu vil være en "spøgelsesmaskine") vil blive videresendt til freja i stedet.

Motivationen for dette skridt har været den langvarige overbelastning af instituttets VAX11/785, både hvad angår

CPU-kraft og diskplads. Det har i lang tid været den eneste løsning på DIKU's generelle ressourcemangel, hvis diku som organisation skulle drive nettet.

----- o -----

2) Diku (Datalogisk institut) overlader brugerkontakt til DKUUG.

I forbindelse med min (Lars Povlsen) orlov (Jeg skal være jens), overlades almindelig drift til Søren Jensen (sergej@freja), mens Datalogisk Institut ikke længere ser sig i stand til at yde hjælp med at slutte nye brugere til nettet og bistå med uddybninger vedrørende brug af elektronisk post og arkiv-service. Dette felt overlades til DKUUG, som jo er bruger-klubben, og må levere den service den selv annoncerer.

For kort at komme ind på motivationen for denne beslutning, skal det siges, at tilgangen til nettet har været stor inden for det sidste tidsrum, og det er jo i sig selv glædeligt. Men dermed er det blevet et ret tidskrævende hverv, idet der dels er selve opkoblingen, men osse bistand med relaterede spørgsmål om brug af elektronisk post og arkiv-service, men osse generelle spørgsmål og unix. Og selvfølgelig kan diku's edbafdeling være behjælpelig i en snæver vending, men det er ikke diku's ansvar at fungere som generel programmeringshjælp for dkuugs medlemmer. Og slet ikke når antallet af folk på nettet vokser til > 40 danske sites. Diku's edbafdeling har selv noget arbejde der skal fra hånden, faktisk.

Så alt i alt vil det ikke berøre nuværende brugere, idet diku (instituttet) jo stadig står for driften, og dermed spørgsmål vedrørende samme.

For yderligere at få alle facts på bordet, følger et uddrag af et brev sendt til piet@cwi.nl, euug network administrator.

:
Second, we are moving the backbone to a donated SUN workstation, to offload our poor, tiny VAX. This will be my last assignment here, and gives sergej a chance to see all installed once at least. This moving implies that our VAX, will be given a new name. Actually its an old name, it was originally sceduled to be called freja, which will now be effective.

:
(Third) My institute have announced, related to my leave, that it cannot support the (network) user contacts, and that dkuug must be responsible for this, since this soaks up a LOT of time, acting as a general purpose comp. departement for all the dkuug members. Dkuug must give this service, as well of hooking new sites up, and explaining them the mysterious ways of uucp mail and USENET. After all, dkuug IS the user group, taking the the credit, and announcing the service.

My institute will still be paying and collecting the bills, and deliver network uptime and sanity, through sergej.

I håb om en fredelig løsning uden dommedagsannonceringer og løse rygter,

--Lars

Keld Simonsen: Diku (Datalogisk Institut) og nettet

Bestyreren fra Datalogisk Institut, Stig Skelboe, har meddelt mig, at instituttets edb-afdeling har haft driften af DKUUGs net oppe til drøftelse, og at det var afdelingens beslutning at man fra 1. maj ikke længere ønskede at yde support til foreningens medlemmer med hensyn til opkoblinger til nettet og vejledning iøvrigt. Men instituttet var stadig indstillet på at yde "teknisk support" til nettet. Institutet anbefalede at DKUUG ansatte en person selv til at varetage al kontakt med udenfra kommende brugere af nettet.

Jeg har så forsøgt at konsultere Lars Povlsen som DKUUGs bestyrelsesmedlem med særligt ansvar for nettet, men han kunne kun henvise mig til forhandlinger med institutbestyrelsen. Han ville ikke påtage sig på DKUUGs vegne at løse problemerne på grund af manglende tid (han er indkaldt til aftjening af værnepligt fra 1. maj).

Jeg har derfor påtaget mig at løse problemet. Jeg har senere opsnuset at Datalogisk Institut har afsat Søren Jensen, sergej@diku til den tekniske support. Dette betyder at der ikke er umiddelbare problemer for dem der for nuværende er på nettet, og der derfor ikke er nogen hastesituation.

Det er selvfølgelig beklageligt at Datalogisk Institut ikke længere ønsker kontakt fra netbrugerne, men da instituttet har udført dette arbejde gratis, er der kun grund til at takke instituttet for dets indsats indtil nu, og iøvrigt påskønne tilbudet om teknisk support fremover.

Jeg forsøger i første omgang at realisere den af Datalogisk Institut foreslåede løsning. Den vil give mindst gene for brugerne. Der vil være nogle øgede omkostninger til aflønning af en person, og disse beløb forestiller jeg mig indkrævet ved et særligt opkoblingsgebyr ved første tilmelding til nettet på højst 1000 kr, samt med en overheadsafgift på de øvrige tjenester. Omsætningen på nettet er i øjeblikket ca 200.000 kr årligt og belastningen for pasningen af nettet er ca 20 timer ugentligt, hvoraf en stor del er den tekniske support. Jeg forudser derfor en stigning i priserne på 15 - 30 % - hvilket vel ikke er overvældende. Dette vil også sige - efter Ericssons donation af SUN-2/120 maskinen, samt de iøvrigt donerede modems mv., at nettet ikke er subsidieret og der derfor ikke skulle komme flere større overraskelser foreløbigt.

Ifald jeg ikke kan få kabalen til at gå op - der skal laves en aftale og ansættes en person - på Datalogisk Institut, har jeg positive reaktioner fra andre større netbrugere på at de ville kunne overtage backbone-funktionen. Jeg er selv indstillet på i en overgangsperiode efter 1. maj at yde hjælp med opkobling i det omfang Datalogisk Institut ikke vil gøre det. Iøvrigt er jeg ikke længere ansat på Datalogisk Institut, idet jeg har sagt op pr 1. april pga samarbejdsproblemer (der ikke principielt har med denne sag at gøre). Men jeg kan stadig træffes på min mailboxadresse keld@diku (diku!keld for folk med gamle mailere), telefon 01 12 01 15.

Iøvrigt vil jeg benytte anledningen hvor der er nogle problemer at snakke om til at indkalde til et møde:

Onsdag den 22. april kl 19:00
Center for Anvendt Datalogi
Stuiestræde 6 o.g.
1455 København K

hvor jeg vil forslå at vi diskuterer ovennævnte problemer, samt fremtidige aktiviteter og services inden for netværksområdet, og hvor vi måske også kan inddrage distribution af programmel. Jeg kunne tænke mig at vi på denne måde lavede en netværksgruppe i foreningen. Onsdag den 22. er valgt fordi foreningen iøvrigt holder et eftermiddagsmøde der hos IBM i Lyngby.

Keld Simonsen

Red. (det er også mig): Det omtalte møde hos IBM d. 22. april blev udsat til 7. maj, jfr. annoncering andetsteds i dette blad.

Møde om nettets fremtid

Op der blev holdt et møde d. 22. april, som omtalt ovenfor. Der var 10 folk tilstede, Søren Jensen og Niels Schachtschabel fra DIKU, Sys Sidenius fra Altos, Jens Trier fra KTAS, Peder Chr. Nørgaard fr DAIMI, Isak Korn og Jan Hauge Andersen fra Ericsson, Kim Chr. Madsen og Kim Storm fra Ambrasoft samt Keld Simonsen fra Center for anvendt Datalogi og DKUUGs bestyrelse.

Keld Simonsen redegjorde for nettets øjeblikkelige problemer, nogenlunde som omtalt i de to foregående artikler. Han fortalte at han havde talt med Datalogisk Instituts bestyrer om en skriftlig samarbejdsaftale, som der skulle være gode muligheder for at få igennem. DIKU ville stille teknisk support gratis til rådighed, mens DKUUG skulle stå for brugerkontakten. Økonomisk skulle det betyde, at DKUUG overtog hele det økonomiske ansvar og ansatte en person til brugerkontakten. Det ansloges at der var brug for en halvtidsperson, og at de øgede udgifter ville beløbe sig til ca. 100.000 kr på årsbasis. Man drøftede følgende budget på årsbasis:

mail transmissionskonteringer	130.000
news 8 * 8000	70.000
mail opkoblingsafgift, 1000 * 10	10.000
mail abonnement, 500 * 40	20.000
news opkobling, 1000 * 5	5.000
news abonnement, 1000 * 10	10.000
trafikafgift, mail&news 100 kr/Mb * 700 Mb	70.000
<hr/> Indtægter	<hr/> 315.000
Transmissionsudgifter	200.000
Løn	100.000
Reparationer, indkøb	15.000
<hr/> Udgifter	<hr/> 315.000

Herudover skulle der betales efter regning hvis opkobling krævede besøg på åstedet. Man mente at det var op til DKUUGs bestyrelse at fastsætte de nøjagtige priser. Man fandt at archive-servicen kunne være til nytte for nogen, men mente at der skulle tages ekstra pris for denne service.

Peder Chr. lovede at stå for et udkast samt redigeringen af en vejledning for nye sites, med forskelle i uucp versioner belyst. Kim Chr. Madsen lovede at skrive en artikel til bladet om de nye grupper i news. Isak Korn lovede at komme med en 130 Mb disk ekstra til SUN-2/120 maskinen. Man betonedede ønskeligheden i at post blev afleveret så hurtigt som muligt, helst med det samme. DKUUGs nye ansatte skulle kun besvare net-spørgsmål, ikke generelle forespørgsler, og der skulle være en fast, begrænset telefontid. Man ville gerne fungere som referencegruppe for DKUUG, men ville helst overlade beslutningerne til bestyrelsen. Man ville gerne have at nettet fortsat kørte i DKUUGs regie, da dette var et af foreningens fremmeste formål, og man ellers bare måtte oprette en tilsvarende forening.

Alt i alt en udbytterig aften i en god stemning, og nogle fortsatte på en nærliggende bar.

Andre UNIX-net nyheder

DKUUG har fået stillet en SUN-2/120 datamat med 130 Mb disk til rådighed fra Ericsson Information Systems A/S, og den er ved at blive kørt ind som netknode på DIKU. Ericsson har lovet at komme med endnu en 130 Mb disk til den.

Det nye conferencebånd til EUUGs Helsinki-konference er nu tilgængeligt via archive på DIKU. Det ligger i kataloget EUUGD11 på DIKU, og det meste af hvad der lå i kataloget euug.d8.3 er samtidigt fjernet. EUUGD11 indeholder GNU software, (herunder GNU emacs 18.41, som måske overføres bedst med uucp direkte, se evt. besked i README i relevant katalog), UNIX|STAT 5.3 samt en kollektion af programmel taget fra nettet lavet af Björn Erikson fra den svenske gruppe. Se artiklen i DKUUG NYT nr 11 for hvordan man kan få fat i programmet via nettet, ellers kan det snart bestilles fra Amsterdam (Bånd EUUGD11).

MINIX: A CHEAP UNIX CLONE WITH SOURCE CODE

Andrew S. Tanenbaum

Dept. of Mathematics and Computer Science

Vrije Universiteit

Amsterdam, The Netherlands

1. OVERVIEW OF THE MINIX SYSTEM ARCHITECTURE

UNIX is organized as a single executable program that is loaded into memory at system boot time and then run. MINIX is structured in a much more modular way, as a collection of processes that communicate with each other and with user processes by sending and receiving messages. There are separate processes for the memory manager, the file system, for each device driver, and for certain other system functions. This structure enforces a better interface between the pieces. The file system cannot, for example, accidentally change the memory manager's tables because the file system and memory manager each have their own private address spaces.

These system processes are each full-fledged processes, with their own memory allocation, process table entry and state. They can be run, blocked, and send messages, just as the user processes. In fact, the memory manager and file system each run in user space as ordinary processes. The device drivers are all linked together with the kernel into the same binary program, but they communicate with each other and with the other processes by message passing.

When the system is compiled, four binary programs are independently created: the kernel (including the driver processes), the memory manager, the file system, and *init* (which reads */etc/tty*s and forks off the login processes). In other words, compiling the system results in four distinct *a.out* files. When the system is booted, all four of these are read into memory from the boot diskette.

It is possible, and in fact, normal, to modify, recompile, and relink, say, the file system, without having to relink the other three pieces. This design provides a high degree of modularity by dividing the system up into independent pieces, each with a well-defined function and interface to the other pieces. The pieces communicate by sending and receiving messages.

The various processes are structured in four layers:

4. The user processes (top layer).
3. The server processes (memory manager and file system).

2. The device drivers, one process per device.
1. Process and message handling (bottom layer).

Let us now briefly summarize the function of each layer.

Layer 1 is concerned with doing process management including CPU scheduling and interprocess communication. When a process does a SEND or RECEIVE, it traps to the kernel, which then tries to execute the command. If the command cannot be executed (e.g., a process does a RECEIVE and there are no messages waiting for it), the caller is blocked until the command can be executed, at which time the process is reactivated. When an interrupt occurs, layer 1 converts it into a message to the appropriate device driver, which will normally be blocked waiting for it. The decision about which process to run when is also made in layer 1. A priority algorithm is used, giving device drivers higher priority over ordinary user processes, for example.

Layer 2 contains the device drivers, one process per major device. These processes are part of the kernel's address space because they must run in kernel mode to access I/O device registers and execute I/O instructions. Although the IBM PC does not have user mode/kernel mode, most other machines do, so this decision has been made with an eye toward the future. To distinguish the processes within the kernel from those in user space, the kernel processes are called tasks.

Layer 3 contains only two processes, the memory manager and the file system. They are both structured as servers, with the user processes as clients. When a user process (i.e., a client) wants to execute a system call, it calls, for example, the library procedure *read* with the file descriptor, buffer, and count. The library procedure builds a message containing the system call number and the parameters and sends it to the file system. The client then blocks waiting for a reply. When the file system receives the message, it carries it out and sends back a reply containing the number of bytes read or the error code. The library procedure gets the reply and returns the result to the caller in the usual way. The user is completely unaware of what is going on here, making it easy to replace the local file system with a remote one.

Layer 4 contains the user programs. When the system comes up, *init* forks off login processes, which then wait for input. On a successful login, the shell is executed. Processes can fork, resulting in a tree of processes, with *init* at the root. When CTRL-D is typed to a shell, it exits, and *init* replaces the shell with another login process.

2. LAYER 1 - PROCESSES AND MESSAGES

The two basic concepts on which MINIX is built are processes and messages. A process is an independently schedulable entity with its own process table entry. A message is a structure containing the sender's process number, a message type field, and a variable part (a union) containing the parameters or reply codes of the message. Message size is fixed, depending on how big the union happens to be on the machine in question. On the IBM PC it is 24 bytes.

Three kernel calls are provided:

- RECEIVE(source, &message)
- SEND(destination, &message)
- SENDREC(process, &message)

These are the only true system calls (i.e., traps to the kernel). RECEIVE announces the willingness of the caller to accept a message from a specified process, or ANY, if the RECEIVER will accept any message. (From here on, "process" also includes the tasks.) If no message is available, the receiving process is blocked. SEND attempts to transmit a message to the destination process. If the destination process is currently blocked trying to receive from the sender, the kernel copies the message from the sender's buffer to the receiver's buffer, and then marks them both as runnable. If the receiver is not waiting for a message from the sender, the sender is blocked.

The SENDREC primitive combines the functions of the other two. It sends a message to the indicated process, and then blocks until a reply has been received. The reply overwrites the original message. User processes use SENDREC to execute system calls by sending messages to the servers and then blocking until the reply arrives.

There are two ways to enter the kernel. One way is by the trap resulting from a process' attempt to send or receive a message. The other way is by an interrupt. When an interrupt occurs, the registers and machine state of the currently running process are saved in its process table entry. Then a general interrupt handler is called with the interrupt number as parameter. This procedure builds a message of type INTERRUPT, copies it to the buffer of the waiting task, marks that task as runnable (unblocked), and then calls the scheduler to see who to run next.

The scheduler maintains three queues, corresponding to layers 2, 3, and 4, respectively. The driver queue has the highest priority, the server queue has middle priority, and the user queue has lowest priority. The scheduling algorithm is simple: find the highest priority queue that has at least one process on it, and run the first process on that queue. In this way, a clock interrupt will cause a process switch if the file system was running, but not if the disk driver was running. If the disk driver was running, the clock task will be put at the end of the highest priority queue, and run when its turn comes.

In addition to this rule, once every 100 msec, the clock task checks to see if the current process is a user process

that has been running for at least 100 msec. If so, that user is removed from the front of the user queue and put on the back. In effect, compute bound user processes are run using a round robin scheduler. Once started, a user process runs until either it blocks trying to send or receive a message, or it has had 100 msec of CPU time. This algorithm is simple, fair, and easy to implement.

3. LAYER 2 - DEVICE DRIVERS

Like all versions of UNIX for the IBM PC, MINIX does not use the ROM BIOS for input or output because the BIOS does not support interrupts. Suppose a user forks off a compilation in the background and then calls the editor. If the editor tried to read from the terminal using the BIOS, the compilation (and any other background jobs such as printing) would be stopped dead in their tracks waiting for the the next character to be typed. Such behavior may be acceptable in the MS-DOS world, but it certainly is not in the UNIX world. As a result, MINIX contains a complete set of drivers that duplicate the functions of the BIOS. Like the rest of MINIX, these drivers are written in C, not assembly language.

This design has important implications for running MINIX on PC clones. A clone whose hardware is not compatible with the PC down to the chip level, but which tries to hide the differences by making the BIOS calls functionally identical to IBM's will not run an unmodified MINIX because MINIX does not use the BIOS.

Each device driver is a separate process in MINIX. At present, the drivers include the clock driver, terminal driver, various disk drivers (e.g., RAM disk, floppy disk), and printer driver. Each driver has a main loop consisting of three actions:

1. Wait for an incoming message.
2. Perform the request contained in the message.
3. Send a reply message.

Request messages have a standard format, containing the opcode (e.g., READ, WRITE, or IOCTL), the minor device number, the position (e.g., disk block number), the buffer address, the byte count, and the number of the process on whose behalf the work is being done.

As an example of where device drivers fit in, consider what happens when a user wants to read from a file. The user sends a message to the file system. If the file system has the needed data in its buffer cache, they are copied back to the user. Otherwise, the file system sends a message to the disk task requesting that the block be read into a buffer within the file system's address space (in its cache). Users may not send messages to the tasks directly. Only the servers may do this.

MINIX supports a RAM disk. In fact, the RAM disk is always used to hold the root device. When the system is booted, after the operating system has been loaded, the user is instructed to insert the root file system diskette. The file system then sees how big it is, allocates the necessary memory, and copies the diskette to the RAM disk. Other file systems can then be mounted on the root device.

This organization puts important directories such as */bin* and */tmp* on the fastest device, and also makes it easy to work with either floppy disks or hard disks or a mixture of the two by mounting them on */usr* or */user* or elsewhere. In any event, the root device is always in the same place.

In the standard distribution, the RAM disk is about 240K, most of which is full of parts of the C compiler. In the 256K system, a much smaller RAM disk has to be used, which explains why this version has no C compiler: there is no place to put it. (The */usr* diskette is completely full with the other utility programs and one of the design goals was to make the system run on a 256K PC with 1 floppy disk.) Users with an unusual configuration such as 256K and three hard disks are free to juggle things around as they see fit.

The terminal driver is compatible with the standard V7 terminal driver. It supports cooked mode, raw mode, and cbreak mode. It also supports several escape sequences, such as cursor positioning and reverse scrolling because the screen editor needs them.

The printer driver copies its input to the printer character for character without modification. It does not even convert line feed to carriage return + line feed. This makes it possible to send escape sequences to graphics printers without the driver messing things up. MINIX does not spool output because floppy disk systems rarely have enough spare disk space for the spooling directory. Instead one normally would print a file *f* by saying

```
lpr <f &
```

to do the printing in the background. The *lpr* program insert carriage returns, expands tabs, and so on, so it should

only be used for straight ASCII files. On hard disk systems, a spooler would not be difficult to write.

4. LAYER 3 - SERVERS

Layer 3 contains two server processes: the memory manager and the file system. They are both structured in the same way as the device drivers, that is a main loop that accepts requests, performs them, and then replies. We will now look at each of these in turn.

The memory manager's job is to handle those system calls that affect memory allocation, as well as a few others. These include FORK, EXEC, WAIT, KILL, and BRK. The memory model used by MINIX is exceptionally simple in order to accommodate computers without any memory management hardware. When the shell forks off a process, a copy of the shell is made in memory. When the child does an EXEC, the new core image is placed in memory. Thereafter it is never moved. MINIX does not swap or page.

The amount of memory allocated to the process is determined by a field in the header of the executable file. A program, *chmem*, has been provided to manipulate this field. When a process is started, the text segment is set at the very bottom of the allocated memory area, followed by the data and bss. The stack starts at the top of the allocated memory and grows downward. The space between the bottom of the stack and the top of the data segment is available for both segments to grow into as needed. If the two segments meet, the process is killed.

In the past, before paging was invented, all memory allocation schemes worked like this. In the future, when even small microcomputers will use 32-bit CPUs and 1M x 1 bit memory chips, the minimum feasible memory will be 4 megabytes and this allocation scheme will probably become popular again due to its inherent simplicity. Thus the MINIX scheme can be regarded as either hopelessly outdated or amazingly futuristic, as you prefer.

The memory manager keeps track of memory using a list of holes. When new memory is needed, either for FORK or for EXEC, it searches the hole list and takes the first hole that is big enough (first fit). When a process terminates, if it is adjacent to a hole on either side, the process' memory and the hole are merged into a bigger hole.

The file system is really a remote file server that happens to be running on the user's machine. However it is straightforward to convert it into a true network file server. All that needs to be done is change the message interface and provide some way of authenticating requests. (In MINIX, the source field in the incoming message is trustworthy because it is filled in by the kernel.) When running remote, the MINIX file server maintains state information, like RFS and unlike NFS.

The MINIX file system is similar to that of V7 UNIX. The i-node is slightly different, containing only 9 disk addresses instead of 13, and only 1 time instead of 3. These changes reduce the i-node from 64 bytes to 32 bytes, to store more i-nodes per disk block and reduce the size of the in-core i-node table.

Free disk blocks and free inodes are kept track of using bit maps rather than free lists. The bit maps for the root device and all mounted file systems are kept in memory. When a file grows, the system makes a definite effort to allocate the new block as close as possible to the old ones, to minimize arm motion. Disk storage is not necessarily allocated one block at a time. A minor device can be configured to allocate 2, 4 (or more) contiguous blocks whenever a block is allocated. Although this wastes disk space, these multiblock zones improve disk performance by keeping file blocks close together. The standard parameters for MINIX as distributed are 1K blocks and 1K zones (i.e., just 1 block per zone).

MINIX maintains a buffer cache of recently used blocks. A hashing algorithm is used to look up blocks in the cache. When an i-node block, directory block, or other critical block is modified, it is written back to disk immediately. Data blocks are only written back at the next SYNC or when the buffer is needed for something else.

The MINIX directory system and format is identical to that of V7 UNIX. File names are strings of up to 14 characters, and directories can be arbitrarily long.

5. LAYER 4 - USER PROCESSES

This layer contains *init*, the shell, the editor, the compiler, the utilities, and all the user processes. These processes may only send messages to the memory manager and the file system, and these servers only accept valid system call requests. Thus the user processes do not perceive MINIX to be a general-purpose message passing system. However, removing the one line of code that checks if the message destination is valid would convert it into a much more general system (but less UNIX-like).

