

QRUN • QNYTT

QNFORMASJONSORGAN FOR QUNIT,
QEGNESENTRET VED QUNIVERSITETET I QTRONDHEIM

QR. 2
QÅRG. 9

10. MAI 1982



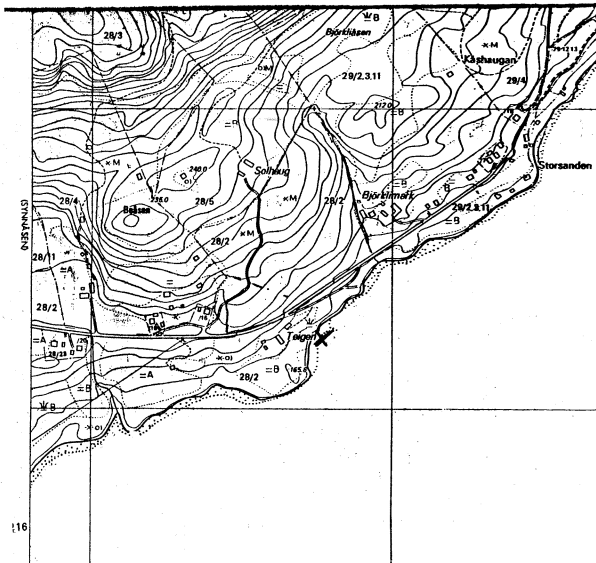
De unge, fremadstormende databehandlere tar over, de eldre må stille seg bakerst i køen.

Se side 3.

UNIVAC'en som dro til Klæbu



Her ved Teigen, i enden av Selbusjøen, er det planlagt å reise et teknisk museum for Trøndelagsregionen.



Det var nok flere som med tungt hjerte så UNIVAC 1108's endeligt i møte. Etter at den ble "satt ut av spill" sommeren 1980, har den stått nedpakket i plast i kjelleren på Lerkendal. Like etter nyttår var det ikke bønn lenger, 1108 måtte fjernes. Skulle virkelig denne fullt brukbare arbeidshest kannibaliseres? Var det ikke snakk om at det skulle opprettes et teknisk museum i Klæbu? Det ble tatt en telefon og forespurt om interessen for datamaskinelle antikviteter.

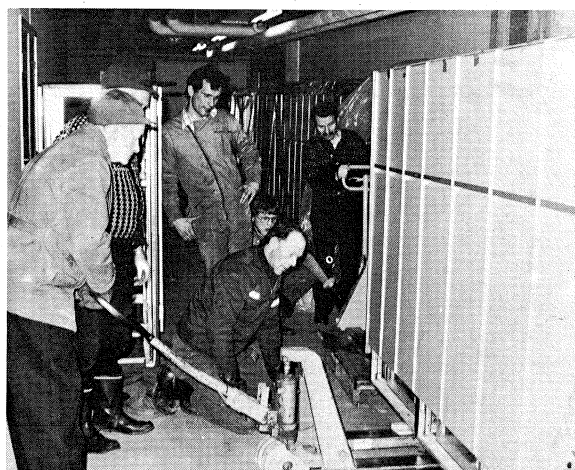
Svaret fra Klæbu kom uten nøling: ja, datamaskiner kunne man ta i mot, kanskje kunne man til og med tenke seg og sette den i drift og bruke den til registrering av museumsgjenstander. Ole Svaan, formann i museumsnemda, satte seg i sving og fikk ordnet midlertidig leieavtale for UNIVAC'en i Svean kraftstasjon. Kort tid etter kom en dugnadsgjeng fra Klæbu med lastebil og kran, og i løpet av en kveld var hele anlegget forflyttet til et nytt midlertidig hvilested. - Mæsta som når vi fløtta trøskverk i gammel daggå det her, var en av kommentarene vi hørte.

UNIVAC til teknisk museum i Klæbu, det var flere vantro bemerkninger til dette ryktet. Noen ble visstnok til og med nødt til å

spandere kake etter å ha tapt veddemål på saken. Men klæbyggene mener alvor. En komite har vært i arbeid ett års tid, fylkeskonservatoren er representert i komiteen, og Norsk Teknisk Museum har vært konsultert i sakens anledning.

Håndgivelse på et område på 40 - 50 mål er ordnet ved Klæbu Bygdemuseum på Brøttem, idyllisk beliggende ved enden av Selbusjøen. Her har det vært brygge for båten "Kong Sverre" som tidligere gikk i fast rute på Selbusjøen. Denne vil bli forsøkt hentet hjem igjen fra Tromsø og istandsatt som en del av museumsanlegget. Landbruksmaskineri og veteranbiler er også sikret for fremtidig fremvisning, og nå også en amerikansk datamaskinveteran.

Det er tre mil til Brøttem, og det har vært ymtet om at dette stedet kan bli "Trøndelags Høvikodden", dit man drar på søndagstur med barnebarn og skjelvende peker på noen store grå skap og sier: Der er gamle 1108, det var på den datamaskinen jeg kjørte mitt første FORTRAN-program".



Flere av UNIVAC-skapene veide godt over et tonn, og de måtte tas med list og lempe. Men klæbyggene hadde trening i å flytte treskeverk, så de klarte oppgaven fort og effektivt.

RUN-NYTT

Redaktør : Anne B. Reitan Sivertsen
RUNIT

7034 Trondheim-NTH

Tlf. 075 93027

Redaksjons-
assistent : Arne Asphjell
Tlf. 075 93048

Utkommer : 4 nummer pr. år

Abonnement: Gratis ved henvendelse til
RUNIT's ekspedisjoner eller
redaksjonen.

Bidrag : Mottas med takk!

ETTERTRYKK TILLATT NÅR KILDE ER ANGITT.

INNHOOLD:

UNIVAC'en som dro til Klæbu	side 2
Leder:	
De unge fremadstormende databehandlere	3
Er programmet for stort?	4
Reduserte priser i sommermånedene	5
Samlefattura	5
Spesialpris for egenfinansiert forskning	5
SCSS - et interaktivt statistikkprogram	6
Enkel sortering i ASCII-FORTRAN	10
Ny SPSS-versjon	11
PRINT - et nyttig program	11
Effektiv testing med ASCII FORTRAN	12
Spar CPU-tid og penger	15
Studentkjøringer med spesielle behov	15
POLYDOC	16
COBOL programgenerator	16
Discrete Event Modelling on SIMULA	16
SINCLAIR ZX-81	17
EDITOR-TRIKS	18
@-TRIKS	20
Nye UTS-terminaler på sal 212	21
Overføring av filer via NORD-nett	22
COST-prosessoren er utvidet	23
Nye sentralenheter på PDP-11	24
Publikasjoner fra RUNIT	25
Haakon Wiig har datasenter på hybelen	26
Dataoverføring internt i UNIT-miljøet	29
RUNIT gruppeforum	30
Gruppe for brukerkontakt og programvare	30
Ny UNIVAC tekstbehandling	31
UNIVAC 1100/62 vokser	31
Tastaturet - dataalderens anakronisme	32
Løsning kryssord i nr.1	33
INDEX Tidligere artikler i RUN-NYTT	34
Kryssordvinnere	37
Hvor er RUNIT	38
RUNITs veiledningstjeneste	39
Kryssord	40

DE UNGE, FREMADSTORMENDE DATABEHANDLERE

Databehandling er et merkelig fag. For det ene fordi utviklingen skjer så fort at hvis man ikke passer på å holde kunnskapene vedlike og kontinuerlig etterutdanne seg, blir man fort ubrukelig som databehandler. Innen man blir akterutseilt må man i det minste sørge for å stige i gradene så man kan administrere andre som kan litt om EDB.

For det andre er faget merkelig fordi det ser ut til at det finnes naturlige databehandlingstalenter, folk som kan skrive lure programmer i løpet av en natt, og attpå til virker de med en gang. De er bokstavelig talt født med tastatur mellom hendene.

Haakon Wiig som vi skriver om i dette nummer er en av disse. Og han er ikke den eneste i UNIT-miljøet. På forelesningen om "Personal Computing" sist høst hadde flere dataentusiastiske studenter foredrag.

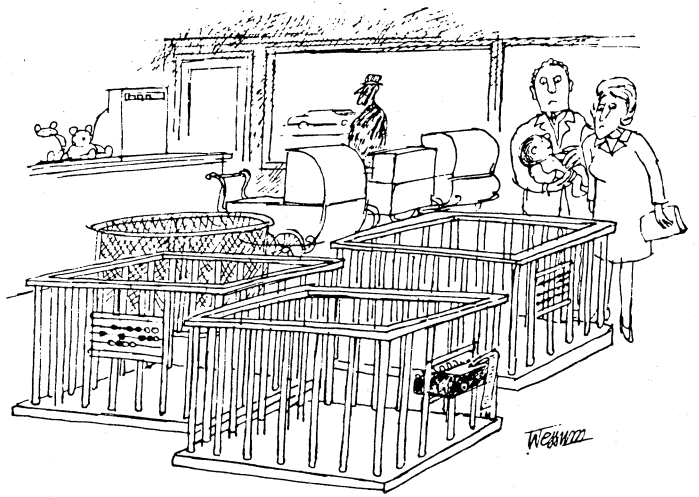
Det er ikke nødvendigvis på datalinjen man finner de ivrigste databehandlerne. De er like gjerne på elektro, maskin eller fysikk. Har man først fått "databasillen" i blodet lar man seg ikke stoppe av det nåløyet det er å komme inn på datalinjen. Da er løsningen å søke på en av de andre linjene og samle med seg så mange detaljer som mulig. Vi tror disse dataingeniørene fra sidelinjene er vel så populære i arbeidslivet som de fra datalinjen. Mange arbeidsgivere foretrekker folk som har et "ingeniørfag" i tillegg til at de behersker databehandlingen. Rekrutteringen til datayrkene ser ikke ut til å være noe problem, det er forbausende å se hvor interessert yngre årganger er i databehandling. Forbausende er det også å se hvor lett de tar det. De kaster seg over tastaturene uten hemninger og kan i løpet av kort tid lage sine egne programmer. Det er bare to ting som bekymrer oss i denne forbindelse:

FORTS. NESTE S.

DE UNGE forts.

Grunnskolens manglende muligheter til å tilby elementær dataundervisning. Til og med i den videregående skole er dataundervisning mangelvare. Dette legger ekstra press på universitets- og høyskolenivået, mye av grunnutdanningen må skje på dette planet.

Vår andre bekymring er hvordan det vil være å eldes i denne bransjen. Når utviklingen går så raskt kreves det som nevnt innledningsvis stor innsats å holde seg ajour. Bli de gamle dataårganger sand i maskineriet som blir fullstendig jekket ut av de unge og fremadstormende?



ER PROGRAMMET FOR STORT?

Mange brukere har i den senere tid opplevd å få feilmeldingen:

```
PROGRAM EXCEEDED MEMORY QUOTA
```

Årsaken til dette er at programmet er større enn det som tillates kjørt på det aktuelle kjørenr. Grensene er:

	Satsvist	Interaktivt
studenter	60K	30K
ansatte	"ubegrenset"	60K

I tidsrommet 16.00 til 07.59 gjelder samme grenser for studenter som for ansatte.

Programstørrelsen for et program går fram av utskriften fra lenking (@MAP). Tidlig i denne utskriften finner en noen programlinjer som f.eks kan se slik ut:

```
ADDRESS LIMITS      001000 067640      28065 IBANK WORDS DECIMAL
                    070000 206335      40158 DBANK WORDS DECIMAL
```

Total programstørrelse for dette programmet er $28065+40158=68223$ ord eller vel 68K ord. Denne brukeren fikk feilmeldingen som er referert innledningsvis selv om han hadde fått øket grensene for interaktiv kjøring til 70K.

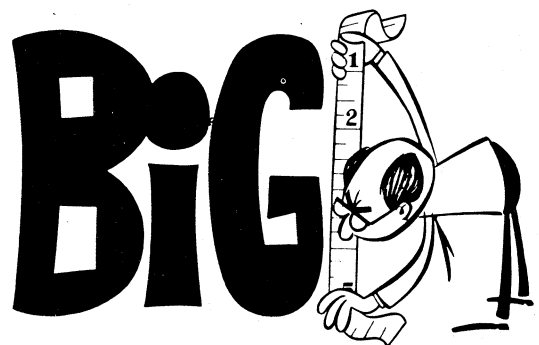
Årsakene til dette er at f. eks. ASCII FORTRAN-programmer under utførelse refererer til flere såkalte COMMON-banker. I praksis betyr det at en del av koden som skal utføres ikke er lenket sammen med selve programmet, men ligger som felles moduler som kan brukes av flere programmer samtidig.

I COMMON-bankene ligger f. eks. koden som sørger for inn- og utmatning og de matematiske funksjoner som er definert i FORTRAN.

I tillegg til programstørrelsen som kan finnes i MAP-listingen kommer pr. idag halve størrelsen av alle COMMON-banker som blir referert.

I programeksemplet som ble nevnt ovenfor dreide dette seg om 8 - 10k ord. Bruk av COMMON-bankede bibliotek er relativt ny ved RUNIT og det vil bli vurdert hvor stor del av COMMON-bankene som skal telles med i brukers programstørrelse når vi får noe mer erfaring med denne type bibliotek.

Paul Rusten.



REDUSERTE PRISER I SOMMERMÅNEDENE

På grunn av ledig kapasitet i sommermånedene vil RUNIT også i år, som et tilbud til brukere med CPU-krevende jobber, forandre prismekanismen i perioden 28 juni - 29 august 1982. Ordningen omfatter typekode R, F, L, U, E og N etter følgende regler:

- For alle eksekveringer (@XQT, NB! ett jobbtrinn) som krever mer enn 4 min. CPU-tid, vil CPU-tid utover 4 min. være gratis. Det er vanlige satser på de 4 første min.
- Maksimal kjøretid for en jobb bør av driftshensyn ikke overskride 30 CPU-min. (For hvert nye 30 min. vil det bli belastet for nye 4 min.)
- Jobbene vil bli kjørt på kvelds- og natt-tid. Kjøringer som ikke er ferdige til kl. 0700 påfølgende dag kan bli kuttet uten at dette gir grunn for reklamasjon.
- Tilbudet omfatter kun CPU-tid. De andre kostnader (fil-leie, magnetbånd etc.) blir fakturert etter vanlige satser.

Brukere som har svært ressurskrevende oppgaver kan, nå som ellers i året, ta kontakt med RUNIT for avtale om eventuelle særordninger.

SAMLEFAKTURA

For datamaskinbruk har RUNIT nå tatt i bruk samle faktura. D.v.s. at en faktura kan inneholde flere kjørenummer, med en linje pr. nummer, uten de detaljerte opplysninger en vanligvis får. Har en likevel bruk for detaljene, er det selvfølgelig mulig å få spesifikasjon som tidligere.

For UNIT-brukerne kommer automatisk alle kjørenummer med fakturabeløp mindre enn kr. 20 på samle faktura til instituttet. Hvis summen for ett institutt ikke overstiger kr 100, blir data lagret til neste måned, og faktura lages hvis summen blir stor nok osv.. Akkumulering foregår over maksimalt 3 måneder.

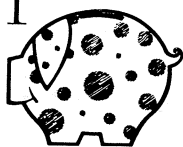
Beløpsgrensene vil vurderes i tiden fremover og kan endres uten varsel. For mars 1982 betydde samle faktura at antall fakturaer sank fra normalt 600 til 350, d.v.s. betydelig redusert arbeidsmengde for ut- sending og oppfølging.

Som en ny tjeneste tilbys nå også muligheten til å angi at spesifiserte kjørenummer skal ut på samle faktura uansett beløp (unntak fra dette er SINTEF - avdelinger og enkelte eksterne brukere). Dette vil senere måtte oppgis ved rekvirering av kjørenummer, men kan nå bestilles hos Ingebjørg Eggen (tlf 2943).

RUNIT håper flest mulig vil benytte seg av dette tilbudet.

Bjørn Gifstad

SPELIALPRIS FOR EGENFINANSIERT FORSKNING



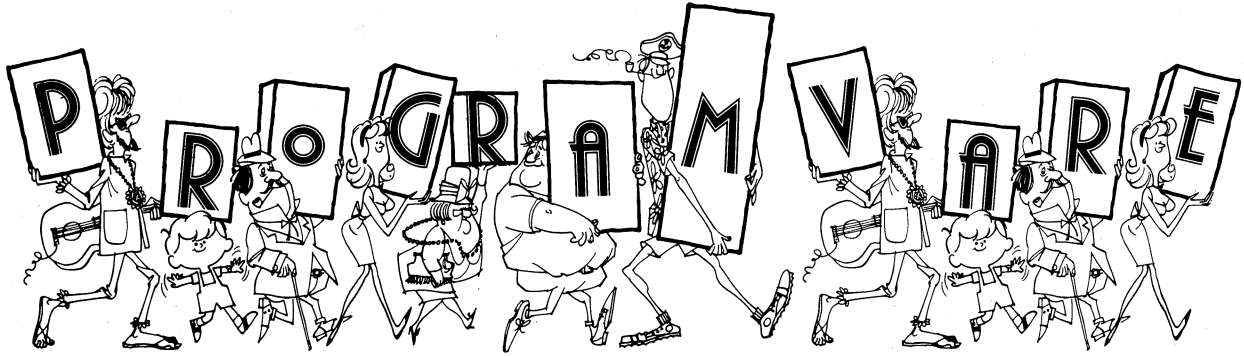
RUNIT's styre har gjort vedtak om en prøveordning som muliggjør redusert pris for datamaskinbruk knyttet til forskningsinstituttens egenfinansierte forskning (d.v.s. forskningsoppgaver uten noen ekstern finansieringskilde, E- prosjekter). I praksis skal dette gjennomføres ved at forskningsinstituttene for hvert enkelt prosjekt av denne type sender en skriftlig søknad til RUNIT. Søknaden må inneholde informasjon om prosjektets hensikt, totale omfang, et anslag for datamaskinbruk og hva slags ressurser som skal benyttes.

Priser for denne type oppgaver blir i første omgang identisk med prisene for UNIT-finansiert forskning. RUNIT vil imidlertid forbeholde seg retten til å endre dette når en får samlet statistikk over den totale belastning og fordeling mellom de ulike forskningsinstitutter som denne type kjøring vil medføre.

Typekode for denne type kjøring er H. Kjøringene blir avregnet etter samme satser som typekode F.

Kjøring på typekode H tillates som før nevnt bare etter spesiell søknad, som behandles av Rolf Bysting (tlf. 2940) eller Bjørn Gifstad (tlf. 2966).

Henveler må derfor rettes til dem, og alle relevante opplysninger må om nødvendig fremlegges skriftlig. Tillatelsene vil normalt være tidsbegrenset og vil være gjenstand for oppfølging.



SCSS - Et interaktivt statistikkprogram

I dette og følgende nummer av RUN-NYTT vil det bli gitt en kortfattet innføring i bruk av SCSS. Innføringen vil være ledsaget av en del eksempler.

1. HVA ER SCSS?

SCSS står for SPSS Conversational Statistical System, og er altså en interaktiv statistikkpakke fra firmaet SCSS inc. i USA. Men SCSS er ikke, som mange tror, en interaktiv versjon av SPSS. Det er et helt nytt produkt og minner lite om SPSS. Tidligere SPSS-brukere har likevel den fordel at de kjenner begreper som variable, "VAR LABELS", "VALUE LABELS", "MISSING VALUES", etc. Ellers er eneste kopling til SPSS den at SPSS både kan lese og skrive SCSS-filer. (Denne mekanismen virker ikke i vår nåværende SPSS-versjon, men dette håper vi å få rettet på så snart som mulig).

Full beskrivelse av SCSS finnes i håndboka: "A user's guide to the SCSS conversational system SCSS" som er utgitt på McGraw Hill forlag. Denne kan kjøpes gjennom TAPIR.

Ellers er SCSS meget godt utbygget med hjelpefunksjoner slik at en, etter å ha satt seg noe inn i virkemåten til SCSS, i stor grad kan klare seg uten håndbok.

2. FILER

Kjernen i et SCSS-opplegg er en såkalt SCSS-"masterfil". Etter at "masterfilen" er laget vil det ikke være mulig å endre denne. Hver bruker som skal analysere data har sin egen arbeidsfil (workfile). Det er selvsagt ingen ting i veien for at flere brukere kan ha en arbeidsfil felles, eller at en bruker kan ha flere arbeidsfiler.

Innholdet i en arbeidsfil kan være beskrivelse av omkodinger, beregning av nye variable, korrelasjonsmatriser o.l. Resultatet av omkodinger blir ikke lagret i arbeidsfil eller "masterfil" slik en er vant med fra SPSS. Det er kun beskrivelse av hvordan omkodingen skal foretas som lagres. Når en refererer til en variabel hentes originaldata fra "masterfilen". Hvis det så er spesifisert omkodinger på denne variabelen, hentes beskrivelsen av omkodingen fra arbeidsfilen. Selve omkodingen blir altså foretatt når variabelen skal brukes. Fordelen ved denne måten er:

- * Det finnes alltid en beskrivelse av hvilke omkodinger som er spesifisert for hver variabel
- * De opprinnelige data blir aldri ødelagt

Brukeren behøver vanligvis ikke å tilordne noen filer før en SCSS-kjøring starter. SCSS vil stille spørsmål om filnavn på "masterfil" og arbeidsfil og selv tilordne og eventuelt katalogisere disse. Det anbefales på det sterkeste alltid å angi komplette filnavn som "qualifier*filnavn".

3. HVORDAN LAGE EN SCSS-"MASTERFIL"?

En SCSS-"masterfil" kan lages på 3 ulike måter:

- a) Ved å benytte setningen "SAVE SCSS" i en SPSS-kjøring. (Virker foreløpig ikke i vår SPSS-versjon).
- b) Benytte SCSS's fildefinisjons- og data-transponeringsrutiner. Data forutsettes her å ligge på en fil eller et element på forhånd.

- c) Benytte de interaktive fildefinisjons- og dataregistreringsmuligheter som finnes i SCSS.

4. TERMINALDIALOG OG HJELPEFUNKSJONER

Brukeren ledes gjennom SCSS-kjøringen ved spørsmål (PROMPTS) fra SPSS. Ved oppstart- ing av en SCSS-kjøring får brukeren spørsmål om i hvilken form spørsmålene ønskes:

- Svært ordrik med angivelse av svaralternativ og eventuelle standard svar (VERBOSE)
- Normal (NORMAL)
- Svært kortfattet (TERSE).

Den første formen egner seg best for uvante brukere, mens den siste er beregnet på de mer avanserte. Det er også mulig å skifte mellom de ulike formene i en og samme SCSS-kjøring.

Den vanlige formen er altså at SCSS stiller spørsmålene og brukeren svarer. Brukeren kan imidlertid selv overta styringen når dette ønskes. Det gjøres ved at man i stedet for å svare på spørsmålet fra SCSS selv skriver en kommando-linje (PREEMPT). En kommandolinje kjennetegnes ved at den starter med '/' i første kolonne.

Det finnes to typer av kommandoer som brukeren kan benytte:

- a) Vanlige spørsmål som også SCSS vil stille
- a) Kommandoer der SCSS aldri vil stille tilsvarende spørsmål.

Ved å bruke kommandoer av type a) kan brukeren avbryte spørsmålssekvensen til SCSS og lede denne over i et annet spor.

F.eks. hvis brukeren på spørsmålet PROCEDURE? har svart CROSSTABS, vil SCSS innlede en sekvens med spørsmål spesielt for denne statistikk rutinen. Hvis brukeren på et senere tidspunkt skulle finne ut at det var rutinen UNIVARIATE som skulle vært benyttet, kan dette gjøres ved kommandoer /PROCEDURE eller /PROCEDURE UNIVARIATE. Det første alternativet medfører at SCSS stiller spørsmålet PROCEDURE? på nytt, mens den siste varianten medfører at SCSS starter spørsmålsrunden et trinn inne i UNIVARIATE-prosedyren.

Det finnes også en serie kommandoer som kan benyttes av brukeren, men som er av en slik art at SCSS selv ikke vil stille tilsvarende spørsmål. Denne typen kommandoer kan igjen deles i to grupper. I den første gruppen finner vi kommandoer som /HELP og /EXPLAIN. Denne typen kommandoer avbryter ikke spørresekvensen til SCSS. Dersom brukeren på spørsmål PROCEDURE? svarer med /HELP, vil en få listet ut svaralternativene. Deretter stiller SCSS spørsmålet PROCEDURE? på nytt. En annen gruppe kommandoer av type b) der vi f.eks. finner /REVISE og /COMPUTE vil avbryte den pågående dialog, og SCSS vil innlede en ny sekvens spørsmål.

5. EKSEMPEL

Nedenfor vises et komplett eksempel der en SCSS-"masterfil" blir definert ved hjelp av interaktive fildefinisjons- og dataregistreringsmuligheter som finnes i SCSS. I dette eksemplet er det angitt linjenummer for hver linje i dialogen. Disse er "kunstig" laget i dette eksemplet og vil ikke være tilstede i en vanlig terminaldialog. Alle linjer som starter med ">" er skrevet av bruker. Resten er utskrift fra SCSS. Linjer med "> <CR>" indikerer blankt svar (RETURN) av brukeren. I kommentarene som følger refereres det til linjenummer i eksemplet.

- 6: Ønsker ingen innledende forklaringer.
- 8: Ønsker hjelp.
- 16: Velger ordrik dialog.
- 18: Har ingen arbeidsfil fra før.
- 20: Spesifiserer komplett filnavn på ny arbeidsfil.
- 23: Spesifiserer RUNIT*SCSSM-1 som "masterfil". Får feilmelding på dette da denne filen ikke gjenkjennes som en SCSS-"masterfil".
- 30: Be om hjelp med henblikk på "masterfilen".
- 38: Angir at en ikke har noen "masterfil" fra før.
- 55: Spesifiserer at tillatte verdier for identifikasjon er 1 - 100.
- 57: Svarer blankt på VALUE? for å angi at spesifisering er ferdig for denne variabelen.

SCSS forts.

- 65: Spesifiserer "VALUE LABELS" for yrke.
- 87: Feilaktig spesifikasjon av "VALUE LABELS" for en variabel.
- 105: Blankt svar på "VARIABLE NAME?" angir slutt på datadefinisjonsfasen.
- 115: Ønsker liste over variabelnavn.
- 122: Forstår ikke spørsmålet COLLECT? og gjør et forsøk. Får en feilmelding og ber senere om hjelp.
- 137: Svarer blankt for å starte data-registrering for første enhet.
- 139: Angir variabelverdier for samtlige 4 variable på en linje, og SCSS går da videre med enhet nr. 2.
- 141: Angir svar på id og yrke. SCSS spør da i neste runde om lønn for samme person.
- 161: Er nå ferdig med dataregistrering, men har mulighet til å rette eventuelle feilregistreringer. Denne muligheten blir benyttet i dette eksemplet.
- 181: Angir komplett filnavn for SCSS-"masterfilen".
- 189: Skal nå starte analysering og må spesifisere arbeidsfil på nytt. Her benyttes samme fil som under data-definisjonsfasen.
- 204: Velger prosedyren UNIVARIATE. Mer forklaring om de enkelte prosedyrene kunne en her fått ved f.eks. /EXPLAIN UNIVARIATE.
- 208: Ønsker diagram med antall enheter for hver registrert verdi.
- 236: Spesifiserer at jeg ønsker en ny spesifisering for prosedyren UNIVARIATE.
- 242: Angrer på spesifikasjonene som ble gitt på spørsmålet CELLS? på linje 240. Spesifiserer dette på nytt ved en /CELLS-kommando.
- Innføringen i SCSS vil fortsette i neste nummer av RUN-NYTT. Da vil det bl.a. bli gitt eksempler på omkodning av variable og beregning av nye variable.

Paul Rusten

Eksempel:

```

1: @RUNIT*STAT.SCSS
2:
3: SCSS RELEASE 04.2-UW 2.0
4:
5: EXPLAIN? (Enter YES or NO)
6: >no
7: STYLE OF PROMPTING?
8: >/help
9:
10: NORMAL*, VERBOSE or TERSE
11: NORMAL: Full prompt (the default).
12: VERBOSE: Prompt plus possible responses; * marks defaults.
13: TERSE: Brief prompts allowing responses on the same line.
14:
15: STYLE OF PROMPTING?
16: >verbose
17: OLD WORKFILE NAME? (NONE* or filename)
18: <<CR>
19: NEW WORKFILE NAME? (Filename)
20: >runit*scsw-1.
21: EXISTING FILE ASSIGNED
22: MASTERFILE NAME? (NONE or filename)
23: >runit*scsm-1.
24:
25: ERROR # 906 AT RUNIT*SCSSM-1.
26: File is empty
27:
28: MASTERFILE NAME? (NONE or filename)
29:
30: >/help
31:
32: NONE leads to file definition phase. Otherwise, enter the
33: name of the existing masterfile. File name formats are
34: installation dependent; see local documentation for details.
35: New workfile named becomes a definition workfile.
36:
37: MASTERFILE NAME? (NONE or filename)
38: >none
39:
40:
41: FILE DEFINITION..
42:
43: MEDIUM FOR INPUT DATA? (FILE* or TERMINAL)
44: >terminal
45: N OF CASES? (UNKNOWN* or integer maximum)
46: >10
47: MAXIMUM VARIABLES = 200
48: VARIABLE NAME? (New varname, old varname or null response)
49: >id
50: LABEL FOR VARIABLE? (Label of up to 40 characters)
51: >Identifikasjonsnummer
52: TYPE OF VARIABLE? (DISCRETE, CONTINUOUS or ALPHABETIC)
53: >discrete
54: VALUE? (Raw value list (MISSING) = value label)
55: >1-100
56: VALUE? (Raw value list (MISSING) = value label)
57: <<CR>
58: VARIABLE NAME? (New varname, old varname or null response)
59: >yrke
60: LABEL FOR VARIABLE? (Label of up to 40 characters)
61: <<CR>
62: TYPE OF VARIABLE? (DISCRETE, CONTINUOUS or ALPHABETIC)
63: >discrete
64: VALUE? (Raw value list (MISSING) = value label)
65: >1 ufaglært; 2 fagarbeider; 3 funksjonær
66: VALUE? (Raw value list (MISSING) = value label)
67: >4 akademiker
68: VALUE? (Raw value list (MISSING) = value label)
69: <<CR>
70: VARIABLE NAME? (New varname, old varname or null response)
71: >loenn
72: LABEL FOR VARIABLE? (Label of up to 40 characters)
73: >Brutto lønnsinntekt pr år
74: TYPE OF VARIABLE? (DISCRETE, CONTINUOUS or ALPHABETIC)
75: >cont
76: VALUE? (Raw value list (MISSING) = value label)
77: <<CR>
78: VARIABLE NAME? (New varname, old varname or null response)
79: >bil
80: LABEL FOR VARIABLE? (Label of up to 40 characters)
81: >Anskaffelsespris for privatbil
82: TYPE OF VARIABLE? (DISCRETE, CONTINUOUS or ALPHABETIC)
83: >disc
84: VALUE? (Raw value list (MISSING) = value label)
85: >0 (missing) Har ikke bil
86: VALUE? (Raw value list (MISSING) = value label)
87: >1 0-10000
88:
89: ERROR # 175 AT 0
90: Invalid punctuation.
91:
92: VALUE? (Raw value list (MISSING) = value label)
93: >1 kr 0-10000
94: VALUE? (Raw value list (MISSING) = value label)
95: >2 kr 10000-50000
96: VALUE? (Raw value list (MISSING) = value label)
97: >3 kr 50000-80000

```

```

98: VALUE? (Raw value list (MISSING) = value label)
99: >4 kr 80000-110000
100: VALUE? (Raw value list (MISSING) = value label)
101: >5 over 110000
102: VALUE? (Raw value list (MISSING) = value label)
103: ><CR>
104: VARIABLE NAME? (New varname, old varname or null response)
105: ><CR>
106: COMPLETE FILE DEFINITION? (YES or NO)
107: >yes
108:
109:
110: TERMINAL DATA ENTRY..
111:
112: ENTER BY? (CASE#, VARIABLE, or LIST)
113:
114: COLLECT? (Name or number of a row item)
115: >/list names
116:
117: POSITIONAL LIST OF VARIABLES
118:
119: 1 ID 2 YRKE 3 LOENN 4 BIL
120:
121: COLLECT? (Name or number of a row item)
122: >1 1 89000 2
123:
124: ERROR 2 385 AT 1
125: The response must be a single item.
126:
127: COLLECT? (Name or number of a row item)
128: >/help
129:
130: Enter a variable name if you are entering data by variable.
131: Enter a case number if you are entering data by case. You
132: are prompted starting at the first empty item in the
133: variable or case. A null response causes collection to
134: begin at the first empty item in the entire data matrix.
135:
136: COLLECT? (Name or number of a row item)
137: ><CR>
138: 1, ID:
139: >1, 1, 89000, 2
140: 2, ID:
141: >2, 4
142: LOENN:
143: >145000, 3
144: 3, ID:
145: >3, 3, 103000, 3
146: 4, ID:
147: >4, 1, 93000, 4
148: 5, ID:
149: >5, 2, 98000, 3
150: 6, ID:
151: >6, 4, 128000, 4
152: 7, ID:
153: >7, 4, 178000, 5
154: 8, ID:
155: >8, 3, 111000, 0
156: 9, ID:
157: >9, 1, 45000, 1
158: 10, ID:
159: >10, 2, 118000, 2
160: PERFORMING BACKUP
161: COLLECT? (Name or number of a row item)
162: >/display all
163: ID YRKE LOENN BIL
164:
165: 1 1 1 89000 2
166: 2 2 4 145000 3
167: 3 3 3 103000 3
168: 4 4 1 93000 4
169: 5 5 2 98000 3
170: 6 6 4 128000 4
171: 7 7 4 178000 5
172: 8 8 3 111000 0
173: 9 9 1 45000 1
174: 10 10 2 118000 2
175:
176: COLLECT? (Name or number of a row item)
177: ><CR>
178: COMPLETE DATA INPUT? (YES or NO)
179: >yes
180: MASTERFILE NAME? (File name for masterfile)
181: >runit*scssm-1.
182: EXISTING FILE ASSIGNED
183: FILE RUNIT*SCSSM-1 GENERATED.
184:
185:
186: ANALYSIS SESSION..

187:
188: NEW WORKFILE NAME? (File name for analysis workfile)
189: >runit*scssw-1.
190: +ENTER 'YES' TO OVERWRITE EXISTING FILE
191: >yes
192: NEW FILE ALLOCATED
193: PROCEDURE? (Procedure name followed by varlist)
194: >/help
195:
196: Enter procedure name optionally followed by variable list.
197: Current procedures are: UNIVARIATE, CROSSTABS, BREAKDOWN,
198: TTEST, SCATTERPLOT, CORRELATIONS, PARTIALS, REGRESSION,
199: FACTOR, and CASELIST. RESIDUALS is entered via REGRESSION.
200: If you name PARTIALS, REGRESSION, or FACTOR, the first
201: series of prompts is issued by the CORRELATIONS procedure.
202:
203: PROCEDURE? (Procedure name followed by varlist)
204: >uni
205: VARIABLE LIST? (Varlist)
206: >yrke
207: CELLS? (ALL* or VALUE,N,TOTPCT,NMPCT,CUMPCT,LABELS or NONE)
208: >value,n
209: STATISTICS? (MEAN*, STDDEV* or list of statistics)
210: >/help
211:
212: MEAN*, STDDEV*, SEMEAN, MODE, MEDIAN, VARIANCE, SKEWNESS,
213: SESKEW, KURTOSIS, SEKURT, MINIMUM, MAXIMUM, RANGE, SUM, or
214: NONE or ALL
215: SEMEAN, SESKEW, and SEKURT are standard errors of the mean,
216: skewness and kurtosis.
217:
218: STATISTICS? (MEAN*, STDDEV* or list of statistics)
219: >none
220: DISPLAY? (STATS, CELLS, BOTH and/or item list)
221: >cells
222:
223: YRKE
224:
225: VALUE N
226:
227: 1 3
228: 2 2
229: 3 2
230: 4 3
231:
232:
233: TOTAL N = 10 VALID N = 10
234:
235: DISPLAY? (STATS, CELLS, BOTH and/or item list)
236: >/proc uni
237: VARIABLE LIST? (Varlist)
238: >bil yrke
239: CELLS? (ALL* or VALUE,N,TOTPCT,NMPCT,CUMPCT,LABELS or NONE)
240: >n
241: STATISTICS? (MEAN*, STDDEV* or list of statistics)
242: >/cells labels,n
243: STATISTICS? (MEAN*, STDDEV* or list of statistics)
244: >none
245: DISPLAY? (STATS, CELLS, BOTH and/or item list)
246: >cells
247:
248: BIL ANSKAFFELSESPRIS FOR PRIVATBIL
249:
250: LABEL N
251:
252: KR 0-10000 1
253: KR 10000-50000 2
254: KR 50000-80000 3
255: KR 80000-110000 2
256: OVER 110000 1
257: HAR IKKE BIL 1M
258:
259:
260: TOTAL N = 10 VALID N = 9
261: YRKE
262:
263: LABEL N
264:
265: UFAGLæRT 3
266: FAGARBEIDER 2
267: FUNKSJONÆR 2
268: AKADEMIKER 3
269:
270:
271: TOTAL N = 10 VALID N = 10
272:
273: DISPLAY? (STATS, CELLS, BOTH and/or item list)
274: >/stop
275: END OF SCSS

```

ENKEL SORTERING I ASCII-FORTRAN

I ASCII FORTRAN finnes det en egen rutine som kan kalles for sortering. Denne kalles ved:

```
CALL FSORT (.....)
```

For å vise hvordan en sortering kan utføres gis to enkle eksempler:

Eksempel 1

Programmet leser data fra enhet 5, sorterer dem i stigende orden etter kolonne 1-4, 8-40. Først sorteres det på institusjon. Innen hver institusjon sorteres det så alfabetisk på institutt/avdeling. Resultatet skrives så ut på linjeskriver evt. terminal.

```
@ftn,si data
  integer core(10000)
  call fsort('key=(1/4,8/40),rsz=80,
    -core=10000!',5,6,core)
end
@map,si t,l
  in data
end
@xqt l
nth  elkraftteknikk           94210
nlht  fysisk institutt        96110
nth   industriell kjemi       94160
nth   mekanikk                93560
nlht  nordisk institutt       96410
nlht  avdeling for realfag     96010
nth   bergavdelingen          94800
nth   kjemiteknikk            94130
nlht  eksamenskontor          96696
nth   fysikalsk kjemi         94178
@eof
```

RESULTAT:

```
nlht  avdeling for realfag     96010
nlht  eksamenskontor          96696
nlht  fysisk institutt        96110
nlht  nordisk institutt       96410
nth   bergavdelingen          94800
nth   elkraftteknikk          94210
nth   fysikalsk kjemi         94178
nth   industriell kjemi       94160
nth   kjemiteknikk            94130
nth   mekanikk                93560
```



Eksempel 2

Dataene som skal sorteres ligger på filen INDATA, og resultatet ligger på filen UTDATA. Postlengden er 80 tegn og sorteringsnøkkelen finnes i posisjon 1-11.

```
@asg,ax indata.
@asg,ax utdata.
@use 9.,indata.
@use 10.,utdata.
@ftn,si data
  call fsort('rsz=80,key=(1,11) !',
    -9,10)
end
@map,si a,b
  in data
end
@xqt b
@eof
```

På filen INDATA ligger byene usortert.

```
trondheim
bergen
stavanger
oslo
arendal
moss
skien
fredrikstad
tromsø
røros
```

På filen UTDATA ligger byene sortert alfabetisk.

```
arendal
bergen
fredrikstad
moss
oslo
røros
skien
stavanger
trondheim
tromsø
```

Ny SPSS-versjon



En ny SPSS-versjon, 8.1-UW2.0, er tilgjengelig på UNIVAC 1100/62 og kan kalles ved:

```
@RUNIT*STAT.SPSS
```

Av viktige nyheter/endringer kan nevnes:

- 1) Den nye versjonen kan lese SPSS-filer som er laget med tidligere versjoner. Filer laget med den nye versjonen kan imidlertid ikke leses av tidligere versjoner.
- 2) Dersom SPSS kalles opp med X-opsjon (@RUNIT*STAT.SPSS,X) vil batchkjøringer bli avbrutt umiddelbart etter utgang fra SPSS når alvorlige feil blir oppdaget under SPSS-kjøringen. D.v.s. at eventuelle jobbtrinn som kommer etter en SPSS-kjøring med feil ikke vil bli utført.
- 3) Som alternativ til @USE-setningen kan en nå benytte ordet "UNIVACNAME" i de følgende filbehandlings-setningene:

```
. GET FILE
. GET ARCHIVE
. GET SCSS
. INPUT MEDIUM
. MERGE FILES
. RAW OUTPUT UNIT
. SAVE ARCHIVE
. SAVE FILE
. SAVE SCSS
```

Følgende oppsett viser hvordan dette virker:

```
@RUN
@RUNIT*STAT.SPSS
GET FILE UNIVACNAME='QUAL*FIL1' SPSSFIL

SAVE FILE UNIVACNAME='QUAL*FIL2' SPSSFIL
FINISH
```

SPSSFIL henspeiler her på den navneangivelse en tidligere hadde på GET FILE- og SAVE FILE-setningen.

For bruk av setningene GET SCSS og SAVE SCSS henvises det til egen artikkel om SCSS i dette nummer av RUN-NYTT.

PRINT - et nyttig program

Med programmet PRINT kan en ved bare å referere til filnavnet få generert og eventuelt utført samme sort styrekort for alle elementer på en programfil på UNIVAC.

En kan f.eks. bruke PRINT til:

- 1) Å kompilere alle symbolske elementer på en fil.
- 2) Å liste ut alle symbolske elementer på en fil.
- 3) Å gi en liste over alle symbolske elementer på en fil med antall programlinjer i hvert element.
- 4) Å lage styrekort av vilkårlig type for alle elementer. Styrekort av gitt type skrives ut på en fil, og disse kan så endres med editor.

PRINT kan håndtere elementene i den rekkefølgen de ligger på filen, eller i alfabetisk rekkefølge.

Kall av programmet:

```
@RUNIT*PROG.PRINT,opsjon filnavn.
```

Eksempler:

- a) Skriver ut i alfabetisk rekkefølge en innholdsfortegnelse for filen PFIL, og deretter innholdet i alle symbolske elementer:

```
@RUNIT*PROG.PRINT,TFQ PFIL.
```

- b) Kompilerer alle symbolske elementer på filen PFIL med FTN-kompilatoren, og resultatet legges på samme fil. Programteksten skrives ut:

```
@RUNIT*PROG.PRINT,CJ PFIL.,PFIL.
FTN
```

Programteksten blir ikke skrevet ut dersom opsjonen N benyttes sammen med C og J. Sideskift for hvert element unngås ved å bruke opsjon Z på RUN-kortet (@RUN,/TPZ..).

Brukerveiledning:

RUNIT-minihåndbok: Elementhåndtering med PRINT, 1982-02-15. Den selges i RUNITs ekspedisjon, SBII, 2.etg.



EFFEKTIV PROGRAMUTTESTING MED ASCII FORTRAN

I ASCII FORTRAN finnes det flere gode hjelpemiddel for effektiv programuttesting. De to viktigste av disse blir kort omtalt i denne artikkelen:

- * interaktive feilfinningsmuligheter i "CHECKOUT COMPILER"
- * "DEBUG"-pakker

1. CHECKOUT COMPILER

De interaktive uttestingsmulighetene kan benyttes når kompilatoren kalles på følgende måte:

```
@FTN,CZ fil.element
@EOF
. kommandoer for uttesting
. og inndata til programmet
.
```

Symbolisk tekst for hovedprogram og alle subprogrammer må ligge på samme element. Når C-opsjon benyttes, kan ikke programmet kalle rutiner i relokerbare subrutinebibliotek (f.eks. NAG). Når kompilatoren kalles på denne måten utføres programmet trinn for trinn. Mellom hvert trinn kan brukeren bl.a.:

- liste ut variabelverdier
- endre variabelverdier
- spesifisere neste stopp i program - utførelsen
- lagre bort hele "programtilstanden"
- gå tilbake til et tidligere trinn i programutførelsen
- eksplisitt kalle subrutiner for separat uttesting av disse

Brukeren styrer programutførelsen ved hjelp av spesielle kommandoer.

Disse kommandoene må benyttes når kompilatoren er i såkalt "DEBUG MODE".

Dette inntreer bl.a.:

- før første linje i programmet blir utført
- på punkter i programmet der brukeren har spesifisert avbrudd
- når alvorlige feil er oppdaget under programutførelsen (IGDM, IOPR o.l)
- dersom setningen CALL PAUSE utføres i FTN-programmet
- etc.

Viktige kommandoer er:

- | | |
|-----------------|---|
| GO | - programutførelsen fortsetter til neste avbrudd er spesifisert |
| STEP n | - neste avbrudd skal skje etter n utførte programlinjer |
| BREAK n | - avbrudd ønskes når linje med etikett n er utført |
| SETBP var | - avbrudd ønskes hver gang angitt variabel blir referert eller endret |
| CALL SUB (....) | - eksplisitt kall på en subrutine |
| DUMP var1,var2 | - utlisting av variabelverdier |
| SET varx = c | - sett variabelverdier |
| SAVE n | - gjem bort aktuell programtilstand, gi den nummer n |
| RESTORE n | - gå tilbake til programtilstand n |
| TRACE ON | - ønsker fortløpende utlisting av linjenr. som blir utført |
| TRACE OFF | - slår av sporing |
| HELP cmd | - gir informasjon om de enkelte kommandoer |

Nedenfor er det vist et lite eksempel der @FTN,CZ er benyttet. Følgende program er lagt inn på MOMS.MOMS:

```

1      IMPLICIT REAL (A-Z)
2 *
3 *LES INN BRUTTO INNKJØSPRIS. BEREGN BETALT MOMS
4 *
5 10    READ(5,5,ERR=99) BTOINN
6      IMOMS=BTOINN-BTOINN/1.2
7 *
8 *KALLER SALGP FOR A BEREGNE BRUTTO SALGSPRIS.
9 *
10     CALL SALGP(BTOINN-IMOMS,BTOUT)
11 *
12 *BEREGN MOMS INKLUDERT I UTSALGSPRIS OG MOMS
13 *SOM SKAL BETALES TIL STATEN.
14 *
15     UMOMS=BTOUT-BTOUT/1.2
16     STATEN=UMOMS-IMOMS
17     WRITE(6,6) BTOINN,BTOUT,STATEN
18 *
19 5     FORMAT(
20 6     FORMAT(' INNKJØSPRIS: ',F6.2,/,
21         &' UTSALGSPRIS: ',F6.2,/,
22         &' MOMS TIL STATEN: ',F6.2)
23     GO TO 10
24 99    CONTINUE
25     END
26 *
27     SUBROUTINE SALGP(NTOINN,UTSP)
28     IMPLICIT REAL (A-Z)
29     NTOUT=NTOINN*1.35
30     UTSP=NTOUT*1.2
31     RETURN
32     END

```

Nedenfor vises så en kort uttesting av programmet med @FTN,CZ:

```

>@ftn,cz moms.moms
FTN 10R1U2 04/01/82-11:28(0,)
>@eof

END FTN 67 IBANK 56 DBANK
ENTER DEBUG MODE AT LINE 5
C:>step 3
C:>go
>120.0
STEP BREAK AT LINE 10
C:>dump btoinn,immoms
BTOINN /*
          .12000000+003
VARIABLE NOT DEFINED
IN MAIN PROGRAM
C:>dump imoms
IMOMS /*
          .20000001+002
C:>step 2
C:>go
STEP BREAK AT LINE 30
C:>dump ntout/salgp
NTOUT /SALGP
          .13500000+003
C:>break 10
C:>go
INNKJØSPRIS: 120.00
UTSALGSPRIS: 162.00
MOMS TIL STATEN: 7.00
>3333.33
BREAK AT LINE 10
C:>save 1
C:>call salgp(5555.0,btout)
ENTER DEBUG MODE (RETURN FROM CALL COMMAND)
C:>dump btout
BTOUT /*
          .89990999+004
C:>restore 1

ENTERING USER PROGRAM: MOMS          VERSION: 1
C:>dump btoinn
BTOINN /*

```

```

C:>go
INNKJØSPRIS: *****
UTSALGSPRIS: *****
MOMS TIL STATEN: 194.44
>333.333
INNKJØSPRIS: 333.33
UTSALGSPRIS: 450.00
MOMS TIL STATEN: 19.44
>@fin

```

Under utførelsen spesifiseres først "STEP" 3, dvs. at avbrudd ønskes ved 3 aktive programlinjer. Deretter settes utførelsen i gang ved "GO". Første verdi, 120.0, leses inn og programutførelsen stopper på linje 10. Her ser vi på verdiene til BTOINN og IMOMS før vi ber om stopp etter at de to neste linjene er utført. Dette avbruddet blir inne i subrutinen SALGP. For utlistering av lokale variable i subprogrammer må en angi "/subprogram" etter de variabelnavn det refereres til. Deretter spesifiseres det at neste avbrudd skal skje når en neste gang kommer til READ-setningen.

Med SAVE 1 blir programtilstanden lagret på en fil. Deretter testes subrutinen SALGP separat med "CALL"-kommandoen. Når dette er gjort, hentes programtilstanden slik den var når "SAVE 1"-kommandoen ble benyttet tilbake og programutførelsen fortsetter.

Det er naturligvis en sterk begrensning i anvendelsen av @FTN,CZ at hele programmet må ligge som et element på en programfil, og at ingen biblioteksrutine kan kalles. @FTN,CZ er derfor mest beregnet for uttesting av små selvstendige programmer. I tillegg bør det ligge vel til rette for uttesting av subrutiner, og da spesielt velegnet for CTS-brukere. Subrutinen kan hentes inn i arbeidsområdet og FORTRAN-setningene "CALL..." og "END" legges inn foran første programlinje i subrutinen. Deretter benyttes CTS-kommandoene ASSUME COMP FTN,CZ og RUN. Brukeren kan nå gjentagne ganger kalle subrutinen med ulike inngangsparametre og se på resultatet etter hvert kall. Om det ønskes kan subrutinen naturligvis utføres trinnvis slik at en også kan se på mellomresultater.

2. "DEBUG"-pakker

Etter siste aktive programlinje i hvert ASCII FORTRAN hoved- eller subprogram kan en legge inn spesielle setninger for feilfinning. Og dette kan gjøres i alle programenheter, uansett om programmet kaller rutiner i programbibliotek eller ikke. DEBUG-pakken bør fjernes etter endt uttesting for å spare CPU-tid og penger. Eneste begrensning er at slike DEBUG-pakker ikke kan benyttes sammen med opsjonene V eller Z ved kompilering. DEBUG-pakkene starter alltid med en DEBUG-setning.

EFFEKTIV forts.

I alle programenheter som har en DEBUG-setning kan man legge inn ingen, en eller flere feilfinningspakker. Start på en feilfinningspakke er definert ved setningen:

AT s

hvor s er en etikett i programmet. En feilfinningspakke avsluttes ved en ny AT-setning eller ved programmets slutt.

En komplett feilfinningspakke består av en AT-setning som kan være etterfulgt av en eller flere av de følgende setningstyper:

TRACE ON - Slår på sporing (tracing) av setningsnummer.

TRACE OFF - Slår av sporing.

DISPLAY varlist - gir utskrift av et sett variable.

Nedenfor er det til selvstudium et lite eksempel på ASCII FORTRANs feilfinningsmuligheter.

FORTRAN KILDETEKST:

```

1.      DIMENSION A(10)
2. 100  I1=8
3. 200  I2=12
4. 300  DO 400 I=I1,I2
5.      A(I)=I
6. 400  CONTINUE
7.      CALL SUB(A,8)
8. 500  CONTINUE
9.      I1=5
10.     CALL SUB(A,I1)
11. C
12.     DEBUG TRACE,INIT(A,I1,I2),SUBCHK(A)
13.     AT 200
14.     TRACE ON
15.     DISPLAY I1,I2,A(10)
16.     AT 500
17.     TRACE OFF
18.     END

1.      SUBROUTINE SUB(A,I1)
2.      DIMENSION A(10)
3. 100  I2=I1+4
4. 200  DO 300 I=I1,I2
5.      A(I)=I
6. 300  CONTINUE
7. C
8.     DEBUG INIT(A,I2),SUBTRACE
9.     END

```

RESULTAT:

```

DEBUG UNIT          -1
AT LINE            2 : I1 = 8
TRACE ON
|
I1 = 8,I2 = 12, A(10) = .00000000
IEND
TRACE 200
AT LINE 3 : I2 = 12
TRACE 300
AT LINE 5 : ELEMENT 8 OF A = 8.0000000
TRACE 400
AT LINE 5 : ELEMENT 9 OF A = 9.0000000
TRACE 400
AT LINE 5 : ELEMENT 10 OF A = 10.0000000
TRACE 400
AT LINE 5 : SUBSCRIPT IS OUT OF RANGE FOR ARRAY A
AT LINE 5 : ELEMENT 11 OF A = 11.0000000
TRACE 400
AT LINE 5 : SUBSCRIPT IS OUT OF RANGE FOR ARRAY A
AT LINE 5 : ELEMENT 12 OF A = 12.0000000
TRACE 400
ENTER SUBPROGRAM SUB
AT LINE 3 : I2 = 12
AT LINE 5 : ELEMENT 8 OF A = 8.0000000
AT LINE 5 : ELEMENT 9 OF A = 9.0000000
AT LINE 5 : ELEMENT 10 OF A = 10.0000000
AT LINE 5 : ELEMENT 11 OF A = 11.0000000
AT LINE 5 : ELEMENT 12 OF A = 12.0000000
RETURN FROM SUBPROGRAM SUB
TRACE OFF
AT LINE 9 : I1 = 5
ENTER SUBPROGRAM SUB
AT LINE 3 : I2 = 6
AT LINE 5 : ELEMENT 5 OF A = 5.0000000
AT LINE 5 : ELEMENT 6 OF A = 6.0000000
AT LINE 5 : ELEMENT 7 OF A = 7.0000000
AT LINE 5 : ELEMENT 8 OF A = 8.0000000
AT LINE 5 : ELEMENT 9 OF A = 9.0000000
RETURN FROM SUBPROGRAM SUB

```

"FTN-TIPS"

SPAR
CPU-TID
-og
penger



For et større FORTRAN-program som skal brukes flere ganger, kan en spare CPU-tid ved å kompilere programmet med FTN-optimaliseringsopsjonene V eller Z. Opsjonen V foretar lokal optimalisering, og Z en mer omfattende - global - optimalisering. Se kap. 1.08 i FORTRAN (ASCII) Programmers Reference, UP-8244.1. En kan redusere CPU-tiden til under halvparten. Følgende eksempel viser tiden for et program på 164K.

1) Kompilering (tid i sek.)

Opsjoner	CPU-tid	Total tid
@FTN,OS	5.96	17.12
@FTN,OVS	7.61	19.50
@FTN,OZS	12.20	37.70

2) Bare eksekvering med data - ikke kompilering og kollektering (tid i min.-sek.)

Opsjoner	CPU-tid	Total-tid	@COST (Typek.F)
@FTN,O	6.51.4	7.05.8	360.50
@FTN,OV	3.16.1	3.28.5	171.60
@FTN,OZ	2.48.5	3.03.0	147.50

En bør kompilere med en slik opsjon når programuttestingen er ferdig og programmet skal benyttes i produksjonskjøringer. Kompileringstiden er lenger, som vist i tabellen, og en kan ikke med disse opsjonene bruke alle de feilfinningshjelpemidler som finnes i ASCII FORTRAN.

STUDENTKJØRINGER MED SPESEIELLE BEHOV

For studentkjøringer (typekode A) på UNIVAC 1100/62 gjelder i dag følgende begrensninger:

Kl.: 0800 - 1600:

Max programstørrelse: 30K interaktivt, 60K satsvis.

Max CPU-tid: 10 sek.interaktivt,
1 min. i satsvise
kjøringer.

Max utskrift: Max 50 sider både
i satsvise og inter-
aktive kjøringer.

Kl.:1601 - 0759:

Max programstørrelse: 60K interaktivt, "u-
begrenset" satsvis.

Max CPU-tid: 10 min. interaktivt,
ubegrenset satsvis.

Max utskrift: 500 sider både for
interaktive og sats-
vise kjøringer.

RUNIT har vist fleksibilitet når det gjelder praktisering av disse grensene der det har vært spesielle behov. Dette vil vi også fortsette med, men under mer formaliserte former. For øvingsopplegg, prosjektoppgaver eller lignende der det er behov for å overskride en eller flere av de ovenstående grenser må ansvarlig faglærer heretter sende en skriftlig søknad til RUNIT som dokumenterer de spesielle behov. Det vil da bli tatt ut egne kjørenummer i hvert enkelt tilfelle.

Typekode for "spesielle" studentkjøringer er B.

Kjøring på typekode B tillates bare etter spesiell søknad, som behandles av Rolf Bysting (tlf. 2940) eller Bjørn Gifstad (tlf. 2966).

Henvendelser må derfor rettes til dem, og alle relevante opplysninger må om nødvendig fremlegges skriftlig. Tillatelsene vil normalt være tidsbegrenset og vil være gjenstand for oppfølging.

POLYDOC!

Tenk deg at du sitter på et kontor og må vite hvem som leverer en spesiell type vare, uten å måtte lete gjennom bransjeregister og varekatalog. POLYDOC!

Eller du vil gjerne vite om biblioteket har en bok skrevet om et bestemt emne eller av en bestemt forfatter, men biblioteket er langt unna. POLYDOC!

En telefonkatalog som er lett å slå opp i? POLYDOC!

POLYDOC er et generelt oppslags-system som ikke trenger noe programmering. Det er bare å skrive inn de data du vil ha gjemt i databasen, og så kan de brukes til å lage lister, index-lister, oversikter, statistikker eller telefonkataloger. Dette er POLYDOC i et nøtteskall.

POLYDOC er et programsystem som er laget av NSI (Norsk Senter for Informatikk i Oslo), og er innlagt og i drift på UNIVAC 1100/62 her i Trondheim. Systemet leser inn og organiserer dataene slik at de er lett å finne igjen med noen enkle kommandoer. Har du lyst til å prøve systemet, så kalles det opp ved

```
@ADD POLY*DOC.POLYDOC
```

Passordet er NPASS og alle kommandoer må begynne med ". En enkel forklaring kan fåes ved å skrive ?

Alle spørsmål angående systemet kan rettes til Bjørn Steine, Gruppe for brukerkontakt- og programvare, RUNIT, tlf. (9)2033.

COBPROGEN COBOL program generator

COBPROGEN er et interaktivt programsystem utviklet av RUNITs ADB-gruppe. Systemet genererer COBOL-program som benytter skjerm-bilder definert i UNIVAC's DPS-system. DPS krever tilgang til UTS-terminal.

Med COBPROGEN kan en lage følgende program:

- Registreringsprogram
- Spørreprogram
- Vedlikeholdsprogram

Fremgangsmåten er som følger:

- 1) Skjermbildet defineres ved hjelp av DPS-systemet
- 2) Working storage av skjermbildet genereres av DPS
- 3) Working storage prepareres med PDP-prosessoren
- 4) COBPROGEN

COBPROGEN kan også benyttes uten tilknytning til DPS-systemet for å generere alle divisjonene fram til "PROCEDURE DIVISION" samt "DECLARATIVES" og paragrafer for "OPEN" og "CLOSE" av filene i programmet.

I COBPROGEN er det også innlagt mulighet for å definere COBOL-rapporter ("REPORT SECTION" o.s.v.) interaktivt eller automatisk ved angivelse av elementer fra "Data dictionary".

Beskrivelse er under utarbeidelse. Interesserte kan henvende seg til Skjalg Ahlstrøm tlf. 2956.

DEMOS - "Discrete Event Modelling on SIMULA"

Relokerbare og omnibus versjonene av DEMOS er nå tilgjengelige fra SIMULA-biblioteket. DEMOS brukes ved:

```
@USE DE.,SIM*LIB.
@ASG,A DE.
@SIM,IS ,B
EXTERNAL CLASS DEMOS ("DE.MOS").
DEMOS
BEGIN
.
.
.
END;
@MAP,IS ,Z
IN B
LIB DE
ENT B
@XQT Z
<data>
```

Litteratur:

Birtwistle, G.M.: Discrete Event Modelling on SIMULA, Macmillan (1979).

Eric Ole Barber

SINCLAIR'S ZX-81 - liten, men stor!

Bare for å ha en ting klart, det er ikke mange datamaskiner som selges idag for under 1000.- kr. Det er, såvidt jeg vet, bare Sinclair som gjør det. Og jeg må si jeg var spent da jeg fikk tilsendt pakke fra England i sommer.

Det tok meg bortimot 2 timer å lodde sammen hele stasen (4 IC'er og 20-30 motstander/kondensatorer), og under over alle undere, den virket på første forsøk. Nå må jeg tilstå at jeg hadde kjøpt litt ekstra utstyr samtidig, strømforsyning (ca. 100.-) og ekstra hukommelse (ca. 600.-), så jeg hadde ikke de store plass-problemene det fører med seg å ha bare 1k minne.

Jeg hadde også investert i et par kassetter med programvare, bl.a. et telefonkatalog-program, et for bank-kontoer og et for å notere forskjellige ting og en kassett med enkle spill. Så det første jeg gjorde var å koble til kassettspilleren, og leste et program over til maskinen. Alt som var nødvendig å vite sto i manualen som fulgte med, og telefonkatalogen var på tv-skjermen min etter 2 minutter.

Programmerings-språket BASIC kunne jeg litt av fra før, så diverse klokker og grafiske mønstre ble presentert for interesserte familiemedlemmer i løpet av de nærmeste dagene. Etter en stund ble det litt begrenset hva BASIC kunne få til, så maskinkode ble forsøkt. Etter noen forsøk ble maskinen og jeg omsider enige om hvor det passet å lagre maskinkode-programmene.

For å utvide maskinparken litt kjøpte jeg en printer til maskinen i høst (ca. 600.-), og som et av eksempel-programmene der fulgte det et grafikkprogram som adresserte 160*240 punkter på papiret, og ga ganske pene demonstrasjonskurver som resultat. Nå slapp jeg også å sitte med papir og blyant for å skrive ned programmene. Printerens kan liste programmene automatisk, eller den kan kopiere det som står på skjermen ut på papiret. Papiret som brukes er selvfølgelig, som alt annet fra Sinclair, bare å få kjøpt fra Sinclair, og koster ca. 30.- pr. rull, men det letter arbeidet betraktelig.

Nå er det jo begrenset hvor mye glede man kan ha av en maskin som bare kan kommunisere ved hjelp av en tv-skjerm, så jeg begynte å lete i tidsskrifter etter andre muligheter, og de kom. Man kan få kjøpt generelle byggesett som gir muligheter for å styre 8 releer og bli styrt av 8 brytere.



Det finnes noen som kan gi maskinen beskjed om riktig tid på dagen, det finnes noen som kan lese spenning eller temperatur, og det siste jeg har sett, er et sett som skal gi maskinen muligheter til å kommunisere med andre datamaskiner på samme måte som en terminal (RS232).

Sinclair selv arbeider på en mikro-floppy som skal erstatte kassettspilleren, samtidig som han har utviklet en flat tv-skjerm som ganske sikkert vil passe bra sammen med zx-en. Det er også blitt utviklet hukommelsestillegg på både 32k, 48k og 64k, så kanskje det snart er på tide å legge inn CP/M eller UNIX på maskinen. Da må i tilfelle monitorprogrammet forandres litt, men det er ganske sikkert ikke noe problem.

Det hele ender opp med at for den nette maskinkostnad av 600.- kr. har man en datamaskin som man kan prøve seg på, for nye 600.- kr kan man bruke den til noe fornuftig, for nye 600.- kr kan den skrive beskjeder på papir, for mellom 100.- og 500.- kr kan man få den til å styre omverdenen (stereo, tv, elektrisk fyring, lys etc.), og for den ivrige, nye 1000.- gir deg muligheter som en fullt utbygget mikro-maskin (som ellers ville kostet fra 30.000.- og oppover). Sum under 5.000.-kr. (dette er basert på priser i England pluss toll, frakt og moms).

Da det allerede finnes en del studenter her i Trondheim som har kjøpt zx-81, har vi funnet det hensiktsmessig med en bruker-gruppe, slik at oppkjøp av programvare og liknende kan koordineres. Interesserte kan henvende seg til:

Sinclair brukergroupe
v/ Bjørn Steine
Singsaker Studenthjem
7000 Trondheim
tlf. 20092 etter kl. 1700

eller tlf. 92033 kl. 0800-1600

EDITOR-TRIKS

Studenter og andre som har hatt edb-opplæring på NORD-maskiner, blir ofte svært skuffet når de skal begynne å bruke UNIVAC med linje-basert editor. Det kan derfor være nyttig å kjenne til noen av triksene som kan gjøre UNIVAC's editor vel så enkel som NORD's PED.

De aller enkleste kommandoene, I (=input), blank linje, C (=change) og L (=locate) tar jeg ikke med, da disse forutsettes kjent. Som en avhjelper til L kan vi ta med F (=find), som søker fra begynnelsen av hver linje. Dette kan være til stor hjelp ved tekstbehandling, for å søke seg frem til nærmeste DOC-kommando i teksten som alltid begynner i første kolonne. Som en ekstra støtte kan man, dersom man vil gjenta siste C,L eller F kommando bare skrive kommando-bokstaven (C,L eller F), så blir hele kommandoen repetert. Ønsker man å vite hva den siste kommandoen var, kan man henge på et spørsmålstegn, så blir den skrevet ut uten å bli utført. Dersom man vil gjøre omfattende rettinger eller lignende, men bare innenfor visse kolonne- grenser, fins det en kommando som heter LIMIT, med følgende skrivemåte:

LIMIT kommando fra-kolonnenr. til-kolonnenr.

Denne kan brukes for C,L og P (=print). Dersom man ikke oppgir til-kolonnennummer, blir den antatt å være 132.

Hvis man vil vite hvor mange linjer det er i et element, er kommandoen LAST, som skriver ut linjenummeret på siste linje.



"Only once in every generation is there a computer that can write poetry like this."

Om man vil legge inn ny tekst bakerst i et element, er det nok å skrive A (=append) som medfører at man blir posisjonert etter siste linje og satt i INPUT-mode. Man kan kopiere tekst internt i et element ved å bruke DITTO-kommandoen, og man kan flytte tekst med MOVE-kommandoen.

Dersom man har et element fra før av, og man ønsker å bruke hele eller deler av teksten i et nytt element, kan man skrive kommandoen ADD, som har følgende skrivemåter:

ADD fil.eltnavn

ADD fil.eltnavn fra-linjenr. til-linjenr.

Den siste utgaven vil bare addere linjene mellom de to linjenummerene fra det andre elementet. De nye linjene blir i begge tilfelle lagt etter slutten av elementet. Hvis man ønsker å legge til nye linjer umiddelbart etter der man står, kan man henge et '+'-tegn etter ADD-kommandoen:

ADD+ fil.elementnavn

Ønsker man å gå andre veien, overføre linjer fra elementet man er på til et annet element, er kommandoen:

SPLIT

Kommandoen kan brukes på samme måte som ADD.

Når man skriver programmer i FORTRAN eller COBOL er det viktig at man holder rede på hvilken kolonne teksten begynner på i hver linje. Til hjelp for dette har vi SCALE-kommandoen, som skriver ut en linje med fortløpende kolonne-nummerering.

Ingenting er så oversiktlig som å få det på trykk, og det kan man få til med SITE-kommandoen. Den mest brukte utgaven skrives:

LNS! SB2

og betyr at hele elementet skal skrives ut med linjenummer på linjeskriveren i Sentralbygg II. Man får her to spørsmål:

HDG?

som besvares med en tekst som vil bli skrevet ut øverst på hver side av utskriften

MSG?

hvor svaret blir skrevet ut nederst på siste side sammen med teksten

PLEASE RETURN THIS LISTING TO:

Kommandoen STATUS vil skrive ut hvordan alle kontroll-kommandoer er satt i denne kjøringen, og betydningen er som følger:

TAB:	angir hvilket tegn som gir tabulator	TAB <tegn>
CSCHAR:	tegn som betyr at neste linje er en fortsettelse av denne linjen	CCH <tegn>
LCHAR:	tegn som omgir LOCATE-tekst	LCH <tegn>
MSCHAR:	tegn som brukes istedenfor @ ved INPUT	MSC <tegn>
TCHAR:	tegn som betyr hvilket som helst tegn ved F-kommandoen	TCH <tegn>
CHAR:	tegn som betyr hvilket som helst ved C- og L- kommandoene	TCC <tegn>
CHCAR:	tegn som betyr SHIFT for terminaler med bare store bokstaver	SCH <tegn>
EXCHAR:	tegn som er en spesiell oktall verdi	EXC <tegn>
MCCHAR:	tegn som skiller flere editor-kommandoer på samme linje	MCC <tegn>
TABS:	skriver ut de tabulator-settingene som gjelder	SET n,n,n,..
MAXLINE:	maximal linjelengde	MAX n
CASE:	om det er store og små bokstaver (NORMAL) eller bare store (UPPER)	CASE U CASE N
MODE:	hvilket tegnsett som er i bruk (ASCII eller FIELDDATA)	opsjon P/Q
AUTO:	tar sikkerhetskopii etter n forandringer	AUTO n n1
LIMITS:	hvilke kolonne-grenser som er satt	LIMITS c n1 n2
LOOP SPACE:	hvor mye plass det er igjen til LOOP	

Deretter følger en hel del spesial-tilstander, og disse er:

BRIEF:	all utskrift undertrykkes. Meget bra ved kommandoer som skal gjennomløpe mange linjer, f.eks. change / //all, hastigheten blir mangedoblet.	MEMORY:	husker alle disse modes innen samme run.
DSPLIT:	linjer som blir overført med SPLIT skal slettes	NUMBER:	skriv alltid ut linjenummeret på hver linje.
EOF:	gjør det mulig å skrive inn blanke linjer. @EOF endrer fra input-mode til edit-mode, mens INP går andre veien.	PCNTRL:	skriver ut printer-kontroll-tegn
INSPEQ:	skriver ut det opprinnelige linjenummeret sammen med det nye. Editor forandrer alle linjenummer når en linje slettes eller blir lagt til, men den holder orden på det gamle nummeret også	QUICK:	tar bort alle blanke i utskriften
LSTINP:	alt som skrives inn til editor (nye linjer og kommandoer, skrives ut igjen. Til god hjelp ved macroe og loop, og ved satsvis kjøring.	TRDINP:	ved mode OFF og @@CQUE kan man skrive inn tekst uten å vente på noe svar fra EDITOR
		UNISCP:	brukes ved INLINE og @@CQUE for å få riktig plassering av rette-karakterer.
		XBRIEF:	ikke skriv ut linjer overført med SPLIT og ADD
			Alle disse modes kan endres ved å skrive ON eller OFF foran mode-navnet.

EDITOR forts.

Enkelte ting krever litt nærmere forklaring. Dersom man jobber med innlegging av mye tekst, vil det lønne seg å gi @@CQUE som kommando til operativ-systemet, og deretter OFF TRDINP i editoren. Dette medfører at operativ-systemet ikke må skrive ut > før den kan lese inn nye data, og at editor ikke skriver ut linjenummer før hver linje som skal skrives inn. Da kan man bare skrive fortløpende uten å vente på at maskinen er klar til å ta imot.

LOOP er en måte å få gjentatt flere editor-kommandoer mange ganger, dersom man skal gjennomføre en spesiell endring i et stor element. Man skriver LOOP n, hvor n er et tall som sier hvor mange ganger loopen skal gjenomløpes. Maskinen svarer med LP**> og man kan skrive inn de kommandoene man ønsker utført. Når alle er skrevet inn, skriver man @EOF, og så blir de utført.

MACRO er noe lignende, men de blir bare utført; en gang. Til gjengjeld kan de gjemmes til siden, og kalles med navn. På denne måten kan man lage sine egne editor-kommandoer. Et enkelt eksempel er en kommando som skriver ut første side i elementet (d.v.s. de første 22 linjene) med linjenummer foran.

```
MACRO FP
TOP
LNP 22
@EOF
```

Kommandoen TOP betyr gå til starten av elementet, LNP 22 skriver ut 22 linjer med linjenummer. Det hele startes ved å skrive FP. Ønsker man å gjemme macroen til senere bruk, må man ha tilordnet en egen fil som må hete ED\$MAC. Editor vil alltid sjekke om denne er tilstede når man lager eller bruker en macro, men editor prøver aldri å tilordne den. Det finnes en del spesielle kommandoer som bare kan brukes i LOOP og MACRO. Disse brukes for å overføre parameter fra macro-kall, spesielle verdier fra operativsystemet samt for å kontrollere gjennomløpingen av en LOOP eller MACRO. Det finnes en som heter LPSUB som kan erstatte et tegn i en kommando med spesielle tekster eller verdier, og LPTST som kan hoppe over kommandoer etter som om en test slår til eller ikke. I tillegg kommer:

```
LJUMP
hopper til en spesifisert kommando i et
MACRO-element
```

```
COMP
beregner en tall-verdi
```

```
XTI
tillater at brukeren gir en kommando inne i
en MACRO
```

```
LPEND
avslutter en LOOP inne i en MACRO
```

```
LPX kontrollerer avslutningen av LOOP
og MACRO
```

Et problem med LOOP og MACRO er at de vanligvis avsluttes med en gang de kommer til slutten på et element. LPX -1 vil medføre at MACRO ikke avsluttes før den er helt ferdig. LPX 0 betyr at den avsluttes med en gang, og LPX 1 at alle looper avsluttes etter at den som holder på er ferdig.

Alle kommandoene i EDITOR har visse spesial-kommandoer som kan henges på i tillegg til det som er beskrevet her. Disse kan man finne ved finlesing av EDITOR MINI-MANUAL som kan kjøpes for kr 25.- i Ekspedisjonen i 2. etg. i SBII.

@@ -TRIKS

Det finnes en del "triks" som kan gjøre UNIVAC litt lettere å ha med å gjøre. Det oppfordres til brukeren å lese mer om disse tingene på egen hånd, her skal de bare nevnes for at folk skal ha en ide om hva som finnes.

```
@@CQUE betyr at all input til maskinen
blir mellomlagret, slik at man
slipper å vente på > for hver
linje.
```

```
@@PTI betyr at maskinen er i stand til
å motta store datamengder på en
litt spesiell måte. Den brukes
når man vil lese data fra en
mikromaskin med floppydisk eller
lignende.
```

```
@@SEND,U Dette er en måte å styre
utskrift til en annen terminal,
f.eks. en DIABLO-skriver. Kom-
mandoen @SYM <utskriftsfil>.,,/U
sender utskriften i kø på
brukernummer, og på den aktu-
elle terminalen skrives bare
brukernummer og passord, der-
etter etter @@SEND,U .
```

NABLA-TRIKS forts.

@@TM Betyr "Terminal Message" og kan brukes til å sende beskjeder fra en bruker til en annen. Format:

@@TM brukernr/U beskjed

@@TYPE Kommandoen brukes for å bestemme hvilken type terminalen er. **@@TYPE=ALPHA** er det vanlige, men en kan også bruke **@@TYPE=4**, som angir at det skal skrives ut 132 tegn på linjen.

@@END Avslutter **@@PTI** og **@@CQUE**

Mange av standard kommandoene i OS-1100 kan også sendes med dobbel @ foran seg. På den måten blir de gjennomslipbare for det som ellers skjer. Det gjelder ASG, FREE, ADD, CAT, SYM, USE, START og MSG. De blir omgående utført uten at man mister kontrollen med det programmet man holder på med.

Alle disse kommandoene, både de med dobbel @ og standard OS-1100 kan skrives ut fra et program med ASM-kommandoen ER APRTCN\$. Dette krever dog at man har litt greie på UNIVAC-Assembler.

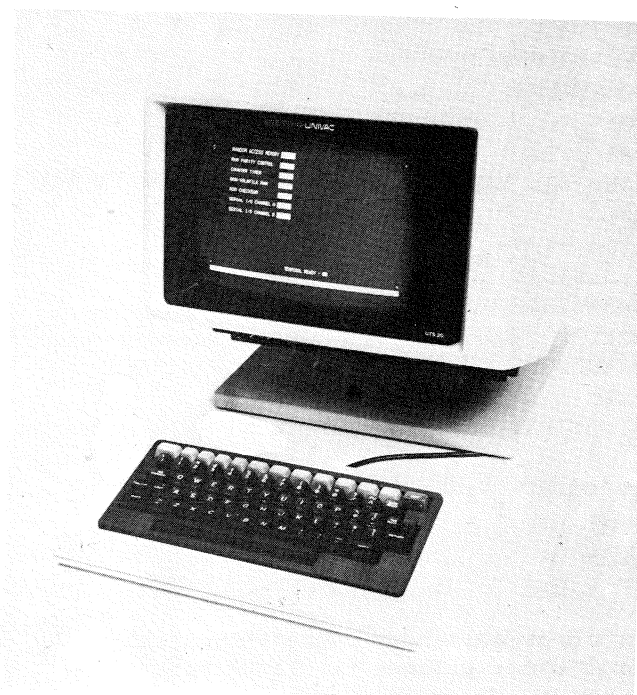
Et problem ved maskinparken på RUNIT er at terminal-konsentratorene, PDP-11-maskinene, av og til får problemer med UNIVAC-forbindelsen. Dette kan medføre en PDP-11-error, og at man mister ekko på terminalen. Ekko kan da fås tilbake ved å skrive **@EKKO** på terminalen. Den svarer med ***UNIVAC 1100***, og ekko er tilbake.

Nytt av året er det at kommandoene XON/XOFF er innført. Det vil si at ved enhver utlissing kan man trykke CNTRL-S for å stoppe utskriften, og CNTRL-Q for å fortsette.

Nye UTS-terminaler på sal 212

De offentlige terminalene tilknyttet UNIVAC som står på tegnesal 212 i SBII, har vært av varierende kvalitet. Feil på terminalene har vært en daglig foreteelse, noen har dårlig skjermbilde, noen har "hikke", og variasjonene i tastatur er store.

Dette blir det nå en bedring på, 10 nye UTS-terminaler blir installert på tegnesal 212. Terminalene er av type UTS-20, som er en UNISCOPE-kompatibel terminal med full-skjermmuligheter. For en mer utførlig beskrivelse av disse terminalene henvises til RUN-NYTT nr.1, side 8.



UTS-20 vil med sin lekre design og avanserte muligheter bidra til et mer publikumsvennlig miljø på tegnesal 212.

PUBLIKASJONER forts.

RELASJONSDATABASESYSTEMET CS4
1981-09-20

(kr. 15)

TEKSTBEHANDLING MED ED/DOC
1981-04-01

(kr. 25)

TEKSTEDITOREN ED
1981-11-20

(kr. 25)

THE NAG LIBRARY OF NUMERICAL
ALGORITHMS MARK-8
1981-09-18

(kr. 25)

HÅNDBØKER/RUNIT-RAPPORTER

GPFS-F USERS GUIDE
høsten 1981 (kjøpes på TAPIR)

(kr. 25)

GRAPHISTO, USERS MANUAL
1979-10-22

(kr. 150)

HÅNDBOK I SYSTEMERING
1980-11-11

(kr. 250)

PROGRAMKATALOG 1981
1981-10-05

(kr. 60)

RA 2 BRUKERVEILEDNING -
1979-09-01

(kr. 160)

SIMULA
1981-09-01

(kr. 80)

SURRENDER
1979-10-03

(kr. 100)

Overføring av filer mellom NORD-maskiner med kommunikasjon gjennom Nord-nett

Det følgende er en oppskrift på hvordan man går fram for å overføre filer mellom NORD-maskiner som har kommunikasjon gjennom Nord-nett.

Eksempel:

Filoverføring mellom NORD-10/B (lokal maskin) og NORD-100/B (remote maskin). Først må en opprette remote forbindelse med den maskinen en vil overføre filer til/fra:

(ESC)

16.16.09 20 MARS 1982
*** ND-10/B - CHILL ***

ENTER brukernr
PASSWORD:
OK
@REMOTE 1
CHANNEL NUMBERS: LOCAL - 602, REMOTE-602

(ESC)

14.19.59 20 MARS 1982
SINTRAN III - VS VERSION G
**** ND-100/B ****
ENTER SYSTEM
PASSWORD:
OK
PROJECT PASSWORD:

PROJECT NAME: DUMMY

R@

(DEL)

Når forbindelsen er opprettet, trykkes (DEL) for å komme tilbake til lokal maskin.

På lokal maskin skriver en kommandoen COMMUNICATION-STATUS. En får da følgende utskrift:

*** ND-10/B - CHILL ***

@COMM-ST

OUTPUT FILE:

REMOTE LINE NUMBER: 1

CHANNEL	0:	LOCAL DEVICE NUMBER =	600
		REMOTE DEVICE NUMBER =	600
CHANNEL	1:	LOCAL DEVICE NUMBER =	601 BACKGROUND
		REMOTE DEVICE NUMBER =	601
CHANNEL	2:	LOCAL DEVICE NUMBER =	602 BACKGROUND
		REMOTE DEVICE NUMBER =	602 BACKGROUND
CHANNEL	3:	LOCAL DEVICE NUMBER =	603
		REMOTE DEVICE NUMBER =	603 BACKGROUND

0 SEQUENCE ERRORS, 6 NAKS RECEIVED, 2 NAKS SENT
12 FRAMES RETRANSMITTED
COMMUNICATION RUNNING

Hvis en maskin har flere enn en remote forbindelse, må en oppgi linjenummer etter COMMUNICATION-STATUS (1 ELLER 2).

Ved hjelp av overstående utskrift kan en finne ut hvilke kanaler som er filoverføringskanaler:

De kanalene der "remote device number" ikke har bakgrunnsprogram er filoverføringskanaler.

Filoverføring fra lokal (her: N-10/B) til remote maskin (her: N-100/B), gjort på lokal maskin:

@COPY-FILE CHANNEL-0."LOCTILREM" LOCTILREM
destination source

Filoverføring fra remote (her: N-100/B) til lokal maskin (her: N-10/B), gjort på lokal maskin:

@COPY-FILE "REMTILLOC" CHANNEL-0.REMTILLOC
destination source

Etter at en er ferdig med filoverføringene må en huske på å logge ut fra remote maskinen; og frigjøre forbindelsen til remote maskinen med @LOCAL kommandoen.

@REMOTE

R@LOG

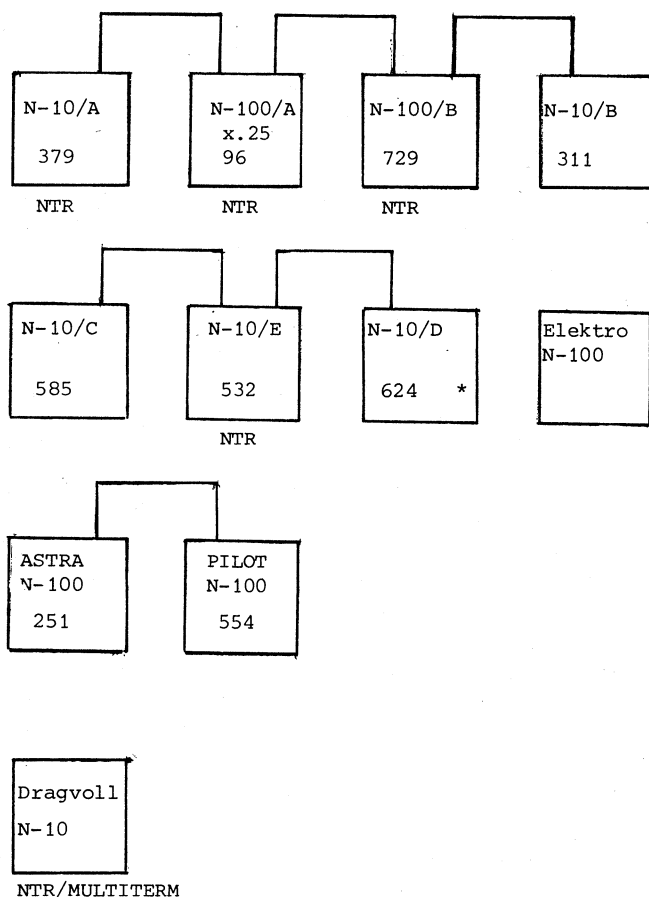
14.21.33 20 MARS 1982
TIME USED IS 0 SECS OUT OF 1 MINS 23 SECS
--EXIT--

(DEL)

*** ND-10/B - CHILL ***

@LOCAL

NORD-NETT VED RUNIT



COST-prosessoren er utvidet

Ved å utføre programmet "@COST" før avslutningen av en kjøring vil man få et godt anslag over hva kjøringen har kostet, også splittet opp på CPU-pris, IO-pris etc.

Tidligere hadde en kun mulighet til å få beregnet kostnadene avgrenset til hver enkelt kjøring. Programmet er nå utvidet slik at en kan få akkumulert kjørepriser og antall kjøring over en lengre periode. De akkumulerte kostnader lagres på en datafil som først må opprettes av brukeren. Man må angi en opsjon og denne filen i prosessorkallet.

I utvidet form har da kall til COST denne formen:

```
@COST,<opt> QUAL*FILE.
```

QUAL*FILE: Den brukerdefinerte datafilen

I-opsjon : Initiering (nullsetting) av akkumuleringsfilen

A-opsjon : Akkumulering av kostnadene

Kall til COST uten opsjoner gir samme effekt som tidligere.

Fremgangsmåten ved bruk av utvidet COST blir da:

Opprett en fil som skal brukes til å akkumulere kostnadene for en unik bruker eller brukergruppe. Forslag: Bruk gjerne kjørenummer (kontonummer) som filnavn!

1) @ASG,UP EKSEMPEL*ZZ9100M.

Nullstill filen med I-opsjon i COST-kallet:

2) @COST,I EKSEMPEL*ZZ9100M.

I de påfølgende kjøring nyttes så A-opsjon for å akkumulere kostnadene og antall kjøring.

3) @COST,A EKSEMPEL*ZZ9100M.

For interaktive kjøring vil utskrift fra 3) ha denne formen:

PRIS BASERT PÅ BRUKERKODE: M

	SISTE RUN	DEMAND	BATCH	TOTALT
ANTALL RUNS:	31	10	41	
CPU-PRIS KR:	256.86	970.02	32.26	1002.28
IO-PRIS KR:	.19	34.33	6.84	41.18
UTSKRIFT KR:	20.94	95.14	28.22	123.36
TILKNYTN. KR:	48.63	271.83	.00	271.83
TOTALPRIS KR:	326.62	1371.32	67.51	1438.83

Den kostnadsbevisste bruker bør selvfølgelig kalle COST med A-opsjon ved alle kjøring; hvis ikke vil de akkumulerte verdier vise for lite.

Et problem eksisterer i denne situasjonen:

Dersom en batch-kjøring går i feil på et tidlig stadium og COST-kallet befinner seg helt på slutten av kjøringen, vil @COST normalt ikke bli utført. RUN-NYTT nr. 3 1980, s. 17 viser et forslag til løsning av dette problemet.

Knut Berget

Nye sentralenheter på RUNIT's PDP-11 konsentrator

Natt til 25 mars i år var det et generasjonsskifte ved RUNIT. En av våre gamle PDP-11/20 maskiner gikk over i pensjonistenes rekke, etter å ha tjenestegjort som terminalkonsentrator ved RUNIT i mer enn 10 år. En respektabel alder for en datamaskin!

De siste par årene hadde den desverre, i likhet med sine to søsken, blitt stadig verre plaget av senilitet. Det var hovedsakelig primærlageret som skapte problemer. Dette begynte å få "hull" som var ute av stand til å holde stabilt på informasjon. Hva det kan føre til burde vel ikke være vanskelig å forestille seg.... Det ble vanskelig å få systemene til å kjøre noenlunde stabilt, og uforklarlige feil oppsto.

På grunn av en stadig større høyere feilfrekvens ble det besluttet å skifte ut PDP-11/20 maskinene med noe mer tidsmessig utstyr. Valget falt på en av Digital Equipment Corporation's (DEC) nyskapninger, PDP-11/24, den minste prosessoren i DEC's 4. generasjons PDP-11 serie.

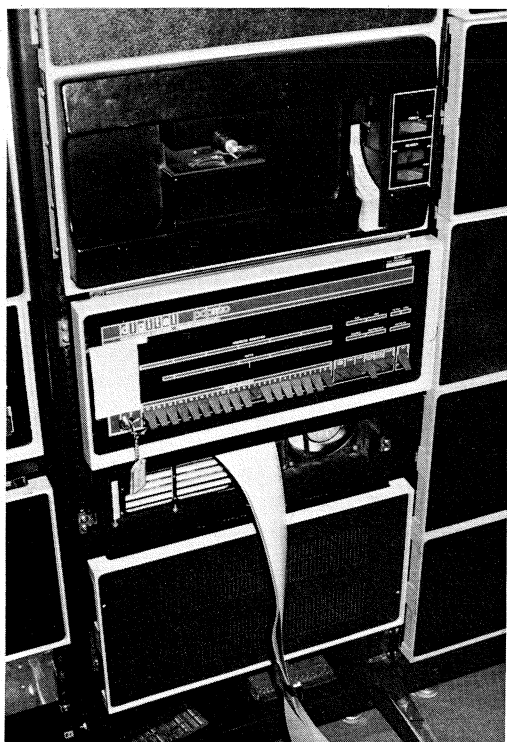
Prosessoren har svært beskjedne ytre mål. (Bilde 2.) Kortet som ligger øverst i boksen inneholder hele sentralenheten samt to grensesnitt mot henholdsvis konsollterminal og kassetstasjon. På et kort som ligger under, (som dessverre ikke synes) er det gitt plass til 256K byte MOS primærlager. Det siste er forresten utbyggbart til en 1 mbyte. Prosessorkortet er dessuten forsynt med ledige posisjoner til flytende aritmetikk- og kommersielt instruksjonssett. Til sammenligning fylte den gamle PDP-11/20, med bare 56K byte primærlager, og uten grensesnitt mot konsoll og papirbåndstasjon, de tre nederste hyllene på bilde 1. (Disse hyllene var dessuten stappfulle!)

Processor + kassetstasjon fyller nå opp en hylle (Bilde 3).

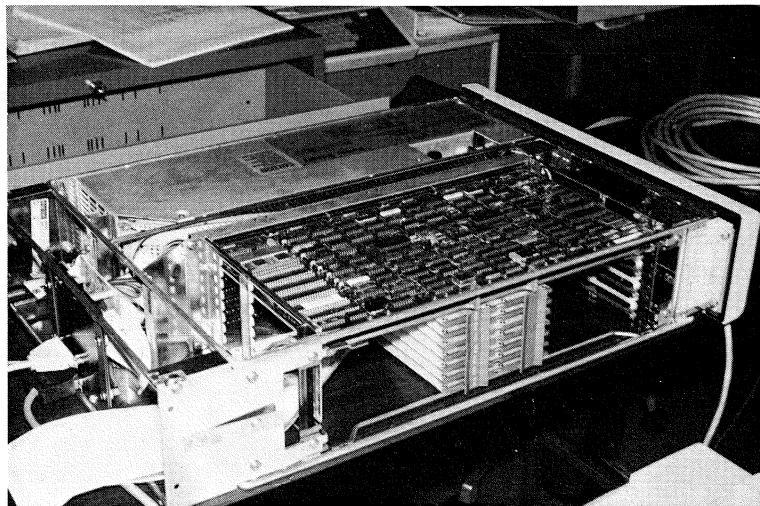
PDP-11/24 ble lansert sommeren 1981, og like før jul fikk RUNIT sine 5 eksemplarer som en av de første i Norge.

De fleste vil antakelig si at dette høres vel og bra ut, men hva vil vi merke som brukere?

La meg med det samme tilstå at programvaren er stort sett identisk med den på PDP-11/20. Det er heller ikke der vi har forsøkt å oppnå gevinst i første omgang. Det vi derimot er ute etter, er å få konsentratorene til å bli så driftsstabile at brukerne ikke merker at de faktisk har et ledd mellom seg og UNIVAC-anlegget.



1) Gamle ærverdige PDP-11/20-A.



2) "Inside" PDP-11/24.

Vi regner med en del problemer i startfasen, men forsøker etter beste evne å begrense dem, slik at man ikke kommer verre ut i den første tiden etter utskiftingene enn før.

Resultater fram til i dag gir oss grunn til optimisme, - driftsloggen taler ganske klart i så måte. Det skal riktignok ikke skjules at vi, har hatt, og har en del mindre ustabiliteter, men det legges ned et betydelig arbeid på å løse dette.

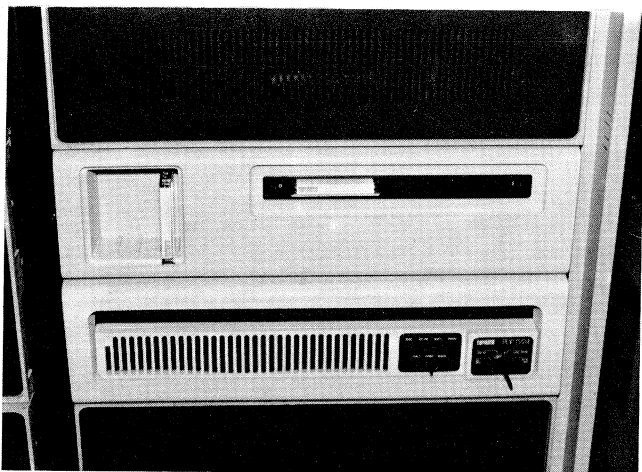
Vi bruker de samme terminalkontrollerne nå som PDP-11/20 brukte, og vi var veldig spente på hvordan kryssingen av gammel og ny teknologi ville bli. Dette har gått over all forventning, og vi har ikke hatt problemer som kan spores direkte tilbake til dette.

Ellers kjører vi i samme oppsett som før, med 3 konsentratorer - A, B og C.

Det foreligger planer om å sette en fjerde konsentrator i drift, men det er ikke fastsatt når dette skal skje.

Programutvikling til PDP-11 konsentratorene har foregått forskjellige stadier. I begynnelsen (1971 - 72) foregikk utvikling på PDP 11/20 direkte. Sikre kilder fra dengang kan berette om harde tider. Utskriving av en kildetekst på ca. 1000 linjer tok godt og vel 1 time! Alt foregikk selvfølgelig ved hjelp av papirbånd.

En lettelse var det sikkert da det ble lagt kryssassembler og - lenker til bruk på UNIVAC. Men fremdeles tok programutvikling lang tid. Halve dagen gikk gjerne med til systemgenerering. Utprøving måtte skje på tider av døgnet hvor brukere flest hadde lite behov for å bruke terminal mot UNIVAC, man måtte nemlig bruke en produksjonsmaskin til utprøving.



3) PDP-11/24 nederst med kassettbåndstasjonen TU-58 øverst.

I dag er all programutvikling lagt over til RUNITs VAX 11/750 anlegg og vi har reservert en PDP-11/24 til test/utprøvningsformål. Fra å skrive en kodebit til den er klar til utprøving tar det nå ca. 5 minutter.

Til slutt vil jeg nevne at det planlegges å skifte ut programvare på PDP-11 konsentratorene. RUNIT har anskaffet et RSX-11/S monitorprogram som er tenkt å danne basis for ny programvare. Når det skal skje er det for tidlig å uttale seg om, men det kan kanskje være en ide å følge med i fremtidige utgaver av RUN-NYTT?

Per Arne Enstad

PUBLIKASJONER FRA RUNIT

Følgende håndbøker er utgitt av RUNIT og kan fås ved henvendelse til RUNITs ekspedisjon i 2.etg. SBII.

BROSJYRER

SLIK BLIR DU BRUKER HOS RUNIT (gratis)

SLIK BRUKER DU INTERAKTIV TERMINAL (kr.5)

MINIHÅNDBØKER/INFO

APL (A Programming Language) 1981-02-04 (kr.25)

BMDP (Biomedical Computer Programs) 1981-04-01 (kr.25)

BRUK AV STANSEMASKIN (kr.15)

DOKUMENTPROSESSOREN DOC 1981-11-20 (kr.15)

ELEMENTHÅNDBØKER MED PRINT 1982-02-15 (kr.10)

ETABLERING AV POLYDOC-DATABASE VED RUNIT 1982-02-05 (kr.25)

FMPS (Functional Mathematical Programming System) 1980-06-06 (kr.25)

GLIM 1981-04-14 (kr.15)

INNFØRING I CTS 1981-09-20 (kr.25)

MAGNETBÅND TIL OG FRA ANDRE ANLEGG 1981-09-20 (kr.10)

MEDLEMSKARTOTEK OG ADRESSELAPP-PROGRAMMER VED RUNIT 1981-09-20 (kr.15)

MULTIBANKING MED ASCII-FORTRAN PÅ UNIVAC 1100/62 1980-11-09 (kr.15)

P-STAT 1981-04-14 (kr.15)

forts. side 21

Haakon Wiig har datasenter på hybelen



På en 7 kvadratmeter stor hybel ved Kuhaugen i Trondheim har Haakon WIIG sitt private datasenter: En MYCRON3 og en MYCRON 2000, med to sentralenheter og 1 megabyte RAM. Periferutstyret inkluderer 3 Tandbergskjermer, to skrivere, 6 diskettstasjoner og en 10 megabyte Winchesterdisk. Som vi ser av bildet må den ene skriveren plasseres i senga. Vi våger å påstå at

ingen datasentral kan oppvise mer datakraft pr. kvadratmeter enn det vi finner her. Haakons problem er at han har bare en stikkontakt, og må ta ut kjøleskapet for å unngå forstyrrelser i databehandlingen. Derfor drikker han lunken melk mandag morgen etter en 30 timers dataøkt i løpet av weekenden!

Programmerer 20K kode pr. måned.

Fra høsten 1980 har Haakon vært student ved NTH, og går nå i 2. elektro. I tillegg er han prosjektleder for utviklingen av MYTEKST, et av MYCRONS viktigste programvareprosjekter.

Under praksisopphold hos MYCRON hadde han mot programvareleveranse byttet til seg en MYCRO-3. I løpet av første klasse på NTH laget han en editor for MYCRO. Denne viste han MYCRON-folkene og de ble så begeistret at han fikk ansvaret for å viderutvikle den til det som nå markedsføres som MYTEKST.

- Men har du bakgrunn og erfaring nok til å lede et slikt prosjekt?

- Jeg har lært mye hjemmefra, "faderen og moderen" drev trykkeri en gang, og jeg har bl.a. vært aktiv i skoleavisa og speiderbevegelsen. Editoren som la grunnlaget laget jeg for meg selv etter egen ide uten firmatilknytning, og det tror jeg var en fordel. I MYTEKST har vi forsøkt å kombinere det beste fra flere lignede systemer. Ellers liker jeg godt å snakke med aktive brukere av systemet, og gjennom dem få impulser til forbedring.

- Hva arbeider du med i øyeblikket?

- Nå holder jeg på med Telex og kommunikasjon over høynivåbuss. I øyeblikket kan 4 terminaler tilkobles bussen, men i løpet av kvelden regner jeg med å kunne kjøre 16. Etterpå står Teletext og fotosetting for tur.

Haakon Wiig

Vi hadde nesten dårlig samvittighet for å legge beslag på så mye av Haakons tid. Dette var en lørdag, og 30 av de ca. 40 timene Haakon jobber for MYCRON pr. uke gjør han unna i løpet av weekenden. Det er ialt 8-10 personer som deltar i Haakons prosjekt, hvorav 3 er av hans medstudenter. For å kunne ivareta prosjektledelsen reiser han til MYCRON i Oslo to ganger pr. måned, og på flyet opp og ned leser han Tracy Kidder: The soul of a New Machine.

- Eneste boka jeg får tid til å lese i år, sier han. Selv om han er prosjektleder og "deltidsansatt", er det Haakon som lager flest moduler i måneden. Han viser oss en tykk ringperm som er januarproduksjonen (ca. 20k ferdig kode).

Søndagen etter skal han jobbe med et halvprivat oppdrag for et ingeniørfirma, et program for generering av anbud for elektriske opplegg i industribygg.

- Men hvordan får du tid til alt dette, både NTH-studier, heltidsjobb og andre aktiviteter?

- Mitt motto er "gjør det nå, og gjør det skikkelig med en gang". Det man blir sliten og stresset av er å gå tenke på alt man skulle ha gjort - gå rett på og gjør det i stedet! Når jeg rekker over såpass mye er det fordi jeg jobber konsentrert og intenst, og skifter til noe annet når jeg blir trett - det er hvile i å gå til noe nytt. Om formiddagen jobber jeg med teori på skolen, om ettermiddagen er jeg på Omega verksted (elektrostudentenes hobbyhule) og jobber med maskinvare, og om kvelden driver jeg med programmering her på hybelen. Det å sitte her med egen maskin er en fin situasjon, ingen forstyrrende telefoner. Jeg hadde ikke klart å studere uten å ha disse maskinene. Jeg er heldig som har en jobb ved siden av, da kan jeg bedre sortere ut det som er viktig på skolen - man får en ubevisst følelse for hva som er nyttig.

Kombinasjonen av teori og praksis er veldig viktig, det er for lite av det i skolen i dag. Jeg har lært mer av gutta på Omega-verkstedet enn jeg har gjort på NTH.

- Du som driver med både maskinvare og programvare - hva synes du er mest interessant?

- Hverken det ene eller det andre, men

samspillet mellom de to. Man kan f.eks. lage maskinvare som gjør programvaren enkel etc. Gjennom MYCRON-jobben får jeg også god erfaring med brukersiden av databehandlingen, det er meget viktig. Men programmering er det jeg kan jobbe mest intenst med. Fordi jeg liker det, kan jeg jobbe et par døgn i strekk uten å merke det.

- Men damer får du vel ikke tid til?

- Det har vært tynt med det til nå, men må nok satse på det også med tid og stunder.

Egen bedrift i Brummunddal?

- Hvilke planer har du for fremtiden, vil du fortsette hos MYCRON etter studiene?

- Jeg liker meg godt hos MYCRON, men kunne gjerne tenke meg å starte egen bedrift, kanskje som et slags datterselskap av MYCRON.

Det viser seg at Haakon har temmelig klare planer for hvordan bedriften skal realiseres. "Gullhøna" som skal skape hovedinntekten skal være en personlig datamaskin i luksusklassen "som har alt med en gang". Den store feilen med det som tilbys i dag er at grunnversjonen er så ribbet, man må kjøpe tilleggsutstyr for flere ganger grunninvesteringen før det blir brukbart. Haakon jobber på en gård i Brummunddal hver sommer, og det er i Brummunddal bedriften skal ligge. Jordbruket har også inspirert til produktdefinisjon: Hans "personal computer" skal ikke være verken en tohjulsstraktor eller en bulldoser, men en Massey Ferguson 135, seig og evigvarende.

Maskinen skal koste ca. 40.000 kroner og være fullt utbygd med rikelig RAM, 8 tommers diskettstasjon og skriver i grunnversjonen. - Det er der jeg håper å tjene første millionen, sier han, og det er på programvare det går an å tjene penger, ikke maskinvare.

Når han har fått laget produktet som skal gi bedriften den økonomiske ryggrad, skal inntekten brukes til fri forskning på andre ideer. Artige prosjekter, sier Haakon, som for eksempel:

- Døvetelefon. Den Televerket har er for dyr og komplisert. Slike ting går det ikke an å tjene penger på, men et firma bør satse på den type ting likevel som en slags moralsk forpliktelse.

Haakon Wiig

Elektronikken i kirkeorgler er alt for gammeldags. Orgelbyggere er flinke musikere, men de har ikke peiling på elektronikk. Sammen med en kollega på verkstedet har jeg planer om å forbedre elektronikken som ligger i overføringen fra tangent til pipe. Et par av de andre prosjekter jeg allerede har vært litt borti, er jeg også interessert i å videreutvikle: energistyring i boliger, resultatservice for idrettssarrangement og datastyrt lysbildeshow.

Det er enda flere ideer i Haakons hode, og nye dukker stadig opp. Men er det bare data som svirrer i hodet på ham - er han "fagidiot" denne gutten som har gjort EDB til en livsstil? Nei, blant alle diskettene og den lange rekka av BYTE ("BYTE er like viktig som avisen") i hylla hans, ser vi også en perm det står "Ny skole" på. Datastyrt undervisningsopplegg, undrer vi.

- Pensum og lærebøker er gammeldagse. Skoledebatten er fokusert på karakterer eller ei, og det blir de ikke enige om likevel. De burde i stedet diskutere pensum. I skolen i dag lærer de f.eks. brytningsindeksen for glass, men de har ikke peiling på steroanlegg, og det har de bruk for å vite noe om. Det å lære å stille sin egen økonomi, skaffe seg leilighet, ekteskap etc. er mye viktigere enn å vite flateinnholdet av Belgia. Hvis jeg får tid skal jeg skrive ei bok om forslag til nytt pensum i skolen.

- Datafag ville vil inngå i det pensumet?

Haakons raske datakarriere

Databehandling stiftet Haakon bekjentskap med høsten 78 i 3. gym. på Riis Videregående Skole i Oslo, da han valgte BASIC-programmering som valgfag. Etter et par dager slapp han å følge undervisningen og "bodde" på datarommet til han ble kastet ut av vaktmesteren om kvelden. Maskinen var en Tandberg TDV 2114, eller EC10 "skoledatamaskin". Utpå høsten fikk de en diskett med assembler, og da var det gjort, sier Haakon. Med et så maskinært redskap kunne hele maskinens indre dissekeres, og det gjorde Haakon. Disse eksperimentene resulterte bl.a. i BEDIT, en editor for å analysere/reparere/etc. innholdet av disketter.

På nyåret 1979 skulle utstillingen Energia



Haakon har fått meget inngående kjennskap til operativsystemene CP/M og MP/M av å romstere i MYCRO-maskinens indre, og han har holdt to forelesninger for fulle hus om dette temaet for NTH-miljøet.

- Ja, å avmystifisere datamaskinen er viktig. Programmering må være med, som valgfritt fag. Valgmulighetene bør være mange, ingen er like. Prosjekter må det også satses på i mye større grad. Det er lærerikt og givende å lage ting selv.

Og det er såvisst noe Haakon lever opp til. Vi skal følge hans videre karriere med interesse. Brummundal kan bli Norges "Silicon Valley" hvis Haakon får det som han vil.

79 arrangeres på Sjølyst, og Haakons far hadde ide om å lage et datastyrt opplegg for styring av varmeovner i et hus. Familien har hytte i Brummundal i nærheten av Lars Monrad Krohns småbruk, og derfor var det naturlig å ta kontakt med MYCRON for å spørre om de var interessert i et slikt prosjekt. De hadde ikke tid, men hvis man fikk noen til å programmere, kunne de låne ut en maskin.

Gjett hvem som fikk jobben? Haakon jobbet dag og natt i januar måned, totalt 400 timer, og ble ferdig til utstillingen. Resultatet var et datastyrt dukkehus, og i døgn av 2 minutters varighet ble det demonstrert hvordan temperatur og lys i de forskjellige rom kunne styres. Programmet

Haakon Wiig

sjekket olje og strømpriser, og valgte billigste fyringsmiddel. Utetemperaturen var avgjørende for tidspunktet når varmen ble slått på om morgenen. Flere utstillingsbesøkende var interessert i et slikt opplegg, men MYCRON-maskinen kostet 100000 kroner, og MYCRON var ikke særlig interessert i å forfølge ideen.

- Men det burde være mulig å bygge et slikt anlegg for 10000 kroner idag, sier Haakon, kanskje satser jeg på dette som diplomoppgave. Det er bare så mange andre ting jeg også kunne ha lyst til å gjøre diplom på.

Fem dager etter at ventetida var over begynte Haakon som visergutt hos MYCRON. Selv om han beholdt visergutt-jobben hele det året han jobbet der var det altnuligmann han var. Det var for lite folk i alle avdelinger, og Haakon sa ja til alle oppdrag han ble tilbudt. Foruten å hente post, laget han kabler, var ute på serviceoppdrag hos kunder for å fikse et program eller en skriver, laget et lagerprogram for et trelastfirma, arbeidet med MYADM, et programsystem for trygdeetaten, til og med markedsføring fikk han prøve. Han viste også MYCRON-folkene sitt editorprogram for disketter, BEDIT, som han fikk i oppdrag å oversette til PL/M slik at det kunne brukes på MYCRON-maskinene. BEDIT brukes til å "redde" ødelagte disketter, kopiere prog-

rammer med lås, "debugge" programmer etc. Alt som er på diskett kan avsløres med BEDIT, og utviklingen av programmet har lært Haakon mye om datamaskinens indre organer. Dette er blitt et nyttig produkt for MYCRON, og våren 1980 deltok Haakon i konkurransen "Unge Forskere" med det samme programmet. Det ble førstepris, og blant slektshistorier, geologiske avhandlinger etc., skilte seg ut som det eneste som var kommersielt utnyttet.

Haakon kunne like gjerne ha deltatt med et skiskytterprogram som han laget noen måneder tidligere. Da Brummundal idrettslag arrangerte Nordisk Mesterskap i skiskyting, sørget Haakon for å utvikle et program som tok hånd om resultatservicen. Både han og en MYCRO-3 satt og frøs i minus tyve grader, men det gikk helt knirkefritt, både "speaker" og radio/TV fikk resultatene raskere enn de hadde fått noen gang før.

En MYCRO-3 til 100000 er for dyr for et idrettslag, men Haakon holder på å utvikle en billigere prototype for brummundølene, basert på komponenter innkjøpt fra USA.

Etter et år hos MYCRON begynte Haakon på elektrolinjen ved NTH, men kontakten med MYCRON har han beholdt, og har avansert fra visergutt til prosjektleder for MYTEKST.

DATAOVERFØRING INTERNT I UNIT-MILJØET

Universitetsmiljøet i Trondheim er geografisk spredt. Fram til idag er det vesentlig sprednettet for UNITs hussentral som benyttes for dataoverføring, forutsatt at godkjent overføringsutstyr tilkoples. Sprednettet betales av universitetet, men planlegges og drives av Televerket.

På grunn av sterkt økende behov for datasamband på NTH-plataet, bygges det også ut med separate datakabler mellom bygninger. RUNIT melder behov for nye kabler til Driftsavd. NTH, slik at de kan taes med i Televerkets bestillings- og anleggsprogram. Driftsavd. NTH følger opp prosjektet og Televerket foretar nødvendig detaljplanlegging, utarbeider kostnadsoverslag og eventuelt gjennomfører installasjonen til avtalt tid.

På NTH-plataet er det utført/planlagt utvidelser i kabelnettet for data i perioden fra høsten -81 og ut -82 for ca. 400.000 kr. Denne investeringen bringer nettet opp på et tilfredsstillende nivå.

Egne datafordelere er etablert i Bergavdelingen, Verkstedteknisk, Materialteknisk, Kjemi, og Elektro (til SB II), med hovedfordelere i RUNITs termineringsrom på Lerkendal og i Sentralbygg 2. Videre utbygges nettet med kabler til Fysikk, Elektro (til Lerkendal), Hovedbygget og Varmeteknisk.

Det er videre nødvendig med en særskilt innsats fra hvert enkelt institutt for utbygging av interne sprednettet for data i bygg. Ved NTH koordineres dette av Driftsavd. NTH, men ved behov vil RUNIT virke som konsulent og koordinator i datatekniske spørsmål.

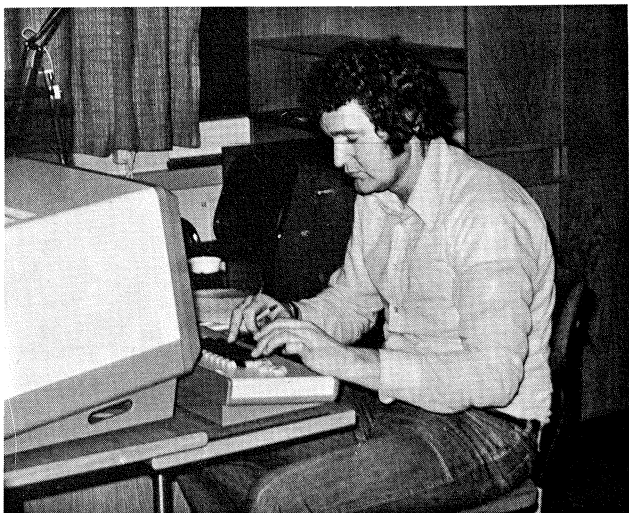
RUNIT står for driften av kabelnettet for data og foretar all kopling og feilretting. Med hensyn til leveringstid på terminaltilknytning i eget nett, forventer vi et akseptabelt servicenivå.

Hans Endresen

RUNIT Gruppeforum

Skal være et organ for vekselvirkning mellom RUNITs grupper og omgivelser. Forumet organiseres som et møte mellom en gruppe og den som påvirkes av eller kan bidra til gruppens virksomhet. Obligatorisk møteprogram er en presentasjon av gruppen: faglig profil, prosjekter og personell. Oversikt over fagområdet gruppa dekker og planer for fremtiden hører også med. Etter innledningen følger en diskusjon, og resultatet av møtet skal være en faglig plan for gruppa på 2 - 5 sider.

Tre av gruppene på RUNIT har avholdt sitt gruppeforum med deltagere både utenfra og fra RUNITs øvrige grupper, og resten av gruppene vil følge i løpet av året.



Gruppeleder Paul Rusten i konsentrert innsats ved terminalen.



Gruppe for brukerkontakt og programvare

Gruppe for brukerkontakt og programvare ble opprettet i 1979. Gruppens hovedformål er:

Å bistå RUNITs brukere med informasjon og veiledning, samt å sørge for et variert programvaretilbud slik at brukerne på en hensiktsmessig måte kan løse sine databehandlingsoppgaver.

Det innebærer kursvirksomhet, veiledningstjeneste, produksjon av informasjonsmateriell etc. i forbindelse med bruken av dataanlegget. Eksempel på programvare som er anskaffet og som det veiledes i bruken av er statistikkpakken SPSS og det matematiske programbiblioteket NAG. Gruppen driver også vurdering av terminaltyper og bistår med veiledning ved terminalanskaffelser. Driften av RUNITs bibliotek tilligger også gruppen.

Av større prosjekter i det siste kan nevnes en grundig gjennomgang av programvaretilbudet og utgivelse av en ny og omarbeidet programkatalog. RUNITs fagbibliotek er "automatisert", det vil si at søkeprogrammet POLYDOC er tatt i bruk for registrering og gjenfinning av litteratur.

Dette året er følgende aktiviteter igang eller planlagt:

- full revidering og nyutgivelse av RINFO
- flere nye "minimanualer"
- interaktivt opplegg for søking etter brukerinformasjon og programvare

Det interne sosiale liv er viktig, og fungerer godt i gruppe for brukerkontakt og programvare. En årlig utflukt i veteranbil hører med.

Gruppeforum: Gruppe for brukerkontakt og programvare

BEHOV FOR VEILEDNINGSTJENESTE, MEN VANSKELIG Å FÅ DEN FINANSIERT

Gruppeleder Paul Rusten innledet møtet med å orientere om gruppens virksomhet (se faktarute). Han avsluttet med å spørre om hvordan gruppens tjenester skal drives i et desentralisert datamiljø, og hvordan aktivitetene skal finansieres.

Dette var et spørsmål som fenget, og mye av diskusjonen etterpå kom til å dreie seg om dette.

To av deltagerne på møtet, Kolbein Bell, Institutt for Statikk, og Kolbjørn Hagen, NLHT, hadde nettopp fått egne VAX-anlegg, men de ønsket fortsatt hjelp av RUNIT m.h.t. å lage håndbøker, skaffe programpakker osv. til disse maskinene. De anså også den rådgivning og assistanse RUNIT kan gi i forbindelse med anskaffelse av lokale anlegg som meget viktig. Det ønskes også bistand av RUNIT til teknisk vedlikehold.

Men problemet er: Den sentrale veiledningstjenesten finansieres stort sett av kjøreinntekter, og etterhvert som stadig flere får egne anlegg reduseres det økonomiske grunnlaget for denne tjenesten. Hvordan dette inntektstapet skal kompenseres hadde ingen patentløsning på, men det mest nærliggende er sentral bevilgning over UNIT-budsjettet.

Når det gjelder programpakker til lokale anlegg, kan ikke disse skaffes sentralt og distribueres til lokale anlegg. Leverandørenes politikk er at programvare anskaffes til hver enkelt maskin, ikke hele miljøer.

Hovedårsaken til det tilbakevendende dilemma mellom sentralt anlegg og lokale ressurser ligger i Universitetets bevilgningsstruktur. Pengene over annum skal dekke et vidt spekter, fra reiser og kontormateriell til datamaskinutgifter. Penger til utstyr er lettere å skaffe, og da blir det attraktivt med eget anlegg og "frie" kjørekostnader. Totaløkonomisk for Universiteter er det en dårlig løsning, men det er vanskelig å gjøre noe med dette.

Museets representant på møtet, Egil Aune, var bekymret for at så mange satset på egne anlegg. Det ville kanskje medføre dyrere tjenester på det sentrale anlegget for de mindre brukerne som ikke har kompetanse eller ikke ønsker å satse på eget anlegg. Odd Meland kunne til dette replisere at RUNITs nominelle priser har gått ned de siste årene, og at det ikke blir snakk om å sette opp prisene, men heller redusere tjenestetilbudet.

KAN VEILEDNINGSTJENESTEN DESENTRALISERES?

Diskusjonen tydet på at det kan bli vanskelig. Det er et vidt felt som skal dekkes, og en person kan vanskelig klare alt. Dessuten vil det bli umulig å skape et faglig miljø blant flere spredte veiledere.

Ellers var det tydelig at brukerne satte pris på at veilednings- og informasjonstjenesten på RUNIT er blitt opprustet de siste årene, og nødvendig så at tjenesten ble nedtrappet. Det var også hyggelig å høre at RUN-NYTT fikk attest for å være et godt organ.

UNIVAC 1100/62 VOKSER

UNIVAC 1100/62 har i løpet av våren vokst seg kraftigere, delvis på bekostning av sin eldre kollega 1100/21.

Like før påske ble primærlageret utvidet med 50% fra 1 Megaord til 1,5 Megaord (6 Megategn). Fra 1100/21 er tre platestasjoner på hver 20 000 TRK flyttet over. Platelagerkanalene på 1100/60 bygges nå ut slik at den kan overta alle de 8 platestasjonene fra 1100/21.

Ny UNIVAC tekstbehandling

For de som ønsker å kombinere databehandling på UNIVAC med tekstbehandling, lanseres nå en interessant nyhet fra UNIVAC. Man må da anskaffe UTS-4000, som består av UTS-40 arbeidsstasjon med sentralenhet, diskettstasjon og dokumentskriver. UTS-40 terminalen kan da brukes som eget tekstbehandlingsanlegg i tillegg til at den også går som terminal mot UNIVAC. Dermed vil det også gå greit å få overført tekst og data fra UNIVAC for lokal tekstbehandling på UTS.

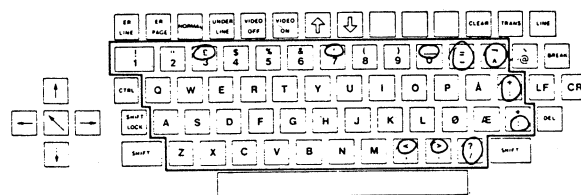
Tastaturet - dataalderens store anakronisme

Datateknikken går fremover med stormskritt, men fremdeles må de elektroniske hjernene motta instruksjoner fra sine menneskelige mestere via et tastatur. Og håndteringen av tastaturet er ikke menneskets sterke side, "pekefingerens" er blitt svært så utbredt etterhvert som "data" berører stadig flere mennesker. Selv om det forskes på mer effektive kommunikasjonsformer, er det lite trolig tastaturet vil få avløsning i dette hundreåret. Denne anakronisme fra forrige århundre vil derfor fortsatt representere den største flaskehals i moderne datateknologi.

QWERTY-tastaturet: konstruert for nedsatt skrivehastighet.

Det er i grunnen utrolig at det såkalte QWERTY-tastaturet (navn etter bokstavene i øverste tasterad) har overlevd så lenge som det har gjort. Da amerikaneren Christopher Latham Soles konstruerte dette tastaturet i 1860-årene, ble plasseringen av tegnene valgt slik at skrivehastigheten skulle reduseres. Årsaken til dette paradoksale designkriterium var at mekanisk sammenstøt mellom skrivearmene skulle unngås, derfor ble de mest brukte bokstavene plassert langt fra hverandre. Det største paradoks er at vi fortsatt lever med dette tastaturet, lenge etter at mekaniske betraktninger er blitt uinteressante. Viktigere måtte det være å effektivisere pekefingerens hoppvals. Men igjen ser en hva standardens makt betyr. Når noe først er blitt en industristandard, skal det veldig mye til for å vinne innpass med noe nytt, selv om det bestående er aldri så mye hull i hodet.

Dataalderen har innført nye taster utenfor det gamle "kjernetastaturet", som f.eks. RETURN, BREAK, CTRL, "krøllalfa" etc. Dette har ikke gjort tastaturforvirringen mindre. Hvordan disse tastene er plassert vil avhenge både av fabrikkasjonsland og produsent. Amerikanerne har en plassering av "datatastene" som er forskjellig fra den europeiske. Det som er iferd med å etablere seg som standard er den såkalte ECMA nasjonal (ECMA: European Computer Manufacturers Association), men problemet er bare at denne avviker på vesentlige punkter fra den allerede etablerte Norsk Standard 4115 for skrivemaskintastatur. Når man vet at



ECMA nasjonal standard. Avvikene mellom denne og Norsk Standard innenfor det området de har felles er avmerket.

skrivemaskiner, tekstbehandlingsutstyr og dataterminaler eksisterer side om side på mange kontor, er dette svært betenkelig.

Nytt tastatur som skal strykes.

En mer fornuftspreget layout av det "indre" tastaturet enn QWERTY har vært forsøkt.

Mest kjent er forslaget til den amerikanske professor August Dvorak, som omplasserte bokstavene slik at fingerbevegelse og tretthet ble minimalisert. Forslaget vakte interesse, men ikke nok til at QWERTY-tastaturet ble jekket ut. Et par andre forsøk har også vært gjort. Et ergonomisk utformet tastatur med en tastgruppe for hver hånd og en del mindre brukte taster mellom disse, har ikke kommet lenger enn til forsøksstadiet. "Writehander", en halvkule med 14 taster som opereres av en hånd med trykking av flere taster i kombinasjon, har også fått en meget begrenset utbredelse.

Det nyeste alternativ til QWERTY-anakronismen kommer fra USA. Professor Edward B. Montgomery fra Dallas har foreslått et tastatur man skal stryke på istedenfor å trykke. Moderne tastatur som ikke består av enkelttaster men en plate med underliggende "fingerfølsomme" kontakter gjør dette mulig (f.eks. Sinclairs tastatur). Ved å studere hyppigheten av ikke bare enkeltbokstaver men først og fremst bokstavkombinasjoner, har han laget et bokstavutlegg som gjør at svært mange ord kan skrives ved at fingeren strykes over en serie taster. Med referanse til

Tastaturet

figuren, ser man f.eks. at ordet THAT kan skrives i to sveip, TH og AT. Ofte forekommende endinger som f.eks. TIOUS kan skrives i et enkelt buet strøk. Fortsatt vil det være nødvendig med en del enkelttrykk, men analyser viser at en oppnår nærmere to strøk pr. enkelttrykk. Vi tror denne ideen må ha noe for seg, det var forbausende befriende å skrive QWERTY da dette ble tastet inn.

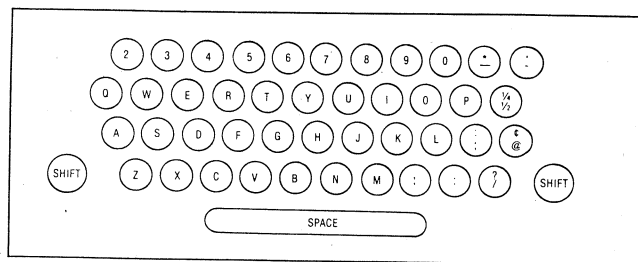
Med datateknikkens hjelp vil det være forholdsvis greit å analysere andre språk og bestemme optimal tegn plassering for disse. Moderne datateknikk gjør det også mulig å omdefinere de enkelte tasters betydning ("intelligente" tastaturer), slik at spesielle tilpasninger lett kan gjøres. Ved å legge en ny sjablong over tastene er det lett å oppnå samsvar mellom funksjon og tastpåskrift.

Talegjenkjenning eller enklere pekefinger-vals?

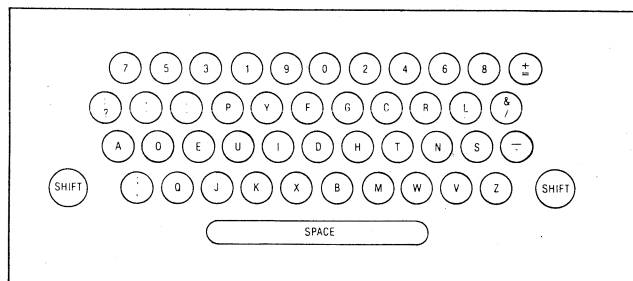
Men vil disse mulighetene gjøre det lettere å kommunisere med datamaskinene via fingertuppene? Forhåpentligvis, men det spørs om de fleste fingre er så fortrolige med QWERTY-melodien at alternativene ikke engang blir forsøkt.

Det neste er da å bruke talen som medium for å bibringe elektronikken våre instruksjoner. Selv om det forskes intenst på dette hos datagigantene både i USA og Japan, er det fortsatt langt igjen. Utstyret er både for dyrt og har for få ord i vokabularet til at det foreløpig er noen konkurrent til tastaturet. 2000 dollar koster det for en boks som forstår 100 ord, og skal ordforrådet utvides til 800 ord blir prisen over 10 000 dollar. I tillegg må boksen lære å kjenne stemmen til den som skal kommandere, og hvis kommandanten er forkjølet en dag, er sjansene for misforståelser store. Tastaturet er derfor fortsatt det enkleste hjelpemiddel for å oppnå "datakontakt", men la oss i alle fall håpe at det inntil videre blir gjennomslag for en enklere pekefingervals.

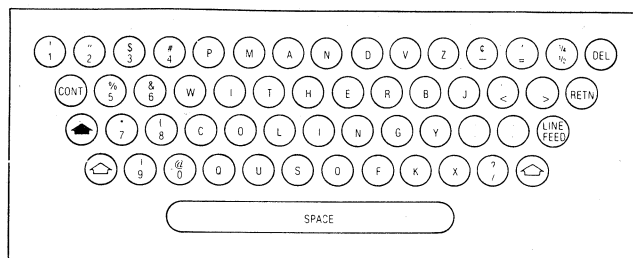
Riktig løsning på kryssord i nr. 1!



QWERTY-tastaturet er håpløst gammeldags.



Dvoraks forbedring av tastatur-layout'en har aldri vunnet innpass.



Vil det gå bedre med Montgomerys tastatur for "strykere"? Hensikten er å la fingeren gli over bokstavsekvenser og dermed redusere antall enkelttrykk.

1	2	3	4	5	6
H	O	L	L	E	R
7	O	C	T	A	L
9	N	R	N	S	H
13	F	R	A	E	B
16	Y	A	R	R	A
19	W	M	T	S	T
21	E	N	I	C	E
22	L	S	I	D	H
23	L	E	S	E	H



INDEX — TIDLIGERE ARTIKLER I RUN-NYTT

Diverse

- | | | | |
|--|--|--|--------------------------------|
| Bort med kort | :Nr.2-81 S. 6 | Lov om personregistre | :Nr.3-79 S. 3 |
| Brukerkontakten styrkes | :Nr.3-79 S. 2 | "Lure"program laget av datastudenter | :Nr.1-80 S.23 |
| Dataetikk | :Nr.8-77 S. 3
:Nr.2-80 S.24
:Nr.3-80 S.14 | Mikroprosessorutstyr i NTH - miljøet | :Nr.3-79 S. 9 |
| Dataforkortelser: | | Minimaskiner i UNIT-miljøet | :Nr.2-80 S.29 |
| DMA | :Nr.7-77 S. 9 | Multibanking med ASCII-FORTRAN på 1100/62 | :Nr.3-80 S.18 |
| GIGO | :Nr.2-79 S. 2 | NORD-planer | :Nr.2-80 S.16 |
| OCR | :Nr.4-78 S.15 | NSD | |
| OEM | :Nr.1-78 S.11 | Norsk Samfunnsvitenskapelig Datatjeneste på Lade | :Nr.8-77 S.22 |
| Datakunst | :Nr.1-78 S.12
:Nr.2-78 S.12
:Nr.3-78 S.12
:Nr.4-78 S. 2 | NSD's Kretsdatabank | |
| Datanett | :Nr.1-78 S. 4 | Inneholder folketellingsopplysninger fra folketellingen i 1960 og 1970 | :Nr.1-78 S.22 |
| Datasikkerhet | :Nr.2-80 S.24
:Nr.2-81 S. 3 | Ny UNIVAC 1100 maskin | :Nr.1-80 S.17
:Nr.2-80 S.11 |
| Datasjakk | :Nr.2-80 S.17 | Nytt operativsystem | :Nr.2-80 S.11 |
| Dataspråk | :Nr.2-79 S. 3
:Nr.2-79 S.27
:Nr.2-80 S.27 | Orakeltjenesten | :Nr.2-80 S.35 |
| Den Norske Dataforening | :Nr.2-80 S.34 | Passord | :Nr.3-78 S. 4 |
| Diskethåndtering | :Nr.2-81 S.13 | Personal Computing | |
| Dividerer du med null i dine FORTRAN-programmer | :Nr.1-80 S. 7 | Nytt databegrep i NTH-miljøet | :Nr.3-78 S. 2
:Nr.3-79 S. 4 |
| DMU- Datamaskinutvalget ved Universitetet i Tr.heim | :Nr.2-80 S.22
:Nr.1-81 S.24 | Presentasjon av RUNITs grupper/ledelse | |
| Elektronikk som kan snakke | :Nr.1-80 S. 2 | Biblioteket | :Nr.3-80 S.26 |
| Er datasentrets dager talte? | :Nr.2-81 S. 3 | Gruppe for informasjonssystemer | :Nr.8-77 S.15 |
| Erfaring med bruk av mikro-datamaskiner | :Nr.2-79 S.12 | Intervju med fagsjef Odd Meland | :Nr.1-82 S. 5
:Nr.2-78 S.20 |
| Håndbøker fra RUNIT | :Nr.1-82 S. 7 | Teknisk gruppe | |
| fra UNIVAC | :Nr.3-78 S. 6
:Nr.1-80 S.22 | Arbeidsoppgaver i Teknisk gruppe | :Nr.1-82 S.22 |
| Systemeringshåndbok | :Nr.1-81 S.22 | Teknisk/matematisk gruppe | :Nr.3-80 S.27 |
| Innlesing av papirhullbånd | :Nr.2-78 S. 7 | Priser og programvareanskaffelser | :Nr.1-81 S.14 |
| Jobber som henger - vranglåsproblemet | :Nr.3-80 S.19 | Privatdatamaskin benyttet som intelligent terminal | :Nr.3-79 S.11 |
| Kubeprogram | :Nr.2-81 S.17 | På besøk i lokale datamiljø: | |
| Kjøreplan | :Nr.1-80 S.13 | Databehandling ved Kjemiavdelingen | :Nr.1-81 S. 4 |
| Komputern- Datastudentenes linjeforening | :Nr.2-80 S.40 | Kolbjørn Bell | :Nr.1-80 S. 4 |
| Kostnadsbesparende tiltak - til ulempe for brukerne? | :Nr.1-82 S.23 | Humanistene på Lade/ Dragvoll | :Nr.3-78 S.16 |
| Linjesvitsj ved RUNIT | :Nr.3-80 S.10 | Nevrologisk avdeling, Regionsykehuset | :Nr.3-80 S.25 |
| | | Reguleringsteknikk | :Nr.8-77 S.16 |
| | | SINTEF avd. for konstruksjonsteknikk | :Nr.2-80 S.10 |
| | | Universitetet på Lade | :Nr.2-79 S.12 |
| | | Vassdrags- og Havnelab. | :Nr.278 S.14 |

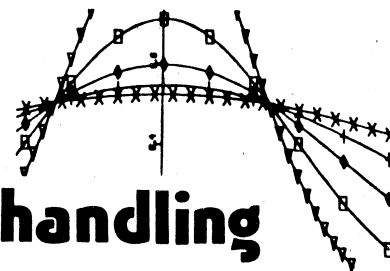
Rammeavtaler	:Nr.2-79 S. 4
RUNIT med tilbud på småmaskinsiden	:Nr.8-77 S.20
Simulering	:Nr.1-82 S.13
Stansing av hullkort på 1100-21	:Nr.4-78 S.10
Statens standard avtaleformularer for EDB-anskaffelser	:Nr.1-80 S.10
Statistikkmisbruk generert av datamaskin	:Nr.2-78 S.10
Studentdatamaskin til eksperimentering	:Nr.2-81 S.15
Teknologiske utviklingstendenser	:Nr.3-78 S. 8
Tekstbehandling	
NOTIS-1 tekstbehandlings-system	:Nr.3-80 S.12
Tekstredning med DOC	:Nr.8-77 S.15
Terminering ved 'MAX TIME'	:Nr.2-78 S. 6
Tidtakermaskinen Jensen	:Nr.8-77 S. 2
UNINETT	:Nr.2-81 S.16
UNIVAC-dominansen i UNIT-miljøet avtar	:Nr.1-82 S.15
UNIX	:Nr.1-82 S.11
Utstansing av hullkort	:Nr.1-80 S.22
Uttesting av programvare for mikroprosessor	:Nr.2-78 S.11
Utstyr og tjenester	:Nr.3-79 S.19
VAX-11/750	:Nr.1-81 S.29 :Nr.2-81 S.14

Filer og filbruk

Bruk færre filer!	:Nr.1-81 S.23
Filen som ble vekk	:Nr.1-78 S. 2
Diverse forslag til hvorfor og hvordan rette opp	:Nr.2-78 S. 2 :Nr.4-78 S. 8
Forsvinner dine filer?	:Nr.2-81 S. 7
Gjenvinning av slettede elementer	:Nr.2-80 S. 7
Kompilering av et stort antall elementer	:Nr.1-81 S.25
Kontroll av katalogiserte filer	:Nr.2-78 S. 8
Ny versjon av FURPUR	:Nr.3-80 S. 6
Pakk filen - spar penger	:Nr.2-79 S. 5
Sikring av filer	:Nr.2-78 S. 5 :Nr.2-80 S.25

Slik skal du tilordne filer på UNIVAC	:Nr.3-80 S. 7
Filoverføring	:Nr.3-80 S. 8
UNIVAC --> NORD	
NORD --> UNIVAC	
NORD <--> NORD	

Grafisk databehandling



Bedre grafisk utstyrs-tilbud	:Nr.7-77 S. 4 :Nr.8-77 S. 9
CALCOMP 936 plotter	:Nr.1-78 S.20
GPGS-F	
Endringer i GPGS-F	
USERS GUIDE	:Nr.1-78 S.20
Ny versjon	:Nr.2-79 S.20
Driver for Tektronix 4662 plotter	:Nr.8-77 S. 9
Oversikt over utstyr og hva dette kan brukes til	:Nr.2-78 S.17
Plotting av kurver, histogrammer m.m.	:Nr.2-78 S.17
SURRENDER	
Presentasjon av 3-dimensjonale data	:Nr.3-79 S.18
TEGRUT	:Nr.2-78 S.18
Tektronix digitaliseringsbord	:Nr.2-80 S.16

Magnetbånd

Bruk av private magnetbånd	:Nr.8-77 S.11
Endret typebetegnelse for 9-spor 800 bpi magnetbånd	:Nr.1-80 s.22
Magnetbånd til og fra andre anlegg	:Nr.2-81 S. 6
Magnetbåndutstyr ved RUNIT	:Nr.3-78 S.11
Nye priser på magnetbånd	:Nr.1-80 S.21
Plass på magnetbånd	:Nr.3-78 S.11
Sikring av filer og magnetbånd på UNIVAC	:Nr.1-80 S.20
Til brukere av magnetbånd på UNIVAC	:Nr.1-80 S.20

Programvare

Digital signalbehandling :Nr.2-80 S.15

DPU-1000

Et hjelpeprogram for overføring av filer, jobber og kommandoer mellom forskjellige vertermaskiner koplet sammen via TELCON. :Nr.1-81 S.11

EISPACK

En samling FORTRAN subrutiner som løser standard egenproblem for diverse typer matriser :Nr.2-79 S. 9

Filnavn og kall for sentral programvare :Nr.1-82 S. 2

FUR

En erstatning for FURPUR :Nr.1-81 S.11

GLIM

Algoritme for tilpassing av generaliserte lineære modeller :Nr.2-80 S.14
:Nr.1-81 S.15

IPF-1100

En integrert programpakke for interaktiv dialog med datamaskinen :Nr.1-81 S.11

NAG, RUNITs numeriske :Nr.2-81 S. 8
programbibliotek :Nr.4-78 S.16

NO308 GAUSSN

En FORTRAN subrutine som beregner et n-dimensjonalt intergral :Nr.2-79 S. 9

OPTIMA 1100

Programsystem for planlegging og oppfølging av nettverksorienterte prosjekter :Nr.1-80 S.18

PADS

Et språkuavhengig verktøy for feilfinning :Nr.1-81 S.10

POLYDOC

Et programsystem for interaktiv søking i store informasjonsmengder :Nr.1-82 S.24

Programkatalog

Korte beskrivelser av de programmer som RUNIT tilbyr sine UNIVAC-brukere :Nr.2-81 S. 8

Programmer for data-adressert post :Nr.1-82 S.12

Programvareanskaffelser :Nr.1-82 S. 8

PSTAT-78

Et interaktivt statistikk-program :Nr.1-81 S.16

RA2 - ny versjon :Nr.1-82 S. 8

RA3

En flerbrukerutgave av databasesystemet RA2 på NORD-maskiner :Nr.2-81 S- 9

RKF 45

Program for numerisk løsning av ordinære differensialligninger :Nr.3-78 S. 7

RUNITs programbibliotek :Nr.1-82 S.10

SCICONIC

Et programsystem for matematisk programmering :Nr.1-82 S.10

SCSS

Et interaktivt statistikk-program :Nr.1-82 S.10

SPSS

Spørsmål, råd og vink :Nr.3-78 S.10
Spørsmål, råd og vink :Nr.4-78 S.18
Nye SPSS-versjoner :Nr.1-80 S.19
SPSS-informasjon :Nr.1-80 S.19
Versjon 8.1-UW1.0 :Nr.3-80 S. 9

STATISTIKKPROGRAMMER :Nr.1-82 S.14

BMDP :Nr.2-80 S.14
COFAMM :Nr.2-80 S.15
EFAP :Nr.2-80 S.15
EXPAK :Nr.2-80 S.15
LISREL IV :Nr.2-80 S.15
MULTIVARIANCE :Nr.2-80 S.16
En brukers erfaring :Nr.2-80 S.16

TEGAS

Et program for simulering og testing av digitale system :Nr.2-81 S. 9

UKILT-1100

Programmer for løsning av nettverksproblemer :Nr.2-79 S. 9

UNIDAS

Et system for gjenfinning av dokumenter :Nr.1-81 S.10

WMTS-1

Programmer for analyse av multiple tidsrekker :Nr.1-82 S.10

Prosessorer

Bruk av @PRT :Nr.2-78 S. 8

COST - beregning av pris :Nr.3-80 S.17

DOC

DOC-prosessor :Nr.3-78 S.10
Tekstredigering med DOC :Nr.8-77 S. 5

EDITOR

Effektiv sletting.
Bruk av D+ :Nr.1-78 S.18
Bedre sikring mot tap av oppdaterte elemente :Nr.7-77 S. 5

NEWS

Prosessor som gir utskrift av aktuelle nyheter :Nr.2-81 S. 9

STATUS

Informasjon om jobber :Nr.7-77 S. 8
under utførelse i maskinen :Nr.2-78 S. 6

Styring av utskrift, utskriftsenheter :Nr.1-78 S.15

SUSPEND og RESUME

Hjelpemiddel for styring av utskrift ved interaktiv kjøring :Nr.8-77 S.10

Språk

APL (A programming Language)	:Nr.8-77 S.12 :Nr.3-78 S. 5 :Nr.2-80 S.12 :Nr.2-81 S.14
ASCII-COBOL Level 5R1 (ny versjon)	:Nr.3-78 S. 9 :Nr.3-80 S.10
ASCII-FORTRAN En brukers erfaring Programbibliotek Overgang fra Fielddata til ASCII	:Nr.2-81 S. 5 :Nr.2-8- S. 4 :Nr.4-78 S. 6 :Nr.1-81 S.25
CTS Effektiviser programutviklingen med CTS CTS gjør databehandling lettere Brukervennlig dialog med CTS Ny versjon Ny CTS-"prescanner" for ASCII-FORTRAN	:Nr.3-79 S.16 :Nr.2-80 S.18 :Nr.3-80 S. 4 :Nr.1-81 S.26 :Nr.2-80 S.21 :Nr.1-82 S.14 :Nr.1-82 S. 2
FORTRAN Blokkstrukturert FORTRAN-oversettere FORTRAN V, ASCII-FORTRAN, Reentrant FORTRAN Flyttbare FORTRAN-programmer	:Nr.1-82 S. 9 :Nr.4-78 S. 4 :Nr.2-79 S. 6 :Nr.1-80 S. 5 :Nr.1-81 S.25
INDIRA Interaktivt Databasespråk for ikke-programmerere mot RA2	:Nr.1-78 S.10
MARY Et maskinorientert programmeringsspråk	:Nr.4-78 S.12
PASCAL	:Nr.1-81 S.13
SIMULA Versjon 2.0MZ av SIMULA-oversetteren Ny versjon 3R5	:Nr.2-78 S. 2 :Nr.3-80 S. 9

Terminaler — bruk og utstyr



Oppringt samband	:Nr.2-79 S.10
Program som er beregnet på avansert terminalbruk	:Nr.8-77 S.14
Terminaloversikt	:Nr.1-82 S.15
Terminaler til undervisning	:Nr.1-82 S.21
Tilknytting av terminaler	:Nr.1-82 S.20
UNIVAC's nye terminaler	:Nr.1-81 S. 8
Skikk og bruk for terminalsittere	:Nr.1-78 S. 7
Startjobber fra terminal	:Nr.2-79 S.10
Studentkjøring via terminal	:Nr.1-78 S.14
Styring av utskrift til interaktiv terminal	:Nr.2-79 S.10
Terminal-kokebok	:Nr.1-78 S. 7
Terminalkonsentrator PDP11A og PDP11B	:Nr.3-80 S.24
Terminaltilknytning	:Nr.3-80 S.23

KRYSSORDVINNERE

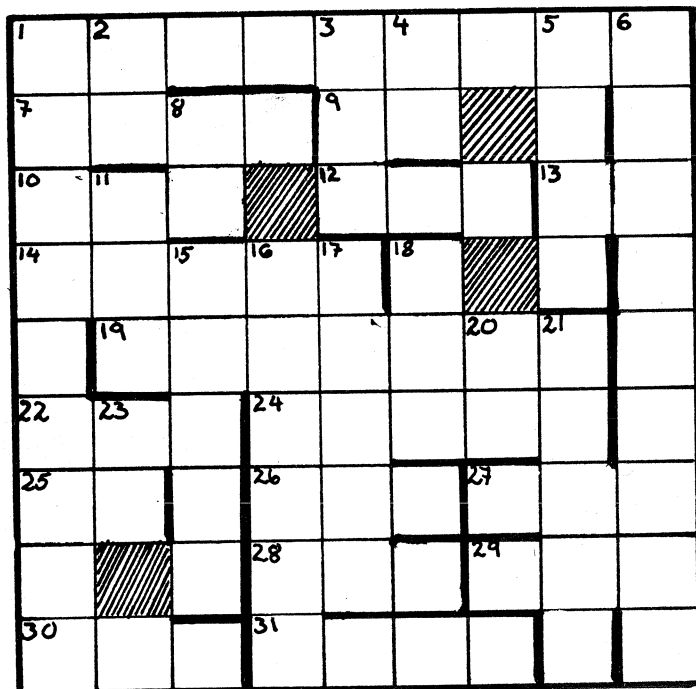
Vårt DATA-KRYSS stiger i popularitet og antall riktige løsninger i samme takt. Denne gang kom det inn hele 19 riktige svar, så heretter ser vi oss nødt til å trekke ut ti som får premie.

Følgende hadde riktig løsning (også Bergen!):

Lars E. Austvik, BIBSYS, RUNIT
M. Benes, SINTEF avd. 48
Ove Eriksen, Med. fak. - EDB-seksjonen, Haukeland sykehus, Bergen
Øystein Haugen, Studpost 73, NTH
Njård Hestnes, ELAB
Tron Hvaring, Inst. for numerisk matematikk, NTH
Ole Husby, BIBSYS, RUNIT
Bernt Marius Johnsen, IDB
Kråkmo & Lundquist, Adresseavisens EDB-avdeling, Trondheim
Leif Kruse, Norsk Data, Trondheim
Eirik Lien, EDB-tjenesten for humanistiske fag, UNIT, Dragvoll
Birger Morland, NSFI
Leif Næss, Inst. for dampteknikk, NTH
Oraklene, Orakelkontoret
Robert Strøm, VHL
Joar Støbakk, VHL
Dag Vareide, Inst. for dampteknikk, NTH
Ole Petter Vik, Control Data, Oslo
Agnar Aamodt, BIBSYS, RUNIT

PREMIE ER AVSENDT!

DATA-KRYSS



Det har kommet en del kommentarer om at våre "DATA-KRYSS" har vært for lette. Håper dette blir et hakk vanskeligere!

Løsning sendes RUN-NYTT's redaksjon innen 1 juni 1982.

Det vil bli trukket ut ti riktige løsninger som premieres!

```

*****
*
* Navn.....
*
* Adr.....
*
*****

```

VANNRETT

- 1 Dataprodukt
- 7 Datafirma
- 9 Datafirma
- 10 Datafirma i midten
- 12 Forskerhule
- 13 Berømt tall
- 14 Geografisk del av datafirma
- 19 Oversett
- 22 Moderne input
- 24 Offentlig datatjeneste
- 25 Datafirma
- 26 Datakvittering
- 27 Dataprodukt
- 28 16 i fokus
- 29 Firma i kjede
- 30 Dataprodukt (omv.)
- 31 Engelsk last

LODDRETT

- 1 Datafirma
- 2 Mesterskap
- 3 Datafirma
- 4 Datafirma
- 5 Langt medium
- 6 Dataprodukt (omv.)
- 8 Like
- 11 Datafirma
- 15 Datafirma (omv.)
- 16 Datafirma
- 17 Tomt for data
- 18 Litt data
- 20 Operator
- 21 Dataprodukt (omv.)
- 23 Datafirma