

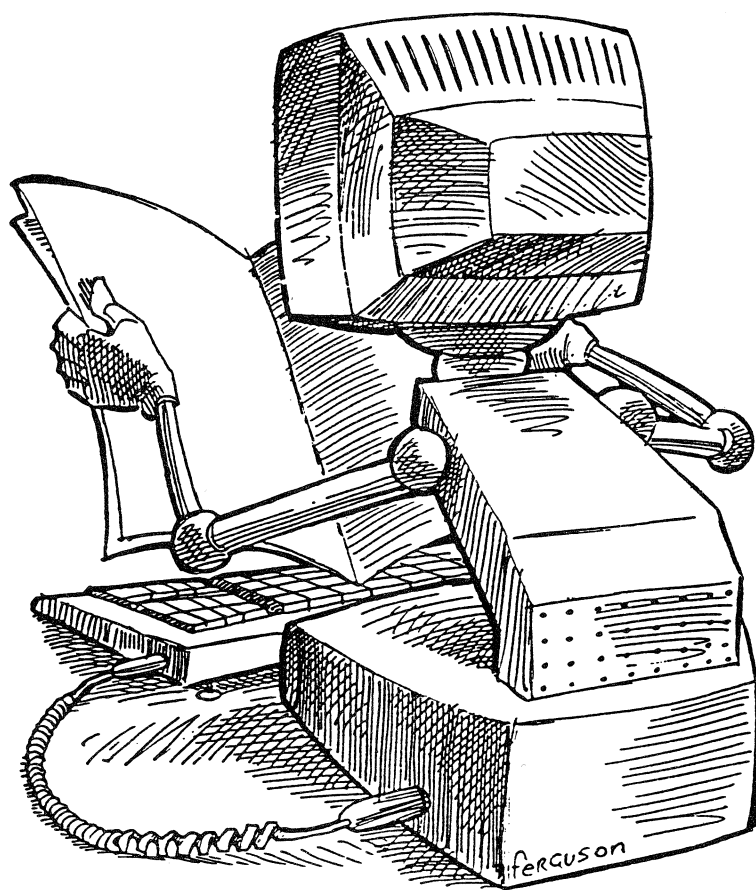
RUN-NYTT

Datainformasjonsorgan for UNIT og SINTEF
Utgitt av SINTEF RUNIT

Nr. 4

15 desember 1991

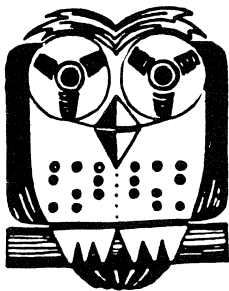
ÅRG 18



RUN-NYTT redaksjonen ønsker sine lesere

EN GOD JUL OG ET GODT NYTT ÅR!

Tjener ugle.unit.no



På UNITs programvare og informasjonstjener ugle.unit.no er det pr.1/12 7800 filer fordelt på 816 kataloger. Disse er fordelt på 5 katalogtrær - pub/ (799, 7497), info/ (12, 282), local/ (3, 5) archives/ (1, 2), og ftplists/ (1, 14). Tallene i parantesene er henholdsvis antall kataloger og antall filer i hvert katalogtre.

Antall filer hentet så langt i 1991 er 46891, hvorav 20410 fra Trondheim (IP adresser 129.241...)

Hvordan finne fram i tilbudet:

I mange av katalogene er det en fil LESMEG (norsk) eller README (engelsk) som forteller om katalogens innhold.

På det nivået en kommer inn med "anonym FTP", er det følgende filer en kan starte med:

LESMEG
 Introduksjonsfil.
KATALOGER.TRE
 En visuell oversikt over katalogstrukturen.
 Laget med programmet vtree.
all.dirs
 Alle kataloger. Laget med: find . -type d -print
all.files.
 Alle filer - bare filnavn. Laget med:
 find . -type f -print
filliste.total
 Alle filer - også størrelse og dato. Laget med:
 ls -lR. Stor fil!

Alle katalog- og filoversikter blir oppdatert hver natt.

Ftp tips

Vanlig bruk av kommando get legger det en henter i en fil på den maskinen en starter ftp fra - generell syntaks er get frafil tilfil. Dette gjelder også LESMEG filer og filoversiktene.

Hvis en, uten å avslutte ftp forbindelsen, vil se på disse overførte oversiktsfilene før en bestemmer seg for filene en så vil overføre, må en i ftp starte en kommando for det på egen maskin med !kommando - f.eks. !more LESMEG.

Det kan isteden være like enkelt å sende innholdet i slike filer direkte til terminalen for rask inspeksjon. Dette kan gjøres slik:

Alt skrives ut fortløpende:
UNIX: get frafil -
VAX/VMS: get frafil tt:

Merk at for en PC bruker som kjører terminaltrafikk mot en tjenermaskin, og som har "logging" påslått i sitt terminalprogram, kan dette være nyttig for alle (mindre) tekstfiler som hentes. Filinnholdet kommer rett ned på PCen.

Stopp for hver full skjerm-side:

UNIX: get frafil | more

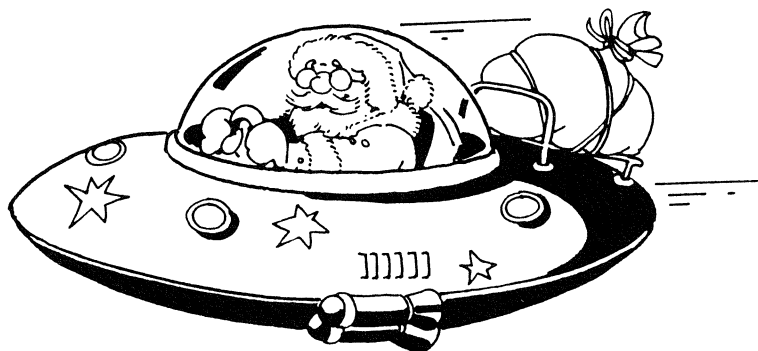
Generelt vil | først i "tilfil" navnet tolkes slik at det som kommer bak er en UNIX kommando. Et annet eksempel er utskrift direkte til skriver skr: get filnavn "|lpr -Pskr". Her er det mellomrom i teksten etter |, derfor må en bruke hermetegn.

Det hender ofte at det er så mange filer i en katalog at bruk av kommandoen ls eller dir i ftp fører til flere linjer utskrift enn er det plass til på en skjerm-side. Da forsvinner de første navnene.

Generell syntaks for ls (og dir) kommandoen i ftp er: ls katalognavn tilfil. En kan lagre fillista direkte på filen tilfil på egen maskin. Men en må oppgi katalognavnet på fjern maskin for å få dette til.

Hvis en vil gjøre dette for den katalogen en har plassert seg i der, setter en et punktum som katalognavn hvis fjern maskin er en UNIX maskin eller en IBM maskin (ls . tilfil.), og *.* hvis maskinen er en VAX/VMS maskin (ls *.* tilfil). Hvis en vil lese fillista direkte på terminalen med more skriver en ls . |more evt ls *.* |more.

Knut L Vik



RUN-NYTT

Adresse: RUNIT
7034 Trondheim

E-post adresse knut.vik@sintef.no
C=no; P=uninett; O=sintef;
S=vik; G=knut;

Redaksjon: Knut L. Vik
Tlf. 07 593047
Anne B. Reitan Sivertsen
Tlf. 07 593027

Utgivelse: 4 nummer pr. år

Abonnement: Gratis ved henvendelse
til RUNITs ekspedisjoner
eller redaksjonen

Opplag 1800

Trykkeri: Nidaros Trykkeri, Trondheim

RUN-NYTT er produsert med Pagemaker
Skrifttype: Bookman 10 pkt

Stoff til RUN-NYTT mottas med takk

***Bruk gjerne artikler fra RUN-NYTT,
men oppgi kilde!***

GODT NYTT ÅR

RUN-NYTTs redaksjon ønsker alle leserne godt nytt år, og vi håper dere har funnet noe nyttig i de 4 nummerne som er utgitt i 1991.

Som dere ser på forsiden, er RUN-NYTT et data-informasjonsorgan for UNIT og SINTEF - når det gjelder BRUK av databehandling. Vi ser det som vårt mål å samle tråder, lage oversikter, fortelle om nyheter, senke brukerterskler.

Vi søker å ha et samspill mellom den trykte informasjonen i RUN-NYTT, og informasjon og programvare lagret på tjenermaskinen ugle.unit.no. Gamle RUN-NYTT artikler kan hentes fra tjeneren, og vi henviser til filer med programvare og utfyllende lesestoff.

Vi ser på informasjonstjenesten og brukerstøtte generelt som en viktig tjeneste. Databehandling er for de fleste studenter, undervisere og forskere et verktøy, og verktøyutvalget må være bra og kjent, samt greit å ta i bruk. Disse tjenestene er også en del av det å effektivt utnytte den økonomiske investeringen i utstyr og programvare.

Gode og lett tilgjengelige faglige dataverktøy og gode støttetjenester for disse er med å gi UNIT og SINTEF et faglig konkurransefortrinn!

Vi tror at det å satse på "bruk og brukere" er rett og viktig, og at det vil bli mer og mer viktig etterhvert.

Knut L Vik

INNHold

Tjener ugle.unit.no	s. 2	Databaser om forskningsprosjekter	s. 16
Sending av binærfiler med e-post	s. 4	Bruk BIBSYS fra eget kontor	s. 18
UNIT får ny IT-organisasjon	s. 5	UNIX - diverse	s. 21
Studentarbeidsstasjonene er populære!	s. 5	NetNews diskusjonsgrupper	s. 22
Nye lisensavtaler	s. 6	Parallele datamaskiner på	
MAPLE	s. 7	Hyperkubelaboratoriet ved IDT	s. 23
Eksempler på bruk av MAPLE	s. 8	Supernet	s. 28
MATHCAD - det ideelle ingeniørverktøyet	s. 10	STATLIB - programbibliotek	
WordPerfect sikkerhetskopi	s. 13	for statistikkprogram	s. 28
Nytt fra DSL-UNIT	s. 14		



Sending av binærfiler med e-post

E-post er i utgangspunktet laget for å sende tekst, og vi får problemer når vi skal sende filer hvor alle 8 bitene i en byte har betydning for innholdet. Det kan gjelde programfiler, en WordPerfect fil med redigeringskommandoer, en fil hvor æ, ø og å er kodet vha. den 8 biten, etc.

Hvor store problemene blir, avhenger av hva en sender, hvem en sender e-post meldingen til, og hva meldingen passerer på veien. Det kan f.eks gå bra med æ, ø og å hvis en sender en tekst fra en PC til en PC, men ikke fra en PC til en Macintosh. Da må en konvertere teksten før eller etter sending.

Minste felles multiplum for alle postsystemer og alle maskintyper er 7 bits ASCII koding - den 8. biten brukes ikke. Så lenge en sender filer hvor alt meningsinnhold er kodet med 7 biter, vil forsendelsen gå bra. Men det utelukker sending av binærfiler av type programfiler eller WordPerfect filer direkte slik de er.

Det er laget løsninger på dette - ideen er å kode om en binærfil til en tekstfil inneholdende tegn kodet i 7 bits ASCII kode. Dette resultatet sendes pr. e-post, og hos mottaker dekodes teksten tilbake til en binærfil.

Det finnes flere program for dette - den mest vanlige måten å gjøre dette på er UUENCODE/UUDECODE koding. Alle tegn som brukes ved kodingen er skrivbare tegn, så en slik fil kan lett legges inn i en e-post melding. Kodingen øker filstørrelsen med ca. 30%.

For å utføre UUkoding og -dekoding finnes det egne program. Slike er det f.eks. for både UNIX, MSDOS, Mac og VAX/VMS. Men det som er meget interessant og nyttig, er at slik koding er lagt inn i noen e-post program, og er tilgjengelig gjennom kommandoer i disse programmene. Det gjelder for de to programmene vi beskrev i forrige RUN-NYTT - Lifeline Mail og Pegasus Mail. Det første brukes på PCer i PC-NFS nett, og det andre kan brukes i Novell nett.

I disse programmene er det meget enkelt å ta dette i bruk. Det som er nyttig, er at de filer som kodes kan sendes til enhver maskin hvor det finnes en dekodemulighet, enten i det motakende postprogram eller vha. et eget program.

Derfor er det mange som kan utnytte dette.

Et nyttig bruksområde er sending av WordPerfect filer med redigeringskommandoer - en trenger ikke lagre en slik fil som en DOS fil og så sende innholdet!

UNIX Mail (SMTP) er basert på å bruke 7 biter i en byte - den 8. biten kastes bare bort. Når en UNIX maskin er en server i et PC nett, og vi vil bruke e-post til å sende til en annen PC på nettet en tekst med æ, ø og å, får vi problemer. På PCene brukes 8 biter til koding av æ, ø og å, og dette ødelegges på veien fram ved at meldingen sendes via UNIX Mail på serveren. Dette opplever vi med Lifeline Mail i et PC-NFS nett. En løsning er å benytte Uuencode og Uudecode i Lifeline Mail også i dette tilfellet!

Første linje i en kodet fil starter med begin og i den siste linja står det end. Mellom disse linjene er resultatet av kodingen. Alle andre linjer i den meldingen en mottar, må oftest fjernes før Uuencode kan foreta dekodningen. Dog - Lifeline Mail ser ut til å takle direkte en melding med konvolutt delen uslettet.

Uencode og uudecode leveres som del av standard UNIX på de fleste UNIX maskiner - f.eks. for SUN, DEC Ultrix og HP. Se man uencode og man uudecode. Utgaver for MSDOS leveres med PC-NFS.

Vår lokale programvare tjener ugle.unit.no inneholder kildekoden til uencode og uudecode (skrevet i C) i underkataloger til katalog pub/unix/bsd-sources/usr.bin. Programmene for OS2 finnes i katalog pub/os2, og for MSDOS i katalog pub/msdos/network.

PC programvare som sendes via NetNews er ofte kodet med uencode, og ofte er et større program splittet i flere filer som må settes sammen før behandling med uudecode. Det finnes små program som ordner dette automatisk (uud, uncat).

Filene som er UUkodet inneholder spesielle tegn som hakeparanteser. Disse tegnene kan kanskje bli endret ved passering av maskiner.

En mulighet er når meldingen passerer EARN/BITNET. Det finnes en kodemetode som heter xxencode og xxdecode og som kan løse dette problemet - da brukes bare bokstaver, tall og pluss og minus ved kodingen.

For videre lesing om dette anbefales artikkelen:

Eric Neuwirth: "A short guide to networking and file transmission".

Artikkelen kan hentes fra fil info/nettinfo/sending.files i tjener ugle.unit.no, og fra UNINETTINFO i katalog brukerhjelp og fil sending.files.

For Macintosh finnes det et annet og mye brukt format for koding til tekstfiler - BinHex. Se artikkelen "Komprimerte og pakkefile" i RUN-NYTT nr. 3, 1991, s 31.

Knut L Vik

UNIT får ny IT-organisasjon

Onsdag den 27. november vedtok Kollegiet hovedprinsippene for ny administrasjonsordning for universitetet. På IT-siden innebærer vedtaket at UNITs ulike EDB-/ADB-/IT-grupper (you name it) med driftsansvar samles administrativt under en felles ledelse, en IT-sjef, som igjen skal sortere direkte under administrerende direktør. Fakultetenes/avdelingenes egne EDB-konsulenter skal imidlertid fortsatt være ved sine respektive enheter.

Dette innebærer en ny situasjon for alle oss som har EDB som ansvarsområde, enten vi arbeider ved sentraladministrasjonene, fakultetene eller andre steder. Den framtidige IT-organisasjonen ("UN-IT", "UNIT-IT" ..?) er på ingen måte fastlagt i størrelse og fasong, men en del ting kan nok allerede nå sees i kortene.

Ny stilling som øverste leder for IT-organisasjonen lyses ut med det første. Den som tilsettes vil blant annet stå overfor oppgaven å få i gang nye former for samarbeid og arbeidsdeling, først og fremst innen vårt eget miljø, men også mot våre samarbeidspartnere som f.eks. SINTEF, Allforsk, UNINETT, BIBSYS osv – og selvfølgelig mot våre

kolleger ved andre universiteter og høyskoler.

At samarbeid fremheves så sterkt, er ut fra den kjensgjerning at utfordringene vi står overfor når det gjelder å tilby gode tjenester er langt større enn vi kan takle dersom vi fortsatt skal finne opp kruttet (les æ,ø og å) hver for oss også i fortsetningen.

Vi må bygge opp tjenester som virkelig kan fungere slik de bør for alle ansatte og studenter. Tenk på IT-tjenestene som kroppens blodkretsløp: veltrimmede lunger, hjerte, hovedårer, fine avgreiningene osv. Spesialiserte deler som fungerer som en helhet. Resten av universitetet er avhengig av dette kretsløpet. Derfor blir det nå satt i gang arbeid med å avklare hvordan kretsløpet skal formes videre. Arbeidet skal føre til felles strategier (les felles standarder) for grunnleggende tjenester som nettverk, kontorstøtte, elektronisk post, arkiv, administrative systemer osv.

Heldigvis går nå leverandørene i den retning som brukerne verden over har mast om i lengre tid, dvs. mot systemer som er åpne, ikke bare i navnet. Om ikke lenge vil det bli mulig å få utført de samme oppgavene enten vi har alliert oss med en PC, Mac eller UNIX arbeidsstasjon. Det ikke bare i Øst-Europa at hegemoniene faller! Vel og bra. Men det vil kreve kløkt og innsats av oss hvis dette skal fungere i praksis.

Det blir med andre ord mer enn nok av oppgaver for oss alle, vi må bare finne en god måte å fordele oppgavene på. Her vil det telle hva hver enkelt av oss mener at vi har å tilby andre, og hva vi skal videreutvikle oss på.

Eric Sandnes
fung. EDB-sjef for UNIT

Studentarbeids- stasjonene er populære!

Antall registrerte brukere på de 4 salene (Solan, Lise, Siri og Mari) er nå 2010. Antall nye brukere i høst er 740.

Som nevnt i forrige nummer, er det i høst utgitt informasjonsskrifter (Apropos-skrifter) til brukerne på salene.

Nye lisensavtaler

I en egen artikkel i dette RUN-NYTT skriver vi om avtalen for matematikkprogrammet Maple. Her vil vi nevne noen andre avtaler som RUNIT har inngått på vegne av UNIT og SINTEF:

1) OSF/MOTIF

En avtale for det grafiske brukergrensesnittet MOTIF er nå inngått for SINTEF. Tidligere er det inngått en tilsvarende avtale for internt UNIT bruk gjennom IDT.

Avtalen for SINTEF er en "Full Service" avtale som bl. annet omfatter oppdateringer og støtte. Versjonsnummeret er 1.1.3. Merk at en må ha ca. 200 MByte diskplass til disposisjon.

2) ADOBE/TRANSCRIPT

Dette er en samling UNIX program fra ADOBE for utskrift til PostScript skrivere. Programpakken inneholder utfiltere og oversettere for bl. annet troff, dtroff, UNIX plot, Tektronix 4014 og tekst filer.

Det er støtte for PostScript print filer laget av applikasjoner som Scribe og Tex, og for filer fra Macintosh og PC.

Det er "spooler interface and communication software" og "font metrics information and description files".

Avtalen gjelder for UNIT og SINTEF.

3) HP programvare

For UNIT og SINTEF er det inngått avtale med HP Norge om "Respons Line" vedlikeholdsavtale for driftsmiljøene HP UX og APOLLO Domain. Dette gjelder all tilgjengelig original HP og APOLLO programvare.

RUNIT vil ha ansvar for formidling av programvareoppdateringer og brukerstøtte mot de institutter og avdelinger som blir med på avtalen. Oppdateringene av programvaren og håndbøkene vil omfatte alle maskinplattformer (HP 9000 serie 300, 400, 700 og 800 samt APOLLO Domain), og vil kunne kopieres fritt til maskiner med lisens for den aktuelle programvaren.

4) SunOS programvare

Med Skrivervik Data er det for UNIT og SINTEF inngått en "Volume Pack" avtale for SunOS programvare.

Avtalen gjelder ca 20 program, bl. annet kompi-latorer og SunGKS. RUNIT vil stå for distribu-sjon og brukerstøtte.

For operativsystemet SunOS har det allerede eksistert en avtale i over et år.

5) Programvare for X terminaler fra Tandberg og NCD.

Denne avtalen er også for UNIT og SINTEF, og gjelder programvare for oppstart av disse X terminaltypene. "X-server", fonter, etc lastes ned til terminalene.

For mer informasjon om avtalene og produktene, samt pris for å delta, kontakt undertegnede.

Inge Dahl, RUNIT
Tlf. (59)6913
inge.dahl@runit.sintef.no



MAPLE

UNIT og SINTEF har nylig inngått "site avtale" for matematikkprogrammet MAPLE. Programmet kommer fra Canada, og er opprinnelig laget ved Waterloo University.

Med Maple kan du:

- utføre symbolsk (algebraisk) matematikk. Dette er det programmet er spesiallaget for.
- utføre numeriske beregninger.
- lage plot i 2D og 3D, og f.eks. få utskrift på en PostScript skriver
- få oversatt Maple uttrykk til C eller Fortran kode.
- få oversatt tekst og matematiske formler til Latex for utskrift.

Sammenlignbare program er Mathematica, Macsyma (symbolsk matematikk), Matlab (numeriske beregninger) og MathCAD.

Lisensen gjelder alle tilgjengelige maskintyper - viktigst her er MSDOS 386, Macintosh, VAX/VMS og diverse UNIX maskiner. Utgave for MS Windows kommer 1. halvår 1992. Programmet har brukergrensesnitt mot X Windows og SunView.

Maple er bygget opp av en liten kjerne og et stort bibliotek med over 2000 funksjoner som tas inn etter behov. Disse to delene er maskinuavhengige. Rundt disse er det et maskinavhengig brukergrensesnitt. Kjernen er skrevet i C og kompilert kode bruker mindre enn 500K av minnet.

Programmet inneholder et "programmeringsspråk", så en kan lage ferdige program for oppgaver en skal gjøre mange ganger. En kan også "logge" en interaktiv sesjon og bruke den som grunnlag for et program. De programmene vi lager selv utgjør en naturlig utvidelse av Maples bibliotek, og vi kan lage egne hjelpeskjerner for våre egne program.

Maples egne biblioteksfunksjoner er skrevet i Maples programmeringsspråk, og kildekoden til biblioteksprogrammene er tilgjengelig for brukerne.

I år har det kommet en ny utgave, utgave V (Visual), med en rekke forbedringer bl. annet når det gjelder X Windows grensesnittet, 3D grafikk, nye funksjoner, raskere flyttallsaritmetikk og hjelpesystem.

MSDOS utgaven krever minst 2Mbyte minne, det er ønskelig med 4 Mbyte. Nødvendig lagerplass er 5 Mbyte, mens det er ønskelig med 10 Mbyte tilgjengelig plass pga. "swapping". Tilsvarende gjelder for Macintosh.

Grafikkdelen er omfattende og merk at en kan få plottet både symbolske uttrykk og numeriske resultater - det siste f.eks. etter beregninger med måleresultater.

Avtalen omfatter også støtte fra den norske leverandøren (Geos Consult, Stjørdal).

Brukerhåndbøker:

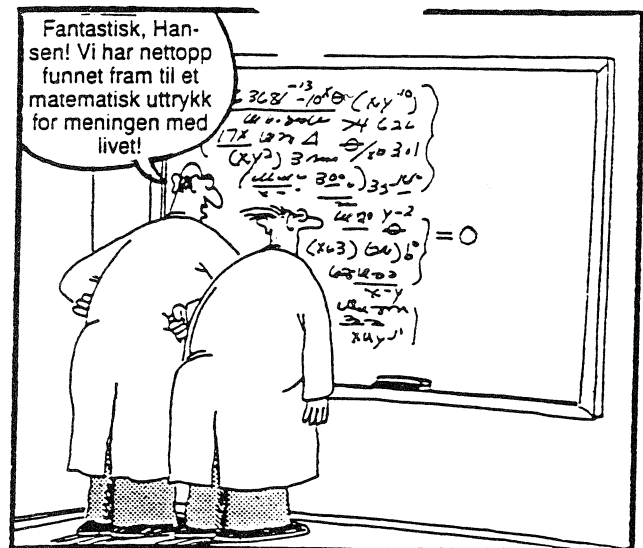
- Maple V Library Reference Manual
- Maple V Language Reference Manual
- First Leaves: A Tutorial introduction to Maple

Disse utgis for versjon V av Springer Verlag.

Vi vurderer dette som et meget nyttig program i vårt tekniske og naturvitenskapelige undervisnings- og forskningsmiljø. Ta kontakt for mer informasjon. De som mottar programmet må delta med en viss sum til spleiselaget for programmet.

Knut L Vik,

Eksempler på bruk av MAPLE står på de to neste sidene



Jeg vil her prøve å vise hvordan vi kan ta et sett med målepunkter inn i Maple, for så å plote denne datamengden i 3D. Deretter vil vi prøve å finne en funksjon som tilnærmer denne datamengden.

Vi starter med å lese definere kortformen av funksjonene i 3 pakker :

- `with(linalg): with(plots): with(stats):`

Pakken `linalg` inneholder bla. funksjonen `matrix` . Hvis vi ikke utførte `with(linalg)`, måtte vi kalle denne funksjonen med ``linalg/matrix`` eller `linalg[matrix]`. Alle kommandoer i Maple må avsluttes med `;` eller `:"` . `:"` undertrykker utskriften av resultatet.

Siden jeg ikke har noen måledata på lager, genererer jeg tenkte måledata : Disse data kunne jeg like gjerne tilordnet en liste, og senere lest inn listen med `read() filnavn`.

Jeg definerer først en funksjon `f`:

- `f := (x,y) -> [x,y,(x^2-y^2)]:`

Definerer dernest en funksjon `g` som er basert på `f` :

- `g := (u,v) -> subs(x=u-15, y=v-15,f(x,y)):`

Genererer nå ei 30×30 matrise med elementene `[u,v,g(u,v)]`. Matrisens indexfunksjon vil gi argumentene `(1..30, 1..30)` til `g(u,v)` :

- `A := matrix(30,30,g):`

Jeg later nå som om dette er måleverdiene som vi har lest inn med `read()`;

Vil nå fjerne det nest øverste parantesnivået i `A`:

`op()` finner operandene på øverset nivå til en datatype :

- `op([[a,b],[c,d]]);`

`[a, b], [c, d]`

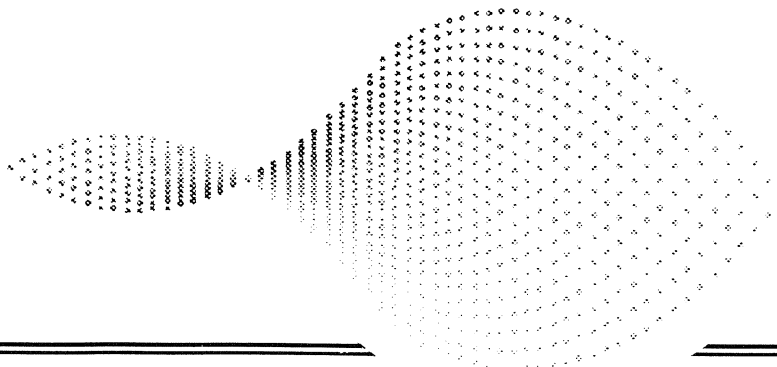
- `f1 := (x) -> op(x):`

Konverterer `A` til en liste av lister, og bruker `map` til å utføre `op()` på alle de 30 listene på øverste nivå : Dette vil fjerne dette parantesnivået, og vi vil stå igjen med ei liste av typen `[[u,v,g(u,v)], [u,v,g(u,v)], ..897,[u,v,g(u,v)]]`

- `B := map(f1, convert(A,listlist)):`

Jeg kan nå plote `B` ved hjelp av `pointplot`. Dette er en kraftig funksjon, fordi vi kan angi både `x`, `y` og `z` - verdi !

- `pointplot(B);`



Jeg kunne også ha plottet datamengden med *matrixplot*. Denne funksjonen antar at vi har en matrise av z-verdier, og at x- og y- verdiene er uniformt fordelt utover x-y/planet.

Vil nå gjøre regresjonsanalyse på datamengden for å finne en funksjon som tilnærmer datasettet: Vi må da fjerne det ene parentesnivået igjen, samt sette inn en vektor [x,y,z] som forteller hvilke variable vi har :

```
• V := putkey((convert(A,vector)),[x,y,z]);
```

```
• r := regression(V, z=a*x+b*x^2+c*x^3 + e*y+f*y^2+g*y^3);
```

```

r := {g = .5*10-14 , c = .16*10-12 , f = -1.0000000000,
      a = -.2296166892*10-10 , e = -.6331374529*10-12 , b = 1.0000000000}
```

```
• t := subs(r, z=a*x+b*x^2+c*x^3 + e*y+f*y^2+g*y^3);
```

```

t := z = - .2296166892*10-10 x2 + 1.0000000000 x3 + .16*10-12 x3
        - .6331374529*10-12 y2 - 1.0000000000 y2 + .5*10-14 y3
```

Vi ser at vi stå tilbake med $x^2 - y^2$ etter at vi har sløffet ledd med forsvinnende små koeffisienter !

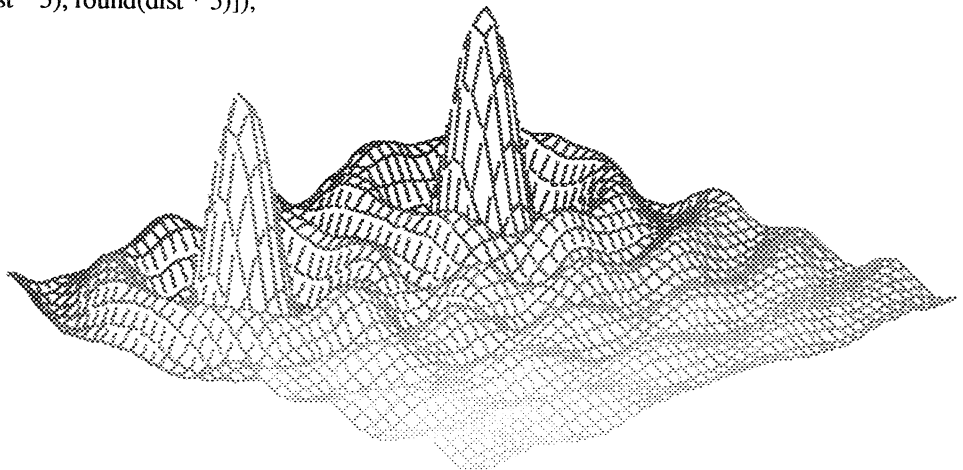
Til slutt ett lite eksempel på definering av egne funksjoner i Maple : Jeg definerer en funksjon pebbles() som simulerer interferensen mellom bølgene som oppstår når to steiner droppes i vann. Funksjonen tar 4 argumenter : Amplituden til bølgen fra de 2 steinene, avstanden mellom sentrene samt tidspunktet vi vil se på utviklingen :

```

• pebbles := proc(amp1,amp2,dist,t)
  plot3d(sin(t+sqrt((x-dist)^2+y^2))*amp1/sqrt((x-dist)^2+y^2)
        + sin(t+sqrt((x+dist)^2+y^2))*amp2/sqrt((x+dist)^2+y^2),
        x = -dist * 2 .. 2 * dist, y = -10 .. 4 * dist - 10,
        grid = [round(dist * 5), round(dist * 5)]);
```

```
end:
```

```
• pebbles(1,1,10,0);
```



Denne rapporten er i sin helhet laget i et såkalt "worksheet" i Maple på en Mac II av Geos Consult A/S. Rapporten er en direkte utskrift fra dette "worksheet".

MathCAD - det ideelle ingeniørverktøyet?

Hvis jeg måtte tilbringe et år på en øde øy, med strøm, en PC/Mac/Sun, og en printer, og bare fikk ta med meg ett verktøyprogram, da ville jeg tatt med MathCAD. Så ville jeg sittet og regnet reelt og komplekst, numerisk eller analytisk, skalart eller matriser. Data og formler kunne jeg hente rett inn fra de elektroniske håndbøkene (f.eks. om elektronikk) som følger med. Jeg ville laget pene grafer av resultatene, og plassert dem mellom likningene der de tok seg godt ut på skjerm og ark. Til slutt ville jeg skrevet den tekst som hørte til, både vanlige avsnitt og tekstblokker ved siden av likninger og grafer. Så ville jeg trykke det ut og legge det i postkassen til neste morgen, slik at jeg ikke skulle føle meg så ensom.

Hvordan det blir seende ut? Se selv, hele denne artikkelen er skrevet i MathCAD Ver.2.54, med en IBM 4019 laser printer med Courier 12 cpi font.

MathCAD som kalkulator

MathCAD er rask å bruke som kalkulator, både med rene tall og med variable som du først har gitt verdi. Og akkurat som i et regneark kan du gå tilbake i regningen og endre på tall og operatører, og få regnet om alt som følger etter.

$$\sqrt{\frac{4 - 3i}{3! + i}} = 0.834 - 0.357i$$

$$\text{erf}(1.23) = 0.918$$

Funksjoner:

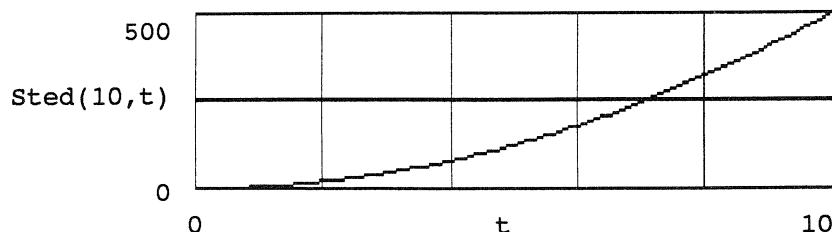
$$\text{Sted}(x,y) := x \frac{y^2}{2}$$

$$\text{Sted}(10,5) = 125$$

Teksten plasser du hvor du vil, i selvformatterende firkantede områder. Alle områder (med tekst, likninger eller graf) kan du klippe, flytte og lime som du vil, sålenge beregningsrekkefølgen ikke endres.

Graf kalles opp med @, etter at du først har definert en "område-variabel":

t := 0,0.1 ..10



Vektor- og matriseoperasjoner:

$$V1 := \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \quad V2 := \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix} \quad V1 \cdot V2 = 16 \quad V1 \times V2 = \begin{bmatrix} 5 \\ 5 \\ -10 \end{bmatrix} \quad V1 \cdot V2^T = \begin{bmatrix} 4 & 2 & 3 \\ 12 & 6 & 9 \\ 8 & 4 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 7 & 8 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} -0.508 & 0.381 & -0.111 \\ 0.444 & -0.333 & 0.222 \\ 0.206 & 0.095 & -0.111 \end{bmatrix}$$

MathCAD er mindre avansert enn f.eks Matlab når det gjelder matriseregning. Men som oftest er den god nok.

Integrasjon og derivasjon

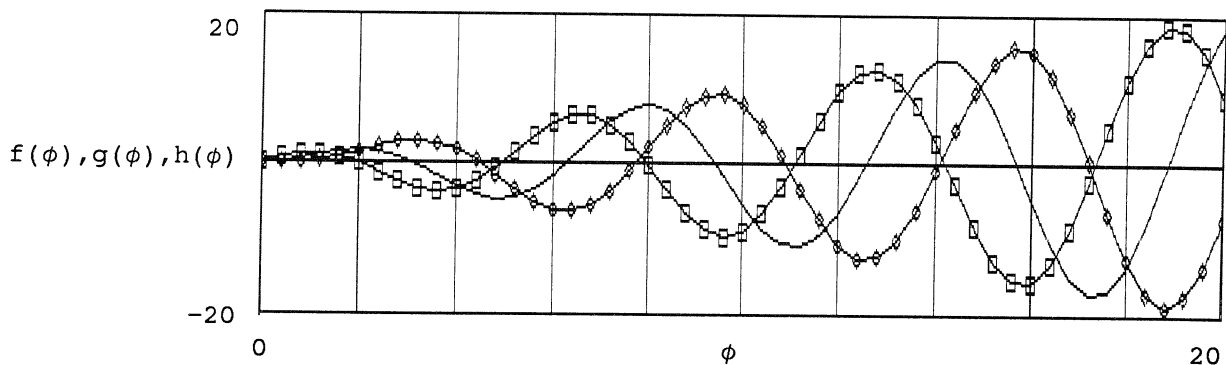
MathCAD kan også numerisk integrasjon og derivasjon. Den siste MathCAD-versjonen (3.0 for Windows) er kombinert med en noe forenklet versjon av MAPLE, som kan algebraiske (analytiske) beregninger.

$$f(x) := x \cdot \sin(x)$$

$$g(x) := \frac{d}{dx} f(x)$$

$$h(x) := \int_0^x f(y) dy$$

$$\phi := 0, 0.4 \dots 20$$



Numerisk likningsløser

Gitt tre resistanser $R1$ $R2$ $R3$ som skal kobles til et T-ledd, som skal gi en refleksjonsfri svekking $a = 0.1$ mellom to transmisjonslinjer av impedans $Z = 50 \Omega$. Beregn $R1$ $R2$ $R3$:

$$\text{Given} \quad R1 + \frac{R2 \cdot (R3 + Z)}{R2 + R3 + Z} \approx Z \quad R3 + \frac{R2 \cdot (R1 + Z)}{R2 + R1 + Z} \approx Z \quad \frac{Z - R1}{Z + R3} \approx a$$

$$R(a, Z, R1, R2, R3) := \text{Find}(R1, R2, R3)$$

$R1, R2, R3$ i $R(..)$ er prøveverdier.

$$R(0.1, 50, 25, 25, 25) = \begin{bmatrix} 40.909 \\ 10.101 \\ 40.909 \end{bmatrix} \quad \begin{matrix} R1 \\ R2 \\ R3 \end{matrix}$$

I Math Applications Pack finnes Runge-Kutta 4. ordens differensiallikningsløser. Det er også enkelt å modifisere denne eller lage en annen selv.

Noen andre funksjoner og egenskaper

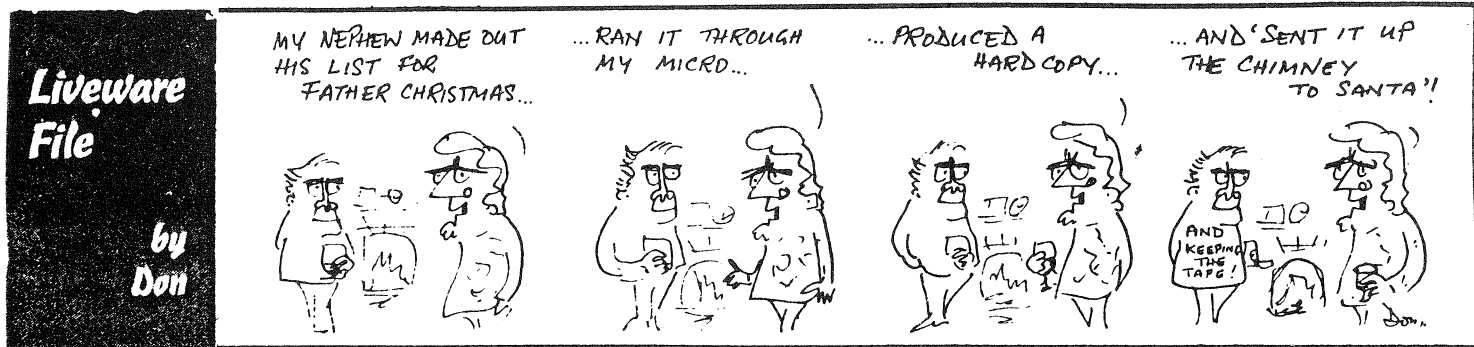
MathCAD har innebygget Fast Fourier Transform, som gjør det enkelt å lage numeriske filtere, f.eks. for bruk på filer med eksperimentelle data. Det er også lett å sette eksperimentelle data rett inn i data-arrays, og deretter behandle dataene statistisk (Statistics Application Pack). WINDOWS-versjonen 3.0 av MathCAD inneholder som nevnt faghåndbøker som man kan "bla" i på skjermen, og hente inn ferdige formler, materialdata, og illustrasjoner. Denne versjonen har bl.a. PostScript driver og fonter.

Vi har i noen år brukt MathCAD som vårt hovedverktøy for 1-2 år ved Avd. for Fysikk og Matematikk, og er godt fornøyd. Begynnerterskelen har vært noe høy, men WINDOWS-versjonen vil avhjelpe dette. Vi ser ikke på MathCAD som noen erstatning for større spesialverktøy som Matlab, MATHEMATICA eller fullstendig MAPLE, men som et all-round, lettbrukt ingeniørverktøy som bl.a. kan brukes til å lage fullstendige øvingsrapporter i alle fag. Fagbok-funksjonen er også unik.

UNIT vil få campus-lisens for det store symbolske matematikkverktøyet MAPLE (som du ikke kan skrive rapporter i). MathCAD 3.0 for WINDOWS har innebygget ca 80% av MAPLE, og vil derfor tjene som en naturlig introduksjon til dette avanserte verktøyet.

MathCAD er til nå solgt i ca 120 000 eksemplarer. Protech A/S (tel. 09-959020) rapporterer om ca 300 per år i Norge. Prisen til UNIT er rimelig.

Svein Sigmond, Inst.for Fysikk, tel. 3624



WordPerfect sikkerhetskopi

Sikkerhetskopiering erstatter ikke vanlig lagring eller backup av dokumenter, men kan være en stor hjelp i nøden dersom det skjer noe uforutsett som hindrer deg i å lagre dokumentet (f.eks. strømbrudd).

Dersom maskinen henger seg eller det blir strømbrudd mens du er inne i WordPerfect, er det som regel ikke annet å gjøre enn å starte maskinen på nytt. Det store spørsmålet blir da: Har du lagret det du skrev før det hele gikk i stå? Hvis du hadde skrevet mye nytt som du ikke rakk å lagre, er det allikevel et håp om å finne det igjen dersom du har bedt Word Perfect ta tidsregulert sikkerhetskopi. Slik gjør du det:

Tidsregulerte sikkerhetskopier - hvor ofte?

Du bestemmer hvor ofte WordPerfect skal lagre sikkerhetskopi av det dokumentet du arbeider med ved å gi kommandoen Oppsett (shift-F1), Omgivelser (3), Sikkerhetskopiering (1), Tidsregulerte sikkerhetskopier (1). Skriv J dersom du ønsker at WordPerfect skal ta tidsregulerte sikkerhetskopier, og angi antall minutter mellom hver kopiering. Etter at denne kommando er gitt vil du merke at det tas kopi ved at meldingen "Vent litt" vises nederst i skjermen hver gang kopi tas.

Hvor ligger sikkerhetskopien?

Du sier fra hvor du vil at kopien skal ligge med kommandoen Oppsett (Shift-F1), Lokalisering av filer (6), Sikkerhetskopier, og oppgi hvilket directory/dokumentregister du vil at de tidsregulerte sikkerhetskopiene skal ligge. Dersom du ikke oppgir noe lagringssted, legges kopiene på samme sted som WordPerfect-programmet, dvs der filen WP.EXE ligger.

Den tidsregulerte sikkerhetskopien av Dokument 1 i skjermen lagres på filen WP{WP}.BK1, mens Dokument 2 lagres på WP{WP}.BK2. Dersom du bruker nettversjonen av WordPerfect, er det muligens bestemt et sentralt sted for lagring av sikkerhetskopi (spør nettverksansvarlig).

Hvordan få tak i sikkerhetskopien når du trenger den?

Når du går ut av WordPerfect på vanlig måte, slettes den tidsregulerte sikkerhetskopien WP{WP}.BK1, og evt. WP{WP}.BK2. Men dersom WordPerfect ble avbrutt på en uvanlig måte, f.eks. ved strømbrudd, maskinfeil, at maskinen ble slått av uten at WordPerfect var avsluttet etc., vil du neste gang du starter WordPerfect få meldingen:

Gamle sikkerhetskopier finnes. 1 **Endre navn**; 2 **Slett**

Denne meldingen får du fordi filen WP{WP}.BK1, og evt. WP{WP}.BK2, ikke er slettet. Velger du nå Slett, sier du samtidig "Nei takk" til å ta vare på de sikkerhetskopiene som ble lagret før WordPerfect ble avbrutt sist du var inne. Da vil sikkerhetskopiene bli slettet, og WordPerfect kan begynne å lagre nye sikkerhetskopier på disse filnavnene.

Ønsker du å ta vare på sikkerhetskopien, svarer du Endre navn, og skriver et nytt filnavn for filen WP{WP}.BK1 (og evt WP{WP}.BK2). Filen med det nye navnet kan du deretter hente inn i skjermen på vanlig måte.

Steivor Bjarghov





Nytt fra DSL-UNIT

En vesentlig del av virksomheten ved Senter for datastøttet læring har også i 1991 dreid seg om utvikling av pedagogisk programvare til bruk i undervisning. Denne aktiviteten blir i hovedsak ivaretatt av diplomstudenter ved NTH.

Andre større aktiviteter senteret har arbeidet med i 1991 er videreutvikling av PC-verktøyet GIGS (Graphical Interface Generating System) og et fjernundervisningsopplegg i miljølære.

Diplomoppgaver 1991

- Pål Strøm, Institutt for geoteknikk, har i vårsemesteret arbeidet med utvikling av et interaktivt brukergrensesnitt (pre- og postprosessor) til et mye brukt beregningsprogram på Inst. for geoteknikk. Beregningsprogrammet regner ut skråningsstabilitet og bæreevne innen geoteknikk ut fra GPS-metoden (General Procedure of Slices, N. Janbu 1968).

I høstsemesteret arbeider følgende studenter med sine diplomoppgaver:

- **Thomas Robsham, Institutt for betongkonstruksjoner**, skal utvikle et dataprogram for beregning og dimensjonering av søyle- og veggfundament i armert betong. Programmet skal behandle moment og aksialkraft fra søyle/vegg og vekt av overliggende masser.
- **Arild Østgård, Institutt for betongkonstruksjoner**, skal lage et program for førøppspente betongbjelker i bruksgrensetilstand. Programmet beregner nedbøyning og riss for forskjellige tverrsnitt ved hjelp av matrisestatikk.
- **Bjarne Austerheim, Institutt for elkraftteknikk**, skal lage et grafisk grensesnitt til simuleringsprogrammet EMTP, der brukeren kan tegne inn elektriske nettverk, og la modulen generere input datafil til simuleringsprogrammet.
- **Maj Paulsen, Institutt for elkraftteknikk**, skal lage et grafisk grensesnitt til et simuleringsprogram vedrørende stabilitet av kraftsystemer.

- **Svein Johansen, Institutt for elkraftteknikk**, skal lage et grafisk brukergrensesnitt til forskjellige lastflytberegningsrutiner. En lastflyt fastlegger de ukjente elektriske størrelser i et elektrisk nett, bestemt av nettets topografi og elektriske egenskaper. Programmet skal gjøre det mulig å tegne nettet på skjermen interaktivt, samt lese inn og ut de forskjellige elektriske parametre. Programmet skrives i C++ med bruk av objektorientert programmeringsteknikk. Det legges vekt på at det skal være skjermuavhengig, og at det skal være enkelt å utvide programmet senere med nye elektriske objekter og beregningsrutiner.

Alle studentene som har vært tilknyttet senteret i 1991, har benyttet GIGS til generering av brukergrensesnittet i programmene.

Videreutvikling av GIGS (Graphical Interface Generating System)

Historie

Applikasjonsgeneratoren GIGS er utviklet ved DSL-UNIT gjennom en toårsperiode. (Se RUN-NYTT nr 4, 1990). Verktøyet er utviklet til bruk på IBM-kompatible PC-er under DOS operativsystem.

GIGS har blitt utprøvd med meget godt resultat i flere prosjekter. Ved senteret benytter studenter verktøyet til utvikling av læremoduler for bruk i undervisningen. Omfattende og teknisk avanserte programmer er blitt utarbeidet i løpet av 4-5 måneders arbeid som hovedoppgave. GIGS kan generelt brukes til å lage brukergrensesnitt til ulike typer program.

Det er blitt vist en positiv interesse for GIGS både nasjonalt og internasjonalt og spesielt internt ved UNIT. Totalt er ca 70 programpakker distribuert til nå. Internt ved UNIT er GIGS fritt tilgjengelig. DSL-UNIT har til nå registrert 14 fagmiljøer ved NTH/SINTEF som benytter GIGS aktivt i appkikasjonsutvikling.

I Norge, utenfor UNIT, er det distribuert ca 30 programpakker. Til andre skandinaviske land er det distribuert 8 pakker, og utenfor Skandinavia 5 pakker. I tillegg har MEDC, Paisley College i i Skottland, kjøpt lisensrettigheter for GIGS til skotske colleges.

GIGS leveres på en 3 1/2" (1.44 MB) eller en 5 1/4" (1.2 MB) diskett.

Komplett dokumentasjon leveres i form av 3 manualer: User's Guide (140 sider), Reference Guide (700 sider) og Utilities and Exercises (75 sider). All dokumentasjon er på engelsk.

Videre utvikling

Senteret har i 1991 arbeidet med utvikling av XGIGS, dvs GIGS for vindussystemet X under UNIX operativsystem. Det er i dette prosjektet blitt lagt sterk vekt på enkelt å kunne flytte applikasjonene mellom forskjellige maskinvare konfigurasjoner og mellom operativsystemene DOS og UNIX.

Under DOS vil det bli arbeidet for å tilrettelegge GIGS for MS-Windows. I løpet av 1991 vil det foreligge en ferdig utarbeidet krav- og produktspesifikasjon for XGIGS. Etter de foreliggende planer regner senteret med å ha en kjørbar versjon under UNIX til skolestart 1992.

Tilfellet Tellus - vårt miljø, vårt ansvar

Dette er et norsk flermedieprosjekt i miljølære som ved hjelp av radio- og TV-program, samt selvstudier, samlinger og ekskursjoner skal lede frem til universitetseksamen tilsvarende 1/2 års studium. Kurset har et omfang på 10 vekt-tall, hvorav 2 utgjøres av en innføringsdel. Før oppstarten 16. september hadde det meldt seg over 1100 søkere. 230 er tatt opp til innføringsdelen, mens 660 studenter er tatt opp til 10-vektstallsstudiet som avsluttes med eksamen i september 1992.

PC-hjelpemidler

Som et supplement til fagbøkene i innførings- og fordypningsdelen har DSL-UNIT utarbeidet en pakke med PC-hjelpemidler. Den består av 5 dataprogram, samt et hefte med brukerveiledning og et øvingsopplegg. Senterets utviklingsverktøy, GIGS, er benyttet til utarbeidelse av samtlige moduler.

Faglig innhold i PC-modulene:

- **Flervalgstest.** Denne består av en modul hvor brukeren kan hente inn et utvalg av

tester, hvor hver test består av 20-30 spørsmål som er nært knyttet til det faglige innholdet i lærebøkene. I tillegg til de enkelte spørsmål, som hvert har fra to til fire svaralternativ og med mulighet for kommentarer til hvert svar, kan grafikkbilder og hjelpetekster vises i vinduer på skjermen

For å lage testene er det utviklet en forfattermodul. Testene genereres interaktivt, og til hvert spørsmål lages inntil fire svaralternativ med mulighet til å legge inn kommentarer til hvert svar.

Grafikkbilder kan lages med generelle tegneverktøy eller hentes inn i form av scannede bilder. Grafikkformater av type PCX (Paintbrush) eller LBM (DeluxePaint) kan benyttes direkte. Andre formater må først konverteres til PCX eller LBM.

Tekster for fremvisning i vindu i flervalgstesten kan genereres via en innebygd teksteditor eller fra en generell ASCII-editor.

- **Biomasse.** Dette er et simuleringsprogram som tar for seg samfunnsøkonomisk forvaltning av fornybare biologiske naturressurser.

Programmet gir mulighet til å studere sammenhengene mellom biomasse, høstingsnivå og tilvekst, samtidig som det gir et bilde av hvordan biomassen utvikler seg over et gitt tidsrom.

- **Optimal utvinning.** Programmet tar for seg ikke-fornybare ressurser og økonomisk teori. Formålet er å vise hvordan prisbane på en ikke-fornybar ressurs utvikler seg under monopol og fri konkurranse.

Brukeren kan her lese inn variabler som kvelningspris, enhetskostnad, diskonteringsrate, ressursmengde og etterspørselskonstant, og få beregnet startpris, nåverdi og tidspunkt der prisbanen når kvelningsprisen.

- **Forurensningsøkonomi.** Programmet prøver å belyse forskjellige samfunnsøkonomiske tiltak som kan settes i verk for å redusere CO₂-utslipp til et land eller i global sammenheng. Brukeren kan eksperimentere med avgifter, skatter og standarder, og se de økonomiske konsekvensene av de valgte tiltak.

forts side 17

Databaser om forskningsprosjekter

SIF - NAVFs "Sentral for informasjon om forskningsprosjekter" - holder til ved NAVFs EDB senter for humanistisk forskning i Bergen (fra årsskiftet endrer senteret navn til Humanistisk Datasenter).

SIF samler inn informasjon om forskningsprosjekter fra forskningsmiljøene i Norge, og denne informasjonen er tilgjengelig i en serie databaser samlet i Universitetet i Bergen sin IBM 4381 maskin. Disse databasene er det gratis å bruke.

Databaser

Antall prosjektdokumenter er nå ca 31500. Det er både prosjektbeskrivelser (ca 15000) og henvisning til litteratur produsert underveis i prosjektene (underveisresultat).

De viktigste basene er - antall dokumenter pr. 7/10 1991 står i parantes:

HFORSK (5483)	Humanistisk forskning
SFORSK (7543)	Samfunnsvitenskap - NAVF/RSF
NORAS (4839)	Anvendt samfunnsvitenskapelig forskning
MFORSK (1018)	Medisin - NAVF/RMF
NFORS (2725)	Naturvitenskap - NAVF/RNF
NLVFORSK (457)	Landbruksforskning - NLVF
NTNFORSK (593)	NTNF-støttet forskning
FISK (468)	Fiskeriforskning - NFFR
UIO (685)	Universitetet i Oslo
UNIT (315)	Universitetet i Trondheim - AVH
DHMN (371)	Naturvitenskap ved DH skoler

Hovedbaser er HFORSK og SFORSK

Basene bruker søkesystemet SIFT, og når SIFT startes har en tilgang til alle basene.

Tilknytning

Når vi skal ta denne tjenesten i bruk, må vi henvende oss til SIF og få brukernavn og passord til IBM maskinen og til programmet SIFT.

IBM maskinen kan vi nå på følgende måte:

- Oppringt samband til telefonnr.: 05-5440343 (300, 1200 og 2400 baud)

- Datapakadresse 450001011
- IXI adresse 2043420021011
- Med Telnet til adresse livid.uib.no (129.177.1.7)

Det er med Telnet de fleste UNINETT brukere enklest kan nå maskinen, og vi skal se litt nærmere på denne måten.

Det er to ulike Telnet program en kan benytte - Telnet og TN3270. (Vi kan også bruke den nye KERMIT utgaven (3.11) med støtte for TCP/IP kommunikasjon).

a) TN3270

Basen er på en IBM maskin og TN3270 er et program som er spesielt tilpasset kommunikasjon mot IBM maskiner. På en IBM maskin får en meny på skjermen, og en bruker tastaturtaster for å velge. TN3270 er tilpasset dette.

Når vi har startet SIFT, får vi med TN3270 en meny nederst. Merk at hvis det i terminalprogrammet ikke er definert sammenhenger mellom PCens taster og de tegn (taster) vi velger på IBM maskinen med , så må vi trykke på flere taster for å gjøre våre valg. (f. eks ESC 3 for slutt).

Brukerveiledningen har en oversikt over hva som må brukes. Heldigvis kan vi i SIFT utføre alt ved å skrive kommandoene, så usikkerhet mhp. tastene utelukker ikke bruk av denne tjenesten.

Merk at med TN3270 kan vi nederst på skjermen få "More" eller "Holding". Dette får vi når vi er i operativsystemet og ikke i SIFT, og når en skjermsside er fullskrevet. Da må vi trykke på IBM tasten CLEAR for å komme videre til neste skjermsside.

Håndboka sier at ESC. (ESC tast og punktum samtidig) utfører CLEAR. Hvis det ikke virker, prøv CTRL Z (ikke på en UNIX maskin). Hvis ikke noe virker - la skjermen stå i More. Da skifter maskinen automatisk til ny side hvert 60. sekund inntil all ventende tekst er vist.

b) Telnet

Med dette programmet kommer vi inn til IBM maskinens linjebaserte inngang, hvor vi skriver kommandoer og får utskrift linje for linje. Da har vi ingen problemer med taster og "More/Holding" - uansett tastatur.

I SIFT må vi da skrive inn alle kommandoer.

Et PC eller Macintosh terminalprogram har oftest funksjonen "logging" - alt som skrives på skjermen skrives parallelt på en PC eller Macintosh fil. Alle søkeresultater kan på denne måten f.eks. legges direkte på vår egen PC. Når vi kopler oss opp til IBM maskinen på denne måten, sendes tekstlinjene til terminalen uten tegn for skjermstyring innimellom, og det som kan tas fram fra logfilen er greit å lese.

Med TN3270 får vi også logget alle skjermstyringstegn. Da lønner det seg ikke å bruke logging. Vi må isteden gjøre som håndboken beskriver, og lagre søkeresultatet på en fil på IBM maskinen. Denne filen overfører vi så til oss selv med filoverføringsprogrammet FTP (eller Kermit) .

Når vi kommer til inngangen til IBM maskinen får vi med Telnet ikke spørsmål om brukernavn. Vi må starte med å skrive logon brukernavn, hvor brukernavn er det brukernavnet vi får fra SIF.

Noen SIFT kommandoer

list databaser

Får en navneliste over databasene

åpne databasenavn

Åpner databasen "databasenavn" for søking.

Søking skjer i alle de databasene som er åpnet.

finn ord

Leter etter dokumenter hvor ord finnes i beskrivelsen. (SIFT er et fritekst søkesystem)

vis

Skriv ut et dokument

neste side

Bla videre

neste resultat

Neste dokument

slutt

Merk at utlogging fra IBM maskinen gjøres med logoff.

Avslutning

Hensikten med denne noe detaljerte beskrivelsen er å hjelpe i gang de som er tilknyttet UNINETT. Vi understreker at vi benytter UNINETT også når vi kopler oss opp til IBM maskinen med Datapak nummeret. Da må vi bruke et annet terminalprogram enn Telnet - f. eks. Kermit.

Vi oppfordrer den som føler behov for informasjon om forskningsprosjekter i Norge å ta denne tjenesten i bruk.

For mer informasjon - kontakt:

SIF
Postboks 53
Universitetet - 5027 Bergen
Tlf. 05-212957, Fax: 05-322656
E-post: fafig@nobergen.bitnet

DSL-UNIT forts

- **Miljøgiftmodeller.** Dette programmet omhandler distribusjon av giftige stoffer i en organisme og i en næringskjede. Det blir fokusert på to standard toksikologiske distribusjonsmodeller ("I- og II-compartment"- modellen), "steady state"-modellen og akkumulering i næringskjede.

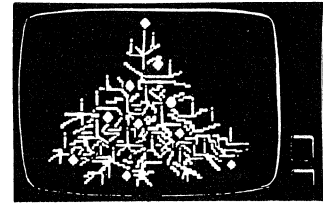
Nærmere informasjon om fjernundervisningsopplegget "Tilfellet Tellus" kan fås fra

Universitetet i Bergen
Senter for etter- og videreutdanning
Strømsgaten 51, 5020 Bergen
(Tlf.: 05 212040).

Gir denne artikkelen deg ideer til å ta i bruk GIGS for å forbedre brukergrensesnittet i program du har laget eller skal lage, eller til å benytte programmet Flervalgstest til å lage "multiple choice" oppgaver innen ditt fagområde?

Ta kontakt for mer informasjon.

Jan Ole Waagen,
DSL-UNIT, 7034 Trondheim
tlf. 594787
waagen@lise.unit.no



Bruk BIBSYS fra eget kontor

Dette er en mye omarbeidet utgave av en artikkel som sto i RUN-NYTT nr 2, 1990. Den samme artikkelen er også spredt som UNINETT notat T 1.1.

Alle kan bli brukere av BIBSYS og søke i basen fra egen terminal eller en PC brukt som terminal. En PC som terminal er det mest praktiske å bruke.

Søking i BIBSYS er gratis. Brukere ansatt på universitetene og tilknyttede forskningsinstitusjoner (f. eks. SINTEF) kan kjøpe dokumentasjonen for kr. 150.-, mens brukernr./passord er gratis. Andre eksterne brukere må betale kr. 300.- for brukernr./passord. Dokumentasjonen er da inkludert.

Vi skal her se på hvordan vi kan nå BIBSYS maskinen, og litt på hvordan en praktisk kan bruke maskina og hva vi må passe på.

Tilknytning

Til BIBSYS maskinen har vi følgende oppkoplingsalternativer:

- 1) For brukere tilkoplede linjesvitsjnettet ved UNIT: Klasse B.
(Ved spørsmål: ENTER CLASS, svar B).
- 2) For brukere tilkoplede terminalnettet på bredbåndsnettet ved UNIT:
call 2500.

Disse brukerne kan også gå via linjevitsjen vha. Call 0.
- 3) Oppringt samband - til telefon nr.:
(07) 945801 (1200 og 2400 baud)
(07) 945802 (1200 og 2400 baud)
(07) 940705 (1200, 2400, 4800 og 9600 baud)
- 4) Datapak:
Terminalinnstilling
"8 bits none" - adresse (NUA): 530001010
"7 bits even" - adresse (NUA): 530001006
- 5) IXI - adresse (NUA):
20434240001010 (8 bits none)

- 6) UNINETT Internet adresse:
castor.bibsys.no ev. 128.39.9.9

Den gamle adressen bibsys.unit.no kan også fortsatt brukes mot denne inngangen.

- 7) UNINETT Internet adresse:
pollux.bibsys.no ev. 129.241.1.61

Oppkoplingen gjennom Internet skjer vha. terminalprogrammet TELNET. Dette er et program vi har på en lokal maskin - på en mikromaskin, en arbeidsstasjon eller en større fellesmaskin. Den lokale maskinen må være tilknyttet Internet, og denne maskinen kan like gjerne være i utlandet, tilgjengeligheten er den samme.

Det er to telnetprogram - det ene heter TELNET og det andre TN3270. TELNET er et VT100 terminalprogram. TN3270 er spesielt laget for bruk mot IBM maskiner - det er et IBM 3270 terminalprogram. Begge disse programmene kan være deler av samme Telnet program (PC, Macintosh)

Når vi starter TN3270 fra en VAX maskin, må vi huske å sette terminaltype til VT100 vha. VAX kommandoen SET TERM /VT100.

TELNET brukes slik: TELNET castor.bibsys.no ev. TELNET 128.29.9.9

I siste KERMIT utgave for MSDOS, v. 3.11, er det kommet støtte for kommunikasjon over Internet, så også dette terminalprogrammet kan nå brukes mot Internet adressene.

Gjennom terminaltjener IBM 7171

Med alternativene 1-5 kommer vi til en terminaltjener - IBM 7171 - som spør: "Enter terminal type:" Ved å trykke Return (CR) får vi en liste over valgmuligheter. Velg VT100N1 hvis du har et VT100 terminalprogram på PC'en. Den som har en fargeskjerm kan også velge terminaltype VT100N2 eller VT100N3. BIBSYS har utgitt oversikten: "Terminaltyper ved søking i BIBSYS".

Når vi har valgt terminaltypen, kommer BIBSYS logoen på skjermen, og en kan logge seg inn.

Internet adresser

Ved *alternativ 6* (castor.uninett.no) er veien inn til maskina gjennom en annen boks - vi kommer først til "Mitek TELNET Server". En kommandomeny får vi med ?. Vi kan velge flere terminaltyper - kommando show_term gir valgmulighetene. Gyldige terminaltyper mot BIBSYS er VT100N1, VT320N7 (7 bit) og TDV2200/9. Merk at TDV2200/9 ikke dekker TDV2215, så denne terminaltypen kan ikke brukes denne vegen.

De som har en VT100 skjerm eller terminalprogram, velger VT100N1. Dette er standard valg. Terminaltype velges med !terminaltype.

Når terminaltypen er valgt, startes en 3270 terminalsesjon, og vi logger oss inn med: VM brukernavn.

Merk at vi får en hjelpeskjerm for terminalserveren vha. ESC h, og at vi avbryter terminalsjesjonen vha. ESC xx (Tasten ESC). Hvis alt stopper opp, og vi nederst på skjermen får What ?, må vi sende CTRL a (RESET)

Med programmet TN3270 har vi allerede valgt terminaltype, og vi kommer rett til spørsmålet: Skriv VM brukernavn.

Ved utlogging fra BIBSYS maskinen kommer vi tilbake til terminaltjeneren og "Skriv VM brukernavn". Vi kan kople oss tilbake til egen maskin med ESC xx.

Ved *alternativ 7* (pollux.bibsys.no) går vi gjennom enda en ny veg. Her er det stor forskjell på å bruke TELNET eller TN3270 programmet.

Vi kommer med begge rett inn på inngangen til BIBSYS maskina. Med TN3270 får vi BIBSYS maskinens "logon" skjerm bilde, og etter innlogging får vi maskinens skjermmodus, hvor vi velger med PF taster, osv.

Med TELNET, som et er linjebasert terminalprogram, kommer vi inn i BIBSYS maskinens linjmodus. Vi får ikke spørsmål om brukernavn - vi må skrive: logon brukernavn. Som vi skal se senere, kan denne måten å kople oss opp på være nyttig selv om det med en gang virker som vi mister mye.

En ting vi med en gang kan merke oss er at med TELNET og denne inngangen slipper vi alle usikkerheter med hvilke taster vi skal trykke på, og vi slipper at det hele kan stoppe opp når skjermen er fullskrevet (situasjonen når vi i skjermmodus

får "holding" eller "more" nederst på skjermen).

Vi må merke oss at bruk av programmet TELNET mot de to Internet adressene gir forskjellig innloggingsmøte med maskina og forskjellige muligheter.

Søking i BIBSYS

Når vi er innlogget i BIBSYS kan vi velge mellom to søkeprogram - PUBSØK og GENSØK.

PUBSØK er et skjermorientert program hvor vi gjør valg ved å trykke på det IBM kaller PF taster. På en PC eller en "ikke IBM" terminal må vi trykke på en eller flere taster etter tur for å sende kodene PF tastene egentlig sender. Se "Terminaltyper ved søking i BIBSYS" for mer informasjon om tastaturutlegg for ulike terminaltyper.

BIBSYS leverer en spesialtilpasset utgave av kommunikasjonsprogrammet KERMIT for MSDOS hvor PF tastene er utlagt på F-tastene på PC tastaturet.

For den som ikke har et terminalprogram spesialtilpasset PF tastene, vil det nok lønne seg å velge GENSØK som er et kommandostyrt program. Det er ikke så mange kommandoer vi trenger.

Dette programmet har også kommandoer for å styre utskrift av søkeresultatet til en skriver (P=slave) tilkople PC-en, og for å lage en oversiktlig utskrift på en loggfil åpnet i terminalprogrammet (P=logg). Dette gjelder dog bare når vi kommer inn i BIBSYS maskina gjennom terminaltjeneren 7171 (alternativ 1-5).

Linjediialog

En nyttig ny tjeneste i GENSØK er linjediialog. Kommandoene er de samme, men utskriften skrives ut linje for linje og ikke skjermstyrt som er standard for GENSØK. Hensikten med denne dialogen er at vi kan bruke GENSØK fra alle typer terminaler, og at vi lettere kan få utskrift på fil eller på en skriver av det som skrives til skjermen.

En meget nyttig måte å arbeide på er å slå på "logging" i et terminalprogram og få lagret på en PC-fil hele terminalsjesjonen, med de søkekommandoene vi gir og resultatene av søkingen.

Linjediialog får en:

- a) Ved oppkoplingsalternativene 1-5, gjennom terminaltjener 7171:
- ved i terminaltjeneren 7171 å velge terminaltype TTY. Da kommer en rett inn i GENSØK og linjemodus og kan ikke bruke skjermmodus i GENSØK.
 - ved i GENSØK å gi kommandoen
Defin dialog=linje (d di=li). Vi kommer tilbake til skjermtutgaven med: D di=skjerm. Vi kan bruke både skjermmodus og linjemodus. Vi har ved innloggingen valgt terminal VT100.

b) Gjennom Internet

Gjennom Internet kan vi bare bruke linjediialog når vi bruker TELNET og adressen pollux.uninett.no - dvs. når vi bruker den linjebaserte inngangen.

Vi kan ikke bruke linjemodus ved oppkopling til castor.bibsys.no, eller når vi bruker terminalprogrammet TN3270.

Merk at hvis vi i linjemodus skal rette feil i en kommando vi gir, og piltastene på tastaturet ikke virker, vil CTRL h flytte "cursor" mot venstre. Skriv så på nytt.

Ved linjemodus og oppkoplingsalternativene 1-5, må terminalen eller kommunikasjonsprogrammet i PC'en være innstilt lokalt ekko (halv duplex). Kommando i KERMIT: set local-echo on. Gjennom Internet gjelder ikke dette.

Noen kommandoer i GENSØK

Forkortning vi kan bruke står med store bokstaver:

Hjelp: HJelp, ? kommando

Søking: Finn

Finn fraktaler, finn trondheim*historie

Liste over referanser i et søkesett - en linje pr. bok:
Liste.

Blaing i lista: MER, BAK

Utskrift av referanser fra et søkesett: SKriv

sk 5 - bok nr. 5

sk 1-10 - bok nr. 1 til og med 10

sk 3,5 p=slave - bok nr. 3 og 5 til skriveren på PC-en

Er utskrevne bøker utlånt: LListe

Avslutt: SLutt

Merk at i linjediialog er det ikke mulig å bruke p=slave. Da må vi bruke de muligheter vi har for "logging" i det terminalprogrammet vi bruker.

Brukerveiledninger

BIBSYS har utgitt brukerhåndbøker både for PUBSØK og GENSØK. I tillegg er det endel kortere orienteringer:

- BIBSYS - et automatisert bibliotekssystem
- BIBSYS. litt om systemet
- BIBSYS. oppkopling av samband
- Terminaltyper ved søking i BIBSYS
- PUBSØK - enkelt litteratursøk i BIBSYS
- GENSØK - kort brukerveiledning
- GENSØK - bruk av linjediialog
- Bruk av MS-KERMIT fra BIBSYS.
- Bibliotekskoder og adresser.
- Enkel søking. (Om PUBSØK LIGHT)

BIBSYS utgir også BIBSYS-Nytt.

Avslutning

I RUN-NYTT nr. 4, 1991, s 26-27, er det fortalt om programmet PUBSØK LIGHT og lånebestilling direkte fra PUBSØK.

NTUB holder kurs i litteratursøking i BIBSYS. Kontakt NTUB, tlf 59 51 10. Kursene holdes 1 gang hver måned i vår- og høstsemesteret.

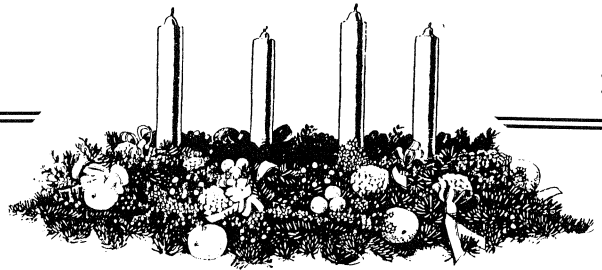
I katalog pub/bibsys i programvaretjener ugle.unit.no er det lagt inn diverse terminalprogram som vi kan bruke mot BIBSYS - se filen LESMEG. Dette gjelder både MSDOS, Macintosh og UNIX program, og utgaver tilpasset norsk tegnsett.

Bruk BIBSYS fra egen terminal og eget kontor - det er enkelt å foreta søking selv!

BIBSYS inneholder nå ca. 787 000 ulike titler. I tillegg er det gjennom LC-basen (Library of Congress) og NB-basen (Norske publikasjoner) tilgang til henholdsvis 1.0 million og 180 000 titler (se RUN-NYTT nr. 2, 1991, s 15, om LC og NB basen)

Kontakt BIBSYS for mer informasjon -
tlf.: (07) 59 70 67

Knut L Vik



UNIX - diverse

I siste RUN-NYTT hadde vi en oversikt over UNIX kommandoer for ulike oppgaver, samt en kort beskrivelse av endel andre muligheter i UNIX.

Vi skal her nevne noe nytt og utdype noe. Vi begrenser oss til forhold som gjelder kommandotolkerne (skallene) csh og tcsh (C-shell) - for de andre skallene er det endel likt med det som beskrives her.

Variable

Vi kan definere variable hvor vi kan lagre f.eks. lange katalognavn vi ofte bruker. Vi har lokale variable for skallet og globale (miljø) (environment) variable. De siste overføres automatisk til nye skall og program som startes. Lokale variable defineres med set kommandoen (set variabel=verdi), globale variable med setenv kommandoen (setenv variabel verdi). For globale variable bruker en å skrive variabelnavn med store bokstaver.

Det finnes forhåndsdefinerte variable som systemet har definert for alle - f.eks. HOME som inneholder navnet på egen hjemmekatalog.

En liste over definerte lokale variable får vi ved å skrive set, og en liste med globale variable med printenv.

Innholdet i en variabel tas fram for bruk ved å sette \$ tegnet foran navnet - f.eks. ls \$HOME.

Alias

Dette er en nyttig egenskap. Vi kan definere egne forkortelser for kommandoer eller program - f.eks. hvis antall opsjoner er mange og linja er lang.

Alias defineres slik: alias aliasnavn kommando

Eks.:

```
alias lst 'ls -l | more'
```

Ved å gi kommando lst stopper fillista ved full skjerm

```
alias prog /usr/ole/oleprog
```

Starter programmet oleprog ved å skrive prog. Vi kunne også lagt katalog /usr/ole inn i PATH kommandoen.

```
alias filtall 'find . -print | wc -l'
"Kommandoen" filtall gir oss antall filer
(og kataloger) totalt i alle underkataloger.
alias nykat 'cd \!*; la -lt | more'
Med nykat skiftes det til oppgitt katalog,
og vi får alltid en filliste sortert etter dato
for siste lagring. Eks.: nykat News.
Vi oppgir parameterverdier sammen med
aliasnavnet (\!*), og det er to kommandoer
adskilt med ;.
```

Vi vil også ha noen aliaser definert av systemansvarlige for alle brukere.

Vi ser hvilke aliaser som er definert ved å skrive alias.

Merk at vi setter teksten mellom anførselstegn når den består av flere ord, og at det foran ! må stå \ for å hindre at skallet tolker ! som "history" kommandoen.

Innloggingsfiler

Alt som vi vil ha fast definert, må vi legge inn i innloggingsfiler. Det gjelder f.eks. variable og aliaser.

Når vi starter et C-shell, vil filen .cshrc leses, og det som står der blir utført. I denne filen definerer vi lokale variable og aliaser, vi kan definere vår standard filbeskyttelse, osv.

Ved innlogging på maskinen leses filen .login etter .cshrc. Ved nye start av C-shell uten utlogging vil .login bli ignorert. I .login defineres globale (environment) variable, aliaser, vi kan starte et vindusystem, definere terminalkarakteristikker, osv.

Filbeskyttelse

I den fillisten vi får på skjermen med ls -l, står det lengst til venstre på hver linje f. eks. - -rwxr-xr- -. Dette viser filenes og katalogenes filbeskyttelse (bemyttelsesmaske). Der er 10 tegn. Lengst til venstre står det i eksemplet -, og det forteller at dette gjelder en fil. Hvis det står d, beskriver linja en katalog.

De neste 9 tegnene er delt inn i tre felt med 3 tegn i hver. Vi får vite beskyttelsen for filens

forts. neste side

eier (user, u), gruppe (group, g) og andre (other users, o). De tre tegnene i hver gruppe betyr leseadgang (r), skriveadgang (w) og rett til å utføre filen som et program (x). Filens eier har alle rettigheter - nemlig rwx.

Dette er kompakt, men ikke særlig forståelig uten forklaring.

Merk at en katalog må ha utførelsesadgang for hver gruppe (av de tre) som skal ha adgang til katalogen.

Det er viktig å være oppmerksom på om vi f.eks. gir alle leseadgang til våre filer.

Vi kan sette ny adgang til en fil eller en katalog med kommandoen `chmod` og standard adgang for alle nye filer med kommandoen `umask`.

Umask kommandoen vil vi sette inn i `.cshrc` filen. Umask brukes slik at vi forteller hvilke adganger filene IKKE skal få!

Vi henviser til annen litteratur for detaljer om hvordan disse to kommandoene brukes. Her er noen eksempler:

`chmod o-rx filnavn`

Fjerner lese og utførelsesadgang for andre (others)

`chmod g=rw filnavn`

Gir gruppen (group) lese og skrivetilgang

`chmod g+w filnavn`

Legger til skriveadgang for gruppe.

`umask 077`

Det er et siffer for hver eiergruppe. Eier (user) har ingen begrensninger, de andre to gruppene har full bregrensning. Dette betyr at bare eier (user) har adgang til nye filer som opprettes.

`umask 027`

Eier (user) har full adgang, gruppe har leseadgang (og utførelsesadgang for kataloger og programfiler), andre (others) har ingen tilgang.

Vi kan også bruke slike tall sammen med `chmod` - her med betydning adgang:

`chmod 654 filnavn`

Eier har lese og skriveadgang, gruppe har lese og utførelsesadgang, og andre har lese- adgang.

Hver av disse tallene er en sum av leseadgang (4), skriveadgang (2) og utførelsesadgang (1).

Knut L Vik

NetNews diskusjons- grupper



Under NetNews er det nå ca 1425 ulike nyhetsgrupper - og flere kommer til. Endel av disse vil sikkert være nyttig for deg.

Lokale grupper grupperes under unit...., og det er 13 grupper - fra `unit.unix-users` til `unit.film`. Innholdet i disse spres ikke utenfor UNIT miljøet. Grupper for hele Norge finnes som `no...` - her er det 19 stykker. Vi nevner `no.uninett.diverse`, `no.unix` og `no.kjemi`

På nordisk nivå finnes det gruppene `nordunet...` Her diskuteres f.eks. nordisk nettsamarbeid.

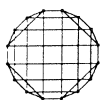
Foreløpig er ikke antall innlegg i disse gruppene så mange. Det er en begynnelse på et nyttig tilbud.

Den ved UNIT som ikke har installert et NetNews brukerprogram hos seg selv, kan ta ut bruker på `due.unit.no` og lese Netnews der. Brukerprogrammet heter `trn` - se man `trn`. Merk at en bruker programmet `Pnews` for å sende innlegg til News gruppene.

Vi henviser til RUN-NYTT nr. 3, 1991, s 4, for informasjon om NetNews brukerprogram for PC og Macintosh.

Knut L Vik





Parallele datamaskiner på Hyperkubelaboratoriet ved IDT

Hyperkubelaboratoriet ved Institutt for datateknikk og telematikk har vokst fram gjennom en 7-årsperiode fordi vi ønsket å eksperimentere med parallelle metoder for utførelse av databaseoperasjoner. Vi hadde spesielt behov for maskiner som kunne utveksle data mellom nodene meget hurtig. Slike fantes ikke og vi bygget dem derfor selv. Etterhvert har vi bygget 5 maskiner og utstyrt dem med den nødvendige programvare. Arbeidet ved Hyperkubelaboratoriet har derfor faglig sett spent ganske vidt: Vi har konstruert og bygget maskiner, laget operativsystem og kommunikasjonssystem, laget en omgivelse for programmering, styring og monitorering av parallelle maskiner, arbeidet med parallelle metoder innen databaseteknikk spesielt, det har også vært arbeidet med parallelle metoder innen numerisk matematikk, bildebehandling og grafalgoritmer. De som har deltatt er prosjekt-, hovedoppgave- og dr.gradsstudenter ved IDT, SINTEF B-ansatte og ansatte ved IDT. De mest aktive utviklere i øyeblikket er Torgrim Gjelsvik, Ole John Aske, Øystein Torbjørnsen, Petter Moe, Eirik Knutsen og Olav Sandstå. Gjennom de 7 årene vi har holdt på har mer enn 50 studenter hatt oppgaver ved laboratoriet. Finansielt har NTNf vært hovedbidragsyter, IDT, ND AS, Nordisk Elektronikk AS, Eltron AS og Intel Norway AS har alle bidratt. Vi har dratt stor nytte av instrumenter og infrastruktur ved IDT.

Parallele maskiner

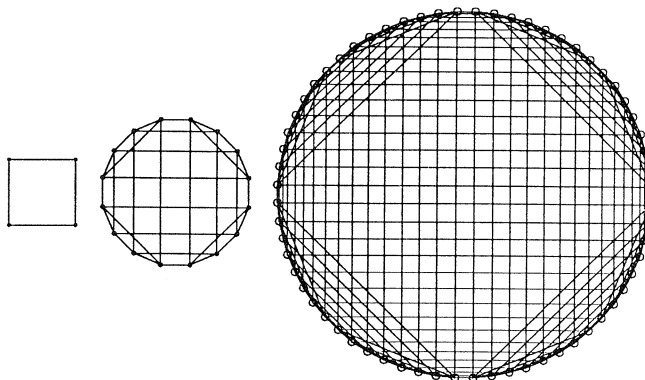
Det finnes 3 hovedtyper parallelle maskiner. Parallele maskiner med felles arbeidslager, SIMD- og MIMD-maskiner. SIMD står for Single Instruction Multiple Data, det betyr at én instruksjon styrer mange samtidige datastrømmer. Det er bare den ene instruksjonen som utføres om gangen. MIMD står for Multiple Instruction Multiple Data og betyr at det er flere instruksjonsstrømmer som utføres samtidig på sine *tilhørende* datastrømmer. Både SIMD- og MIMD-maskiner har *adskilte* arbeidslager med hvert sitt lokale adresserom.

Vi har bare konsentrert oss om MIMD-maskiner. Disse er i databasesammenheng mest interessante. Vi trenger høy kapasiteten for flytting av data. Dette oppnås ved at vi har mange

uavhengige samtidig arbeidende maskiner med egne lagermoduler. En MIMD-maskin består egentlig av en gruppe selvstendige og uavhengige maskiner som er knyttet sammen gjennom et datanett som har høy kapasitet. Uavhengighet mellom de enkelte moduler er viktig når vi skal ha feiltolerante maskiner. Maskinene må være så adskilte at selv om én maskin feiler (på verst tenkelige måte) så må den ikke trekke de andre med seg. Det andre kravet er altså at maskinene eller nodene er så tett koplet sammen at kapasiteten til utveksling av data er stor.

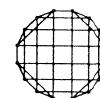
Kommunikasjonen mellom nodene

Kommunikasjonen mellom nodene må være feiltolerant og ha tilstrekkelig kapasitet som helst øker i samme takt som antall noder. Vi har brukt hyperkubisk topologi og *delt* lager mellom nabonoder i hyperkuben. Denne



Figur 1 3 hyperkuber med dimensjoner D lik 2, 4 og 6. Disse har henholdsvis 4, 16 og 64 noder. Ett nodekort er plassert i hvert hjørne i kubene. Hyperkuben er et godt kompromiss mellom pris og ytelse. Prisen er avhengig av antall linjer - $L = D \cdot 2^{D-1}$. Antall noder $N = 2^D$. For at signalveiene skal bli kortest mulig er det viktig at største avstand mellom to kort er liten. I vår 64-noders kube er sylindrens (sirkelens) diameter 20 cm.

topologien tilfredsstillende meget godt disse kravene. I en hyperkuben er det $N = 2^D$ noder. D kaller vi kubens dimensjon. Hvis vi omfordeler dataene i en kube og sprer dem med samme sannsynlighet til alle noder vil arbeidet med å omfordele et dataelement være proporsjonalt med $D/2+1$. En annen god egenskap er at etterhvert som kubens dimensjon øker vil antall alternative veier i kubene øke meget raskt. Feiltoleransen blir dermed bedre.



Navn	Aktuelt # noder	Maks. # noder	CPU	Maks. samlet ytelse		Kapasitet pr. node					
				MIPS	MFLOPS	MB	#SCSI	I/O MB/s	MIPS	MFLOPS	Disk MB
HC16-386	16	16	i386/20	64	5	2-4	1	1.5	4	0.3	144
HC16-386/V	1	16	i386/20	64	512	2-4	1	1.5	4	32	0
HC64-486	4	64	i486/25	1024	64	8-64	2	8.0	16	1.0	400 - 600

Tabell 1 En oversikt over maskinene på Hyperkubelaboratoriet og deres karakteristiske egenskaper. Samlet ytelse gjelder for full konfigurasjon.

Om maskinene

Hyperkubelaboratoriet har i dag tre maskiner som alle er operative og tilgjengelige over NTHs datanett. I betegnelse nedenfor står HC for Hyper Cube, deretter følger øvre grense for antall noder som maskinen kan ha og til sist kommer betegnelsen på nodeprosessor. Maskinene er:

- ° HC16-386
- ° HC64-486
- ° HC16-386/Vector

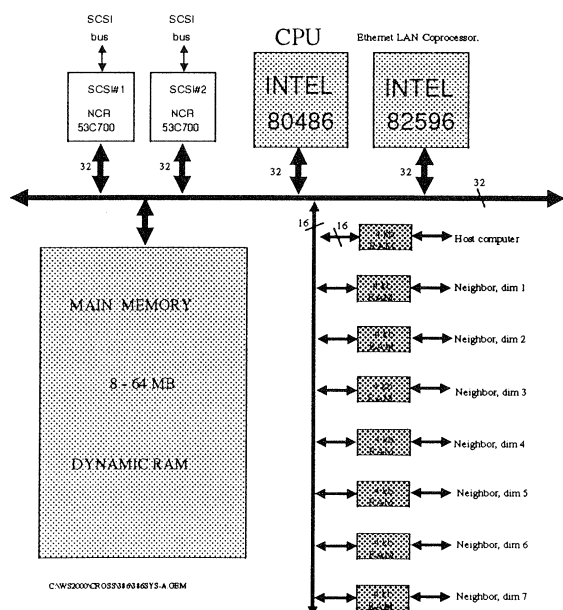
En oppsummering av de viktigste egenskapene ved maskinene er gitt i tabellen.

HC16-386 er fullt ferdigbygget og er HC-labens hovedmaskin. Overføringskapasiteten mellom to nabonoder er opp til 4 MB/s når en bruker våre standard rutiner for kommunikasjon mellom noder. Teoretisk maksimum er 6.6 MB/s. Maksimal relokeringkapasitet er målt til ca.

15 MB/s og det teoretiske maksimum er 32 MB/s. Maskinen er tilkopleet en UNIX arbeidsstasjon som fungerer som vertsmaskin. Det er en direkte forbindelse mellom alle noder og vertsmaskinen. HC16-386 nås gjennom vertsmaskinen som er kopleet til Ethernettet.

HC64-486 er under bygging og i øyeblikket har den 4 noder. Hver node har en egen Ethernet-prosessor. Hver node kan således koples direkte til nettet. Dette er nødvendig hvis en vil ha feiltolerant adgang til kuben eller en vil sikre seg adgang uten flaskehals. Teoretisk overføringskapasitet mellom to nabonoder er 11.8 MB/s. Kuben nås over Ethernettet via vertsmaskinen. Som for HC16-386 er alle noder direkte knyttet til vertsmaskinen. Hver node har to eller tre disker. Hele maskinen har 10 disker med en samlet kapasitet på 2.1 GB.

HC16-386/V er lik HC16-386, men er i tillegg utstyrt med en vektorenhet for numeriske beregninger. Vektorenheten har en maksimal ytelse på 32 MFLOPS. IDT har nettopp bevilget midler til 3 nye vektornoder. Disse antas å være operative i løpet av januar 1992.



Figur 2. Systemskisse for nodemaskin til HC64-486. Hver node har direkte kontakt med vertsmaskinen. Det er også laget en ekstra kontakt mot den 7. dimensjon slik at dette egentlig er en HC128-486. Denne forbindelsen er imidlertid ikke brukt i det hyperkubiske kommunikasjonsnettet, derfor er dette en HC64.

Grunnleggende programsystemer

Operativsystemet

Hver node er utstyrt med operativsystemet TOROS. Det har funksjoner for prosesshåndtering, lageradministrasjon, interruptkontroll, kommunikasjon mellom noder, basis disk I/O-rutiner, kommunikasjon mellom tastatur, skjerm og noder og instrumentering. Disse utgjør et subset av UNIX sine kjerne rutiner. En prosess har adgang til alle data i et program og et program kan inneholde mange prosesser. Lagerallokeringen er dynamisk og vi benytter prosessorens pagingmekanisme for å slippe fysisk flytting av segmenter i lageret. Vi benytter ikke "demand paging" eller "swapping" til disk. Via pagingmekanismen "mappes" sidene inn i de forskjellige segmenter etter behov. Dette er svært fleksibelt og vi slipper problemer med



ekstern fragmentering. Allokering og frigivelse av lager skjer ved kall til systemrutinene.

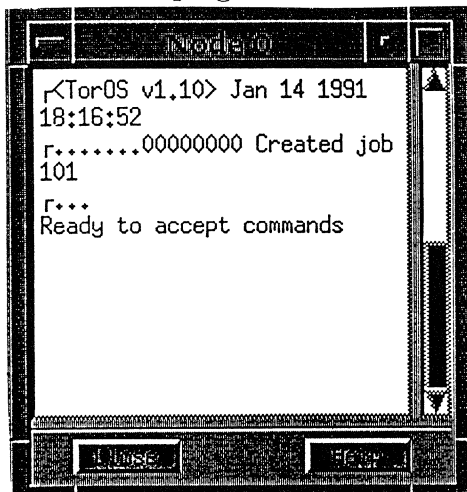
Nodene i en hyperkube er nummerert fra 0 til N-1. Kommunikasjon mellom nodene skjer ved utveksling av meldinger (send/receive). En meldingsadressere er sammensatt av node-, program- og portnummer. Meldingene rutes slik gjennom nodene at vranglås ikke kan oppstå.

Gjennom systemet for meldingsutveksling oppnås to ting: Det ene er å sende data mellom noder (riktigere: prosesser), det andre er å synkronisere samarbeidende prosesser. TOROS har et blokkbasert disk I/O-system for å lese og skrive blokker til disk. Disken er absolutt-adressert. Terminal I/O går gjennom vertsmaskinen.

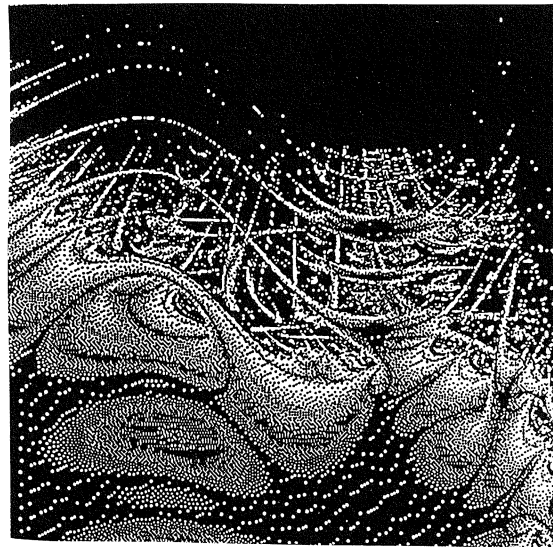
I tillegg til de vanlige operativ- og monitor-funksjoner er det lagt inn et antall tilkoplingspunkter som brukes til instrumentering og overvåking av programmer under utførelse. Disse funksjonene brukes under avlusing og forbedring av programmene. Det kan være vanskelig å finne ut hvordan et parallelt program oppfører seg uten at en har adgang til spesielle hjelpemidler. Gjennom instrumenteringen kan en få opplysninger om belastning på de forskjellige ressurser, trafikk mellom noder, sekvens av hendelser i forskjellige noder, osv.

Kontrollprogrammet

Kontrollprogrammet kjører på brukerens arbeidsstasjon. Brukergrensesnittet er basert på bruk av X Window System. Det er menystyrt og laget med vekt på brukervennlighet. Kontrollprogrammet får sine data fra nodene via kubens vertsmaskin. Vertsmaskinen mottar data fra kontroll-programmet som brukes til å



Figur 2. Tekstvindu for node nummer 0. Brukeren velger selv hvor mange tekstvindu som skal være aktive til enhver tid.



Figur 3 Eksempel på utskrift fra det grafiske vinduet. Dessverre yter ikke gjengivelsen og størrelsen rettferdighet til originalen. HC16-386 brukte 15 timer på å beregne denne fraktalen. Det tilsvarer 245 timer på en 386 PC med numerisk prosessor. Arbeidet er utført av Øystein Torbjørnsen.

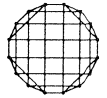
styre de enkelte nodeprogrammer og selve nodene.

I figur 2 er det vist et eksempel på vindu. Her er det åpnet et tekstvindu for node 0. All utskrift (printf i C og C++) i node 0 kommer til dette vinduet. Hvis en har behov for å skrive kommandoer eller data fra terminal må en først åpne et terminalvindu. Data som skrives inn i terminalvinduet sendes til den aktuelle node. Det er også mulig å åpne et grafisk vindu for visning av grafikk. Kontrollprogrammet har grafikk som viser hvilke noder som er aktive, hvor sterkt CPU er belastet osv. Til og med diskene slås av og på gjennom dette programmet.

Programmering

Kuben programmeres normalt i C++, C eller assembler. All programutvikling skjer på vanlig UNIX arbeidsstasjon med vanlige hjelpemidler. Det er nødvendig med en oversetter som genererer kode for Intel 386 eller Intel 486. En må også ha adgang til headerfiler som definerer de forskjellige funksjonskall i TOROS. Også sammenlenking av modulene skjer på den lokale UNIX arbeidsstasjon. Nedlasting av programmer skjer via kontrollprogrammet.

Før et program kan lastes ned må kubens allokering til den aktuelle bruker. Vi tillater ikke mange brukere på samme node samtidig, men vi tillater at mange brukere er inne samtidig på kubens noder. De har imidlertid hver sine sett av noder. Dette er *space sharing*. En bruker kan



```

#include <math.h>
#include <stdio.h>
#include <toros.h>
#include <stdlib.h>
#define N 16 // number of nodes
int MyNode;
msgaddr_t neste;
class Matrix {
    float *p; // Address of first element in matrix
public:
    int col, row; // Size of matrix
    Matrix () {col=row=0; p=0;}; // Empty Matrix
    Matrix (int, int); // n X m Matrix
    float* Allocate (int, int);
    float* E00 () {return p;};
    void Write ();
    Matrix Multiply (const Matrix);
};
Matrix::Matrix (int a, int b) {
    row=a; col=b; p=new float [a*b];
}
float* Matrix::Allocate (int m, int n) {
    row=m; col=n; p=new float [col*row];
    float* q=p; int i=col*row;
    while (i--) *q++ = 0.0;
    return p;
}
void Matrix::Write() { // Write matrix...
}
Matrix Matrix::Multiply(const Matrix B){
    Matrix C; float *c;
    if (MyNode == N-1) c = C.Allocate(row, B.row);
    float *a=p;
    for (int i=0; i<row; i++) { // Rows in A
        float *b = B.E00();
        for (int j=0; j<B.row ; j++) { // Columns in B
            float s=0.0;
            for (int k=0; k<col; k++)
                s += (*a++) * (*b++);
            a -= col; // reuse same row in A
            if (MyNode) s+= *(int*)getmsg(WAITFOREVER);
            if (MyNode != N-1){
                if (!sendmsg(&s, 2, neste, WAITFOREVER))
                    printf("\n Kommunikasjonsfeil");
            } else *c++ = s;
        }
        a += col; // take next row in A
    }
    return C;
}
main () {
    Matrix a(100,10), b(30, 10), c;
    msgaddr_t* p = getmyaddr();
    neste.job = p->job;
    neste.classno = p->classno;
    neste.node = (MyNode=p->node) + 1;
    MyNode = neste.node;
    // .. legg verdier inn i a og b
    c = a.Multiply(b);
    if (MyNode==N-1) c.Write();
}

```

Figur 4 Skisse av program for parallell matrisemultiplikasjon..
Matrisen B er lagret transponert. getmyaddr, getmsg og sendmsg er funksjoner i TOROS.

allokere en subkube med 1, 2, 4, 8 eller 16 noder (og videre oppover når vi får større kuber). Brukeren kjører alltid på en *virtuell* hyperkube hvor nodene nummereres fra 0.

Hvordan få adgang til Hyperkuben?

De som vil kjøre kuben må ha et spesielt passord. Ta kontakt med Ole John Aske (oleja@idt.unit.no) for å få flere opplysninger. Systemet har vært under utvikling helt opp til nå og brukervennlig dokumentasjon er ennå mangelvare. Vi arbeider med saken og vil også holde kurs om systemet på nyåret.

Eksempelprogram - Parallell matrisemultiplikasjon.

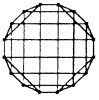
For å gi en indikasjon på hvordan kubene programmeres skal vi skissere et program for parallell multiplikasjon av matriser. Den viktigste hindringen for utnyttelsen av parallelle maskiner er at løsninger for monomaskiner ikke direkte kan overføres til en parallell maskin. Det er heller ingen mekanisk omforming fra et monomaskinprogram til et multimas-kinprogram som gjelder universelt. Oppgaven må løses på en ny måte. En må utvikle generelle løsninger for N maskiner. Som regel består løsningen av N like og samarbeidende prosesser. Data må også fordeles mellom de samme prosessene. Det er imidlertid ingen nødvendighet at alle prosesser er like.

Matrisemultiplikasjon er enkel å parallellisere. La $C=A*B$. Hvert element i resultatmatrisen C , c_{ij} , er vektorproduktet av rad i og kolonne j . For å fordele lasten lagrer vi m kolonner av A og m rader av B på hver node. Dette gjør at hver prosessor kan utføre sin del av vektorproduktet. Nodene er ordnet i rekkefølge 0 til $N-1$. Node k venter til den mottar resultatet fra node $k-1$, deretter adderes dette med eget bidrag før summen sendes til node $k+1$. Nodene 0 og $N-1$ er spesialtilfeller. Node 0 sender sitt resultat til node 1. Node $N-1$ summerer og lagrer i C i stedet for å sende summen videre. I figur 4 er det vist en skisse av programmet. Programmet er skrevet i C++.

Hvis en ønsker resultatmatrisen fordelt på nodene etter et bestemt mønster styres det gjennom valget av startnode for summeringen. Matrisemultiplikasjon lar seg parallellisere nesten 100%. Sending av delresultat mellom naborodene tar en svært liten del av tiden.

Forskningsresultater

Innen databaseteknikk har vi særlig arbeidet med parallelle metoder for å oppnå høy ytelse. Her har vi også meget gode resultater å vise til. Vårt miljø er et av de ledende når det gjelder parallell sortering og parallell relasjon-



salgebra. Relasjonsalgebra er "instruksjonssett" i et hvert databasesystem som bruker relasjonsmodellen. Relasjonsmodellen kan også brukes som "maskinmodell" selv om brukergrenssnittet er objektorientert.

For å illustrere ytelse kan vi bruke følgende eksempel. La oss anta at vi har et register over alle nordmenn med fornavn, etternavn, adresse, osv. Gi hver person 100 byte. Med HC64-486 vil vi kunne svare på "ethvert" spørsmål basert på disse opplysningene i løpet av 8 sekunder. Eksempel på spørsmål: "Hvilke 1000 fornavn er mest brukt? Skriv ut navnene i sortert orden sammen med antall ganger navnet er brukt". Med 1 disk pr. node er tiden begrenset av hvor lang tid det tar å lese dataene ut fra disken. Med to disker vil vi kunne svare i løpet av 4 sekunder. Under 3 sekunder kommer vi ikke, da er vi CPU-begrenset.

Teoretisk er det også gjort mye i på høykapasitets transaksjonssystemer og på systemer med stor tilgjengelighet og toleranse mot feil. På disse områdene mangler vi imidlertid eksperimentelle erfaringer.

Forskningsoppgaver framover.

Utviklingen av MultiBase

Navnet MultiBase henspeiler på to forhold:

- ° Systemet kjører på en multimasin
- ° MultiBase integrerer "alle" typer data.

I tillegg til de vanlige formaterte datatypene har vi fritekst, geometri og topologi, bilder og lyd og regler. Vi vil eksperimentere med å integrere alle disse datatyper i ett system. Det krever nye lagringssystemer og nye brukergrensnitt. Integrasjon er lønnsomt. Det er tungvint å håndtere kart i ett system og eiendomsdata i et annet, f.eks.

Vi vil fortsette forskningen for å oppnå høye ytelser i parallelle systemer. Spesielt må vi skaffe oss mer eksperimentell erfaring med mange samtidige brukere. Parallelle systemer gir helt spesielle muligheter for å gjøre systemene tolerante mot feil. Vi kan lage systemer som svært sjelden er helt ute av drift. Selv under vedlikehold og utskifting av moduler skal systemet beholde sin funksjonalitet. For å oppnå slike egenskaper må data ligge lagret minst to steder - også når moduler har falt ut. Utvikling av feiltolerante systemer er meget krevende og vi må videreutvikle teorien hånd i hånd med at vi gjør eksperimentelle erfaringer.

Sentraliserte systemer vil vinne fram.

På databasesiden vil vi få en svingning fra distribuerte systemer tilbake til sentraliserte systemer. Ny teknologi på to områder gjør sentralisering mer attraktiv. Optiske fiber i datakommunikasjon har så stor kapasitet at avstand spiller liten rolle. Supernettet mellom universitetene i Norge har en kapasitet på 34 Mb/s og vil øke til 800 Mb/s innen to til tre år. Det andre området er bruken av parallelle maskiner. Parallellteknologien gjør det mulig å lage sentraliserte systemer med meget høy kapasitet. Samtidig er enhetsprisen pr. utført operasjon tilnærmet konstant. Det er bedre å ha samlet mye ytelse på ett sted enn å ha tilsvarende ytelse spredt utover. Stikkord i denne sammenheng er: dimensjonerende toppytelse, enklere og sikrere systemer, forenkling av drift og ajourhold, billigere lagring. Vi kan trenge billigere lagring. I bibliotekssystemer f.eks. lagrer vi i dag bare informasjon om bøkene. Skal vi lese selve boken må den fysisk hentes fram. Informasjonen om bøkene ved universitetsbibliotekene i Norge utgjør bare noen få GB, ca. 400 byte pr. bok. Lagring av bare én bokside som et bilde krever mellom 10 og 100 KB avhengig av komprimering.

Parallelle numeriske beregninger

Vi tror at det er nisjer innen tungregning hvor det er fornuftig å bruke egenutviklet utstyr, bl.a. der hvor maskinen kommer til oppgaven. Det er relativt enkelt å lage parallelle maskiner for tungregning, heller ikke trenger vi så mye programvare som f.eks. i databasesammenheng for å starte utnyttelsen av dem. Vi har nettopp forbedret programmeringsverktøyene for vektorenheten. Den kan nå programmeres i et C-lignende språk. Gjennom en vektornode skaffer vi oss i dag 32 MFLOPS for ca. 30000 kr. Det er rimelig tungregningskapasitet. Vi har konkrete planer om å lage en ny vektornode basert på Intels 860. Vi får da adgang til alle verktøy for programmering av denne prosessoren. Hver node kan skaleres til å yte mellom 100 og 400 MFLOPS. Ved å bruke vår etablerte teknologi for sammenkopling av 64 noder vil det gi en toppytelse på 25 GFLOPS. Vi prøver å få i stand utviklingsprosjekt der vi kan kombinere numeriske beregninger med behov for håndtering av store datamengder.

Kjell Bratbergsengen
Institutt for datateknikk og telematikk
kjellb@idt.unit.no

SUPERNETT

Supernett er et fellesprosjekt mellom Televerket og UNINETT for etablering av et høyhastighet datanett. Dette prosjektet har pågått i et års tid, og fra 1/1 1992 blir dette nettet operativt mellom universitetene i Trondheim, Oslo, Bergen og Tromsø.

Mellom disse universitetene er det etablert faste linjer med maksimal hastighet 34 Mbit/s. På universitetene møter dette nettet nyetablerte FDDI (Fiber Distributed Data Interchange) nett med maksimal hastighet 100 Mbit/s. Slike FDDI nett er det nå ved UiB, UiO og UNIT. UiB var først ute og har det største FDDI nettet. I Trondheim er det CRAY som nå er koplet til FDDI nettet, og bruken av CRAY er en viktig pådriver for etableringen av et slikt "supernett".

Målet med et slikt "supernett" er å kunne flytte store datamengder kjapt mellom universitetene, og å eksperimentere med nye applikasjoner som medfører forflytning av store datamengder. Applikasjoner som en nå tenker å prøve ut, er multimedia og fjernundervisning.

Også andre interessante forskningsprosjekt vil trolig etableres i kjølevannet av den tekniske realiseringen av høyhastighetsnettet, og flere har allerede meldt sin interesse i å delta i dette.

For både Televerket og UNINETT er i tillegg prosjektet viktig for å få erfaring i drift av høyhastighetskommunikasjon og FDDI teknikk. For UNINETT er det viktig å ha god kontakt og et konstruktivt samarbeid med Televerket. Gjennom samarbeidet med UNINETT får Televerket erfaring med for dem ukjent teknologi og ukjente tjenester.

Nettet er basert på ruterteknologi og det er nå TCP/IP kommunikasjonsprotokollene som brukes.

Det er viktig å merke seg at UNINETTs stamnett slik vi kjenner det i dag ikke vil bli erstattet av Supernett - ihvertfall ikke i første omgang. De tradisjonelle UNINETT tjenestene vil fortsatt gå over det "gamle" nettet.

Dette er ny og spennende satsing, og det er verd å merke seg at dette prosjektet fører til at Norge er tidlig ute med slike hastigheter i datanettet!

Knut L Vik

STATLIB

programbibliotek for statistikkprogram

STATLIB er et tilbud som har dukket opp på datanettet. Det er en tjeneste for distribusjon av statistikkprogramvare og annet interessant for statistikkfaget. Tjenesten holder til ved Carnigie Mellon universitetet i USA.

Program kan hentes pr. e-post eller pr. anonym FTP.

E-post adressen er statlib@lib.stat.cmu.edu. Statlib bruker samme programvaren for å ta imot e-post meldinger og sende ut programmer som Netlib (matematikkprogramvare) (se RUN-NYTT nr. 2, 1990, s. 11). Start med å sende kommandoen `help` ev. `send index` - da får vi generell informasjon og en hovedinnholdsliste.

Adressen for anonym FTP er [lib.stat.cmu.edu](ftp://lib.stat.cmu.edu) (128.2.241.142).

Fra biblioteket kan vi f.eks. hente Minitab makroer og stoff knyttet til programmet SAS. Hva det siste er, får vi en oversikt over ved å sende meldingen: `send sas from general`.

Nyheter om dette biblioteket sendes ut til News gruppen sci.math.stat.

I UNINETTINFO, katalog matematikk, har vi lagret teksten vi får ved å sende `help`, samt to innlegg i sci.math.stat. om nyheter i STATLIB.

Knut L Vik

