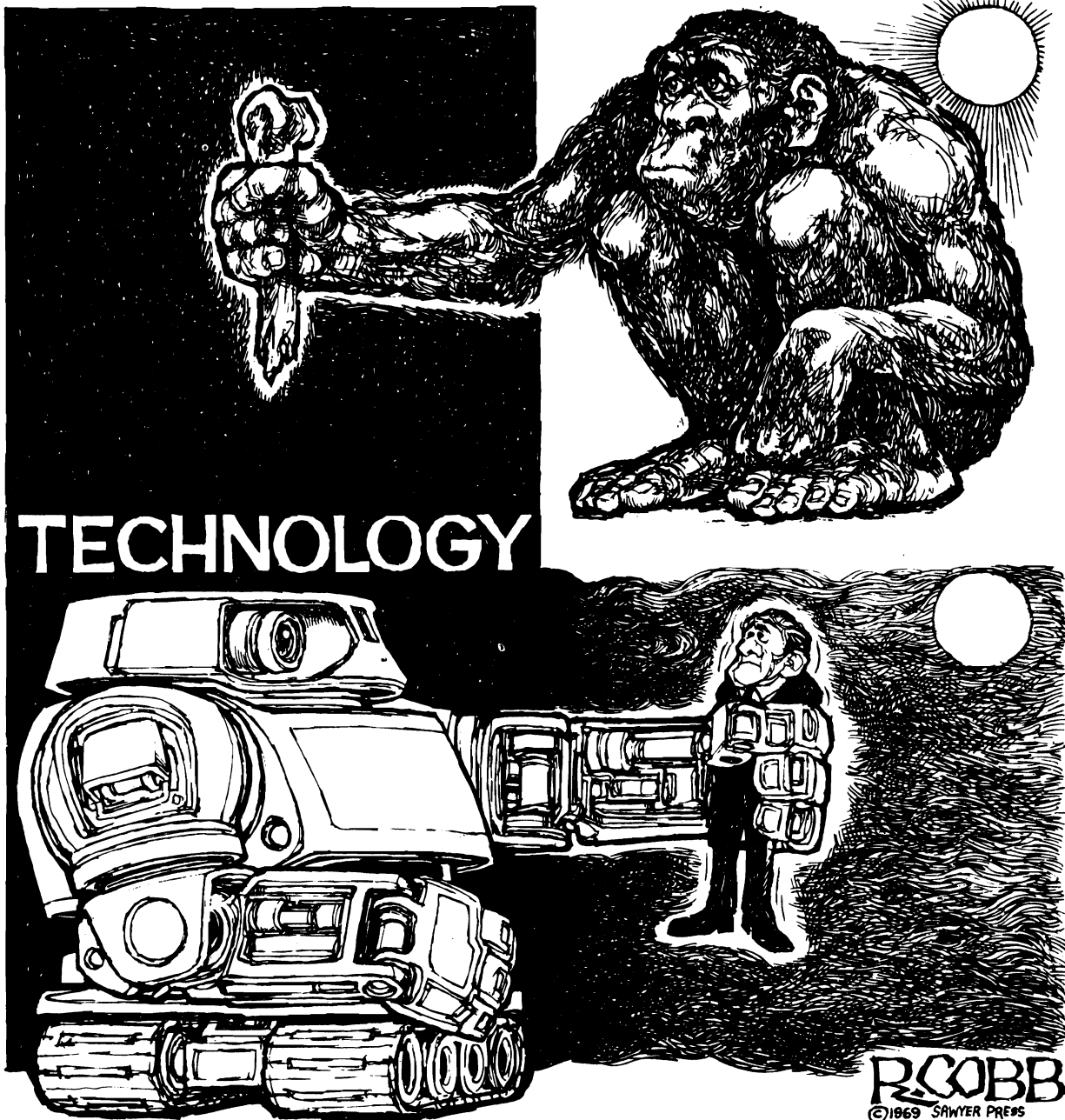


RUN·NYTT

INFORMASJONSORGAN FOR RUNIT,
REGNESENTRET VED UNIVERSITETET I TRONDHEIM

NR.4
ÅRG.5
14.DES.1978



TECHNOLOGY

R. COBB
©1969 SAWYER PERS
ALL RIGHTS RESERVED

DATAKUNST

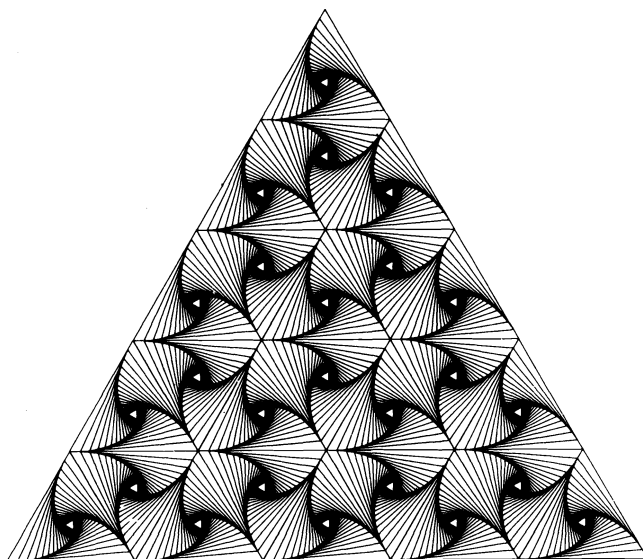
Datakunstutstillingen ved NTH

er nå vel overstått, og må sies å ha vært en "brukbar suksess". Utstillingen sto i nesten 20 dager, og totalt regner vi med at ca. 2000 personer besøkte utstillingen. En del skoleklasser var blandt de besøkende, og såvidt vi kunne oppfatte ble det pratet en del datakunst rundt om i miljøet.

En del av hensikten med utstillingen var å få arkitektavdelingen interessert i å bruke datamaskin som verktøy i sitt arbeid, men desverre så utstillingen ut til å være et mislykket forsøk i så henseende. Gjesteforelesningen om datakunst var lagt opp slik at den ikke krevde forkunnskaper om databehandling, men av de bortimot 100 tilhørerne var det bare et par stykker fra arkitektavdelingen.

Hvor mye av det som går under betegnelsen "Computer art" som er kunst, skal vi ikke uttale oss om, men at det ved hjelp av datamaskin går an å fremkalle bilder som gir visse spesielle synsinntrykk, burde det ikke være tvil om.

De fem lokale bidragsyterne beviste også dette, og vi gjengir her to av bidragene med en forklaring på hvordan de er fremkommet.

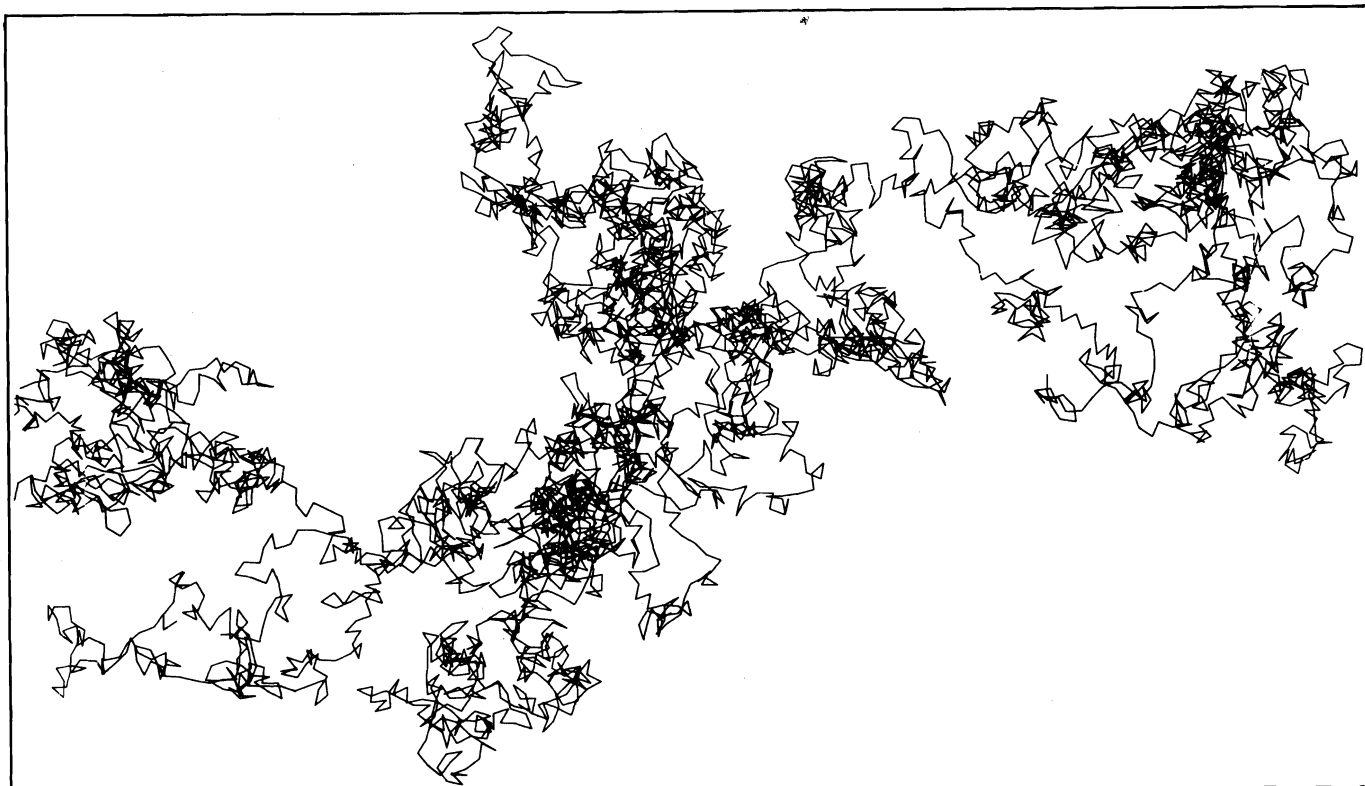


"Trekant" av Jens Andersen, 5. data

Et sett av likesidete trekanter er dreid og samtidig forminsket, utført i flere trinn, og trekantene er tegnet opp for hvert av trinnene.

"Drunkard's walk" av Einar J. Aas, ELAB

"Datamaskinen" har gått skritt med konstant lengde, men valgt en tilfeldig retning hver gang. Retningen er en stokastisk variabel fra en uniform fordeling i intervallet 0 - 360, det vil si tilfeldig trukket hver gang i området 0 til 360 grader.



RUN-NYTT

Redaktør : Arne Asphjell
 RUNIT
 7034 Trondheim-NTH
 Tlf. 075 92997

Redaksjons-
 assistent : Anne B. Sivertsen
 Tlf. 075 93003

Utkommer : Uregelmessig foreløpig.

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

Bidrag : Mottas med takk!

ETTERTRYKK TILLATT NÅR KILDE ER ANGITT.

INNHOOLD:

	side
Datakunst	2
Fremtidsvyer	3
Hvilken FORTRAN-oversetter bør jeg velge?	4
En brukers erfaring med ASCII-FORTRAN	6
RFOR er ikke "quick and dirty"	7
Redningsaksjon for slettede elementer	8
Redningsaksjon for ødelagte filer	9
Bok om feilmeldinger	9
Hvor er ASCII-terminalene?	10
Stansing av hullkort på 1100/21	10
Enklere tilknytning av terminaler	10
Magnetbånd fra andre installasjoner	10
Ny kjøreplan	11
Drodlesvar	11
ASCII-COBOL level 4R1	11
RUNIT er perfekt!	11
Neste RUN-NYTT	11
MARY - et maskinorientert språk	11
Teknologiske utviklingstendenser	14
Dataforkortelsen: OCR	15
NAG - nytt numerisk programbibliotek	16
Håndbok om bruk av interaktiv terminal	17
Billigere kjøringer i desember og januar	17
Forside og bakside	17
Lurt og surt med SPSS	18
Ny SPSS-versjon	18
Orakeltjenesten	19
Julenøtter	20
Kryssord	21
Hvor er RUNIT?	22
Veiledningstjenesten	23

FREMTIDSVYER

Ved RUNIT gjennomføres nå et prosjekt "RUNIT-plan for 1979-82". Prosjektet tar sikte på å få beskrevet

- RUNITs oppgaver i 1982
- RUNITs organisasjonsform i 1982
- Aksjonsplaner for perioden 1979-82

Arbeidet hittil har i vesentlig grad konsentrert seg om det første punktet "RUNITs oppgaver i 1982". I avtalen med UNIT er RUNITs hovedoppgaver:

- å yte EDB-tjenester til universitetsmiljøet i Trondheim
- å drive forskning og utvikling innen EDB og EDB-anvendelser

Det synes ikke på nåværende tidspunkt å være grunn til å forandre på disse hovedoppgavene. Imidlertid må fremtidige EDB-tjenester og områder for F&U gjennomdiskuteres.

Generelt sett kan vi si at utviklingen hittil har vært å stille nødvendig data-maskinkraft til disposisjon for brukerne. Fremtiden vil bli dominert av funksjonsrettede tjenester til brukerne. Brukerne må få det best egnede hjelpemiddel til å løse sine primæroppgaver.

I arbeidet med RUNIT-plan har en derfor tatt utgangspunkt i hvorledes forholdene bør være for brukerne i miljøet og har formulert følgende utsagn:

"UNIT-miljøets brukere skal ha tilgang til veltilpassede databehandlingstjenester for å løse sine oppgaver fra sine egne arbeidsplasser. Arbeidsplassene bør ha egnet utstyr for å utnytte tjenestetilbudet."

Det er utarbeidet et notat ved RUNIT som med bakgrunn i den teknologiske utvikling skisserer de fremtidige forholdene i UNIT-miljøet og hvilke tjenester som kan tilbys fra RUNIT. For å diskutere saken med brukerne i miljøet har en i tillegg til DMU opprettet en rekke referansegrupper. Notatet som danner diskusjonsgrunnlaget er i forslagsform allerede diskutert med referansegruppene og kommentarer innarbeides. Tidlig i januar 1979 vil notatet bli sendt alle brukerne i miljøet gjennom et eget RUN-NYTT og blir så fulgt opp med nye diskusjoner i referansegruppene. På bakgrunn av disse diskusjonene vil det så bli utarbeidet et endelig forslag til EDB-tjenester i UNIT-miljøet og hvilke tjenester som skal tilbys av RUNIT.

RUNIT håper på positivt engasjement fra brukerne i denne saken, slik at vi i fellesskap kan finne de beste løsningene for miljøet og RUNIT.

Per Arne Iversen

Hvilken FORTRAN-oversetter bør jeg velge?

Det eksisterer flere FORTRAN-oversettere på UNIVAC-anleggene, men de mest kurante er:

FORTRAN V (@FOR og @BIGFOR)
ASCII-FORTRAN (@FTN)
Reentrant FORTRAN (@RFOR)

Valget mellom tre kan også være vanskelig nok, og vi vil derfor prøve å utdype forskjellene mellom de tre.

FORTRAN V

FORTRAN V er den gamle sliteren som har vært med fra "UNIVACs morgen" her ved NTH. Denne lider av de fleste mangler FORTRAN som programmeringsspråk er blitt kritisert for, men har den fordel at den er bra standardisert.

FORTRAN V er en utvidelse av den standardiserte FORTRAN IV, som er en variant som de fleste datamaskinleverandører har implementert. I teorien skal et program skrevet i FORTRAN V noenlunde problemfritt kunne kjøres på andre maskiner som har FORTRAN IV. Vi sa i teorien, for i praksis viser det seg at UNIVAC opererer med sine spesielle særegenheter som kan skape komplikasjoner når programmer flyttes. Men FORTRAN V er i alle fall den av de tre som skaper minst problemer når programmer overflyttes til andre anlegg.

ASCII-FORTRAN

ASCII-FORTRAN er det nyeste skudd på stammen. Begrepet ASCII kommer inn p.g.a. at denne versjonen benytter fullt tegnsett med både store og små bokstaver, i samsvar med "American Standard Code for Information Interchange" (ASCII).

I ASCII-FORTRAN er det forsøkt å utbedre mange av de mangler som språkeksperter mener hefter ved FORTRAN, bl.a.:

- . CHARACTER data type, store og små bokstaver og tegnoperasjoner
- . IF-THEN-ELSE-ELSEIF-ENDIF konstruksjoner som reduserer bruken av GO TO (Fortran 77 standard)
- . direkte eksekvering (uten kollektoring)
- . mulighet for interaktiv feilfinning
- . mindre restriksjoner på programstørrelse

Optimaliseringsgraden av den genererte kode kan bestemmes med forskjellige opsjoner i kallet av oversetteren.

ASCII-FORTRAN er nærmere beskrevet i en annen artikkel i dette nummer av RUN-NYTT.

Den versjon av oversetteren som ble tatt i bruk våren 78 er i samsvar med den vedtatte nye FORTRAN-standard, FORTRAN 77.

Serien med ASCII-oversettere er under utbygging fra UNIVACs side, og det er disse det satses mest på å utvikle. Det er også laget ASCII-COBOL og ASCII-PL/1. Felleis for alle er at de er "reentrante" (samme kode kan brukes av flere brukere samtidig), og har felles inn/ut-system og matematikkbibliotek. Det vil si at hvis et ASCII-FORTRAN-program og et ASCII-COBOL-program ligger inne i primærlageret samtidig, vil de benytte samme kodebit for I/U-systemet.

Reentrant FORTRAN

RFOR ble opprinnelig utviklet ved et universitet i USA som hadde bruk for en mindre ressurskrevende og reentrant FORTRAN-oversetter til studentbruk. Oversetteren er nå et standard UNIVAC-produkt.

Kort sagt kan en si at RFOR er "quick and dirty" i forhold til de to andre. RFOR er først og fremst tiltenkt interaktiv bruk, i og med at den er av den "konversasjonelle" typen oversettere. Det vil si at kodelinjene blir kompilert etterhvert som de blir skrevet inn fra terminalen, slik at hvis det er feil kan brukeren rette på dette med en gang. RFOR oppdager selv om brukeren arbeider interaktivt og vil da automatisk oversette i konversasjonell modus.

Den direkte oversettingen medfører at RFOR vil ha mindre mulighet til å optimalisere den oversatte koden, med den følge at eksekveringstidene kan bli noe lengre.

I likhet med ASCII-FORTRAN har også RFOR interaktive feilfinningshjelpemidler innlagt. Dette er f.eks. sporing, dvs. en får utskrevet linjenumrene i den rekkefølge de blir utført. Det er mulig å stoppe utførelsen der en ønsker, og skrive ut verdien av variable og om ønskelig forandre disse.

FORTRAN forts.

RUNIT har utført en del prøver for å sammenligne ytelser av FORTRAN-oversetterne, og resultatet er vist i tabellen nedenfor (FORTRAN V er brukt som referanse):

Oversetter	Kall	Størrelse oversetter K ord	Kompi- lerings- tid	Ekse- kverings- tid
FORTRAN V	@FOR,IS	36	1.0	1.0
ASCII-FOR	@FTN,S	52	1.3	1.3
"	@FTN,VS (mer optimalisering)	54	1.8	0.9
"	@FTN,ZS (enda mer optimalisering)	55	2.3	0.8
RFOR	@RFOR,S	14	1.1	1.1

Referanser til håndbøker:

FORTRAN V:

- UNIVAC Fundamentals of FORTRAN, Programmer Reference, UP-7536
- Sperry UNIVAC 1100 Series FORTRAN V, Programmer Reference, UP-4060
- UNIVAC FORTRAN V Summary, UP-7951 (lommeformat)
- Arne Sølvberg: FORTRAN

ASCII-FOR:

- Sperry UNIVAC 1100-series FORTRAN (ASCII), Programmer Reference, UP-8244 Rev. 1
- UNIVAC FORTRAN (ASCII) Summary, UP-8245 (lommeformat)

RFOR:

- Sperry UNIVAC 1100 Series FORTRAN V, Programmer Reference, UP-4060, Rev. 2, Appendix G.
- 1001 måter å feile med FORTRAN, Tapir.

De viktigste forskjeller mellom FORTRAN-oversetterne:

Konstruksjon	FORTRAN V	ASCII-FOR	RFOR
Character data type	nei	ja	ja
Seriekompilering	nei	ja	ja
Feilfinningshjelpemidler	nei	ja	ja
Interne subprogram	ja	ja	nei

Tabell for å avgjøre hvilken oversetter som passer best:

	FORTRAN V	ASCII-FOR	RFOR
Kompatibilitet med andre installasjoner		x	
Gode muligheter for feilsøking		x	x
Enkle, mindre ressurskrevende oppgaver (studentkjøringer)			x
Større programmeringsoppgaver der kompatibilitet er av mindre betydning		x	
Høy optimalisering av generert kode			x

For nærmere beskrivelse av forskjellene mellom oversetterne henvises til:

FORTRAN V - ASCII-FOR:

Sperry UNIVAC 1100 Series FORTRAN (ASCII), Programmer Reference UP-8244 Rev.1, Appendix A

FORTRAN V - RFOR:

Sperry UNIVAC 1100 Series FORTRAN V, Programmer Reference, UP-4060 Rev. 2, Appendix G, avsnitt G4

FORTRAN - ANSI STANDARD FORTRAN:

Sperry UNIVAC 1100 Series FORTRAN V, Programmer Reference UP-4060 Rev. 2, Appendix C

Virker ikke for FORTRAN V og RFOR:

Mellomrom som skilletegn mellom negative tall i fritt format. Komma må brukes. Takk til Markussen, VHL som sa fra om dette.

En brukers erfaring med ASCII-FORTRAN



Einar Arne Herland fra ELAB forteller her om sine erfaringer med bruk av ASCII-FORTRAN:

Etter å ha brukt FTN-prosessoren ein del frå om lag april 78, har eg gjort ein del positive og ein del negative røynsler som vil kunne vera av interesse for potensielle brukarar. Årsaka til at eg kom bort i denne prosessoren, var at eg hadde behov for dobbel presisjon av komplekse tal. Av samme grunn har eg sett at ein del andre har skrive programma på reell form for å kunne bruke FORTRAN V. Eg fekk imidlertid eit tips om at FTN har dette innebygd, og dermed var det duka for skifte av kompilator. En annan ting som har gjort det påkrevd å bruke FTN, er bruk av funksjonar som feilfunksjonen (ERF) og gamma-funksjonen. Desse to tinga er årsak til at eg har brukt FTN. C-opsjon for direkte utføring av programmet kan vera praktisk sidan ein ellers alltid må kollektare med MAP sidan systemet ellers vil kalle SYS\$*RLIB\$ istaden for ASCII*FOR. Enkelte ting som det er viktig å vera merksam på, er førde opp nedanfor.

1. FORTRAN V rutiner kan kallast frå FTN ved å skrive EXTERNAL *SUBFOR, der SUBFOR er rutinenamnet for FORTRAN V rutina. Årsaka til at ein treng * framfor namnet, er at FTN og FORTRAN V har ulike adresseringssystem ved kall av rutiner. Dette gjev opphav til eit av dei mest intrikate problema ved kombinasjon av FTN og FORTRAN V. Entrepunkta for tilsvarande standardrutinar i FTN og FORTRAN V har nemleg samme namn.

Eit eksempel: Eit FTN-program kallar FORTRAN V rutina SUBFOR. SUBFOR kallar så standardrutina EXP. Deklarasjonen EXTERNAL *SUBFOR gjev riktig adressering til rutina SUBFOR, men kallet av EXP i SUBFOR er i FORTRAN V. På grunn av LIB ASCII*FOR under kollekteringa er det imidlertid ASCII sin EXP som er kollektert, med typisk GUARD MODE ERROR som fylgje. Dette kan ein unngå ved å skrive IN SYS\$*RLIB\$.EXP (forutsatt at EXP er elementnavnet) før LIB ASCII*FOR. Ein må altså kjenne alle standardrutiner som inngår i dei FORTRAN V rutinene som inngår, og inkluderer desse særskilt, før LIB ASCII*FOR. Dersom ei standardrutine blir kalt både frå FTN og FORTRAN V i samme program, er denne metoden ikkje brukbar, sidan ein då har berre eit entrepunkt, men skal ha 2 rutiner. I dette tilfellet kan ein prøve å dele opp programmet ved å legge mellomresultat på fil, slik at ein har 2 køyringar, og dermed får skilt standardrutinene frå kvarandre. Dersom ein på denne måten ikkje har behov for standardrutiner i FTN, kan ein skrive

```
@MAP
.
LIB SYS$*RLIB$
LIB ASCII*FOR
.
```

vel å merke i den viste rekkefylgja. Ved å legge mellomresultatet på fil med NTRAN, skulle ein kunne dele opp i eit FTN og eit FORTRAN V program. Dette har eg ikkje prøvd sjølv.

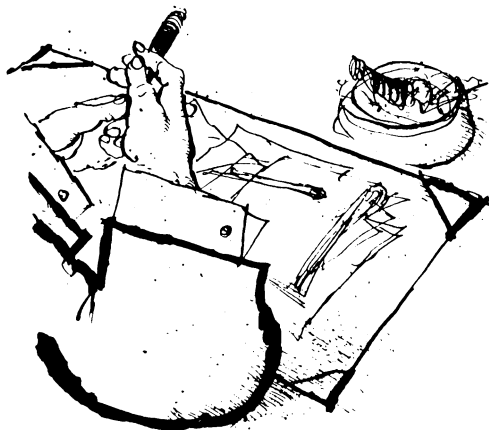
2. DEFINE FILE er ikkje kompatibel mellom FTN og FORTRAN V. På grunn av skrijving av kontrollord på filen ved definisjonssetninga i programmet, blir filen øydelagt ved kall frå motsett prosessor av den den vart oppretta under. Filen er ikkje lenger lesbar frå nokon av systema. DEFINE FILE er altså ikkje brukbar for lagring av mellomresultat dersom ein vil bruke både FTN og FORTRAN V som nemnt ovanfor.

Til slutt kan eg nemna at den som bruker FTN bør vera budd på å sjå feilutskrifta QUARTER WORD MODE NOT ALLOWED ON THIS SYSTEM under utføring av programmet. Årsaka er ein annan feil i programmet, men på grunn av denne misvisande feilutskrifta kan det vera litt ekstra vanskeleg å finne den verkelege feilen.

FORTRAN forts.

Dei egentlege problema med FTN får ein etter mi meining ved blanding av FTN og FORTRAN V, noko som er unngåeleg no sidan programbiblioteket er i FORTRAN V. Dersom ein er merksam på det som er nevnt under 1) ovanfor, skulle ein imidlertid for største-delen kunne unngå problem her og. Ved problem er det imidlertid ikkje så lett å få råd frå Regnesentret som ein kunne ynskje, men dette vil nok endre seg dersom fleire brukar FTN. Dermed er det berre å anbefale dei som måtte ha lyst og behov til å gjera nytte av dei ekstra tinga FTN har å by på.

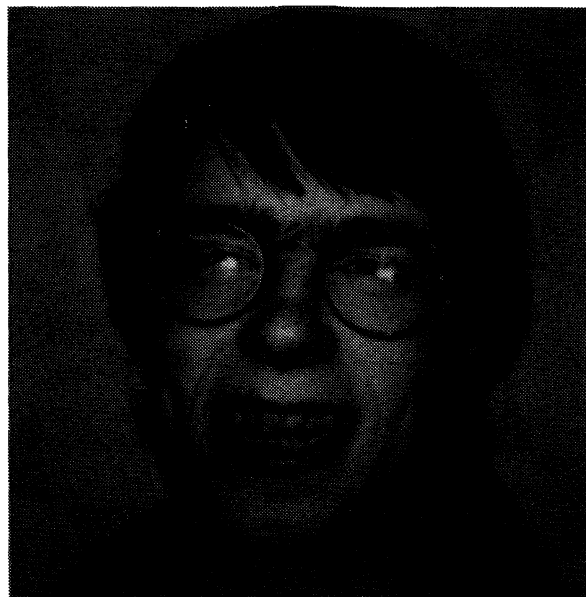
Einar Arne Herland
ELAB



Om jeg kjenner FORTRAN?
Jo, jeg har da hørt om ham . . .

RFOR er ikke 'quick and dirty'!

sier diplomstudent Stig Ole Johnsen:



RFOR er utvidet i forhold til FORTRAN 4 standarden, men holder man seg til denne standarden, så er RFOR problemfri, og program kan selvsagt flyttes mellom maskiner.

Som orakel har jeg aldri støtt på RFOR-program som har blitt galt oversatt.

Jeg har støtt på FOR-program som har blitt galt og meningsløst oversatt. @RFOR er derfor å anbefale framfor @FOR, både på grunn av bedre feilmeldinger og bedre avlusings-system.

Av og til har jeg sett program som har blitt compilert feilfritt av @FOR, men med påfølgende feilutgang under eksekvering. Ved å oversette samme program med @RFOR har vi da fått feilmeldinger/advarsler som har gjort oss istand til å rette opp angjeldende feil.

Skummel feil i FORTRAN V:

```
DO 100 I=1,10
```

↑ null, skal være 0 for Olav

Dette gir en "warning", men setningen oversettes, og det gir en merkelig kode. Vær på vakt!

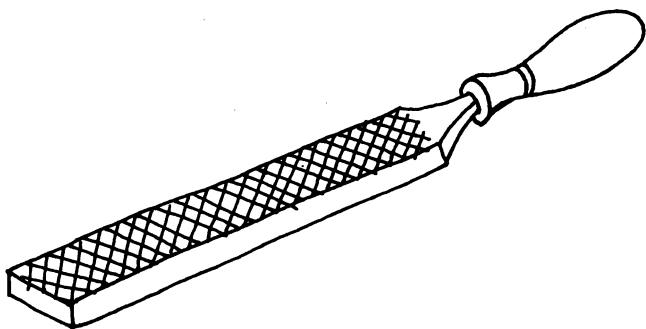
I RFOR virker ikke

```
PARAMETER AARSTL=1978, DAAR=AARSTL
```

Det gir en feilmelding som forteller at parameterdeklarererte størrelser ikke selv kan benyttes innenfor parameter-deklarasjon.

ET APROPOS TIL BRUSÅPNEREN SOM FORSVANT:

Filen som ble vekkk



Redningsaksjon for slettede programelementer

Det hender at en er så uheldig å slette et program fra en programfil uten at dette var meningen. Dette kan for eksempel skje ved introduksjon av et nytt element med samme navn som et tidligere.

Under forutsetning av at det ikke er fysisk fjernet fra filen, kan et slikt element bringes tilbake til "u-slettet" tilstand. Forhold som gjør redningsaksjonen vanskelig eller umulig, er

- . @PACK er brukt etter at elementet ble slettet
- . @ERS er brukt etter at elementet ble slettet
- . Filen er slettet fra katalogen. (@FREE,D eller liknende)

Når vilkårene for vellykket aksjon er tilstede, gjøres det slik:

```
@RUNIT*PROG.PRINT,G PFIL.PROG4
```

Her er det elementet PROG4 på programfilen PFIL som skal reddes. Alle slettede elementer med navn PROG4 vil bli reddet. For å skille dem fra hverandre, får disse elementene et versjonsnavn: DELA, DELB osv. På en innholdsfortegnelse (f.eks. med @PRT,T) kan en så plukke ut hvilket element som det er interessant å beholde, og så kan en slette de uinteressante elementene igjen.

Eksempel på "redningsaksjon" er vist nedenfor.

Eksemplet inneholder følgende trinn:

1. Innholdsfortegnelse. Det siste elementet -ble lagt inn ved en feiltagelse.
2. Redningsaksjon.
3. Innholdsfortegnelse. Resultat av aksjonen.
4. Forandrer navnet på det interessante elementet til det det var og skulle være.
5. Sletter uinteressante elementer.
6. Innholdsfortegnelse. Endelig resultat av redningsaksjonen.

```
>@RUNIT*PROG.PRINT,T PFIL.  
PRINT 02/75 - 28/08 13:00:17
```

①

```
<<<< RUNIT*PFIL. >>>>
```

ELEMENT/VER.	TYPE	DATE	TIME	TXT	SEQ
* PROG1(0)	(ELT)	28 AUG 78	12:55:55	1	1
* PROG2(0)	(ELT)	28 AUG 78	12:55:56	1	2
* PROG3(0)	(ELT)	28 AUG 78	12:55:57	1	3
* PROG2(0)	(ELT)	28 AUG 78	12:55:58	1	4
* PROG4(0)	(ELT)	28 AUG 78	12:56:00	1	5
PROG1(0)	(ELT)	28 AUG 78	12:56:01	1	6
PROG2(0)	(ELT)	28 AUG 78	12:56:02	1	7
* PROG4(0)	(ELT)	28 AUG 78	12:56:03	1	8
PROG5(0)	(ELT)	28 AUG 78	12:56:05	1	9
PROG3(0)	(ELT)	28 AUG 78	12:56:06	1	10
* PROG4(0)	(ELT)	28 AUG 78	12:56:08	1	11
PROG4(0)	(ELT)	28 AUG 78	12:56:44	1	12

```
>@RUNIT*PROG.PRINT,G PFIL.PROG4  
PRINT 02/75 - 28/08 13:01:29
```

②

```
<<<*PROG4 (ELT) NEW NAME IS: PROG4/DELA (ELT) >>  
<<<*PROG4 (ELT) NEW NAME IS: PROG4/DELB (ELT) >>  
<<<*PROG4 (ELT) NEW NAME IS: PROG4/DELC (ELT) >>  
<<< PROG4 (ELT) >>
```

```
>@RUNIT*PROG.PRINT,T PFIL.  
PRINT 02/75 - 28/08 13:01:57
```

③

```
<<<< RUNIT*PFIL. >>>>
```

ELEMENT/VER.	TYPE	DATE	TIME	TXT	SEQ
* PROG1(0)	(ELT)	28 AUG 78	12:55:55	1	1
* PROG2(0)	(ELT)	28 AUG 78	12:55:56	1	2
* PROG3(0)	(ELT)	28 AUG 78	12:55:57	1	3
* PROG2(0)	(ELT)	28 AUG 78	12:55:58	1	4
PROG4/DELA(0)	(ELT)	28 AUG 78	12:56:00	1	5
PROG1(0)	(ELT)	28 AUG 78	12:56:01	1	6
PROG2(0)	(ELT)	28 AUG 78	12:56:02	1	7
PROG4/DELB(0)	(ELT)	28 AUG 78	12:56:03	1	8
PROG5(0)	(ELT)	28 AUG 78	12:56:05	1	9
PROG3(0)	(ELT)	28 AUG 78	12:56:06	1	10
PROG4/DELC(0)	(ELT)	28 AUG 78	12:56:08	1	11
PROG4(0)	(ELT)	28 AUG 78	12:56:44	1	12

```
>@CHG,S PFIL.PROG4/DELC,.PROG4
```

```
>@DELETE,SV PFIL.PROG4/DELC*  
FURPUR 27R3 E33 SL73R1 08/28/78 13:03:43
```

```
>@RUNIT*PROG.PRINT,T PFIL.  
PRINT 02/75 - 28/08 13:04:36
```

```
<<<< RUNIT*PFIL. >>>>
```

ELEMENT/VER.	TYPE	DATE	TIME	TXT	SEQ
* PROG1(0)	(ELT)	28 AUG 78	12:55:55	1	1
* PROG2(0)	(ELT)	28 AUG 78	12:55:56	1	2
* PROG3(0)	(ELT)	28 AUG 78	12:55:57	1	3
* PROG2(0)	(ELT)	28 AUG 78	12:55:58	1	4
* PROG4/DELA(0)	(ELT)	28 AUG 78	12:56:00	1	5
PROG1(0)	(ELT)	28 AUG 78	12:56:01	1	6
PROG2(0)	(ELT)	28 AUG 78	12:56:02	1	7
* PROG4/DELB(0)	(ELT)	28 AUG 78	12:56:03	1	8
PROG5(0)	(ELT)	28 AUG 78	12:56:05	1	9
PROG3(0)	(ELT)	28 AUG 78	12:56:06	1	10
PROG4(0)	(ELT)	28 AUG 78	12:56:08	1	11
* PROG4(0)	(ELT)	28 AUG 78	12:56:44	1	12

④
⑤
⑥

Redningsaksjon forts.

Det programmet som her er brukt, PRINT, er nærmere beskrevet i en egen brukerinstruks. Denne kan fås ved henvendelse til Per Balstad, RUNIT, Kjemihallen, tlf. 2992.

Merk at PRINT også inneholder mange andre muligheter. Det å redde slettede elementer er bare en biting. Brukerinstruksen er verd et gjennomsyn.

Redningsaksjon for ødelagte filer

Dersom det ulykkelige elementet er fysisk fjernet fra filen på grunn av noen av de kompliserende forhold som ble nevnt innledningsvis, - ja så kan det fremdeles være håp, dersom elementet finnes (slettet eller ikke) på sikkerhetskopien av filen. Man kan selv ta inn siste sikkerhetskopi på følgende måte:

```
@SECURE,ILF
REVERT FILE <qualifier>*<filnavn>.
@EOF
```

(NB! Husk å lagre unna eventuelle nye ting på filen, for nå erstattes hele filen med gårsdagens utgave).

Dersom filen er slettet fra katalogen, må den katalogiseres først.

Eksempel:

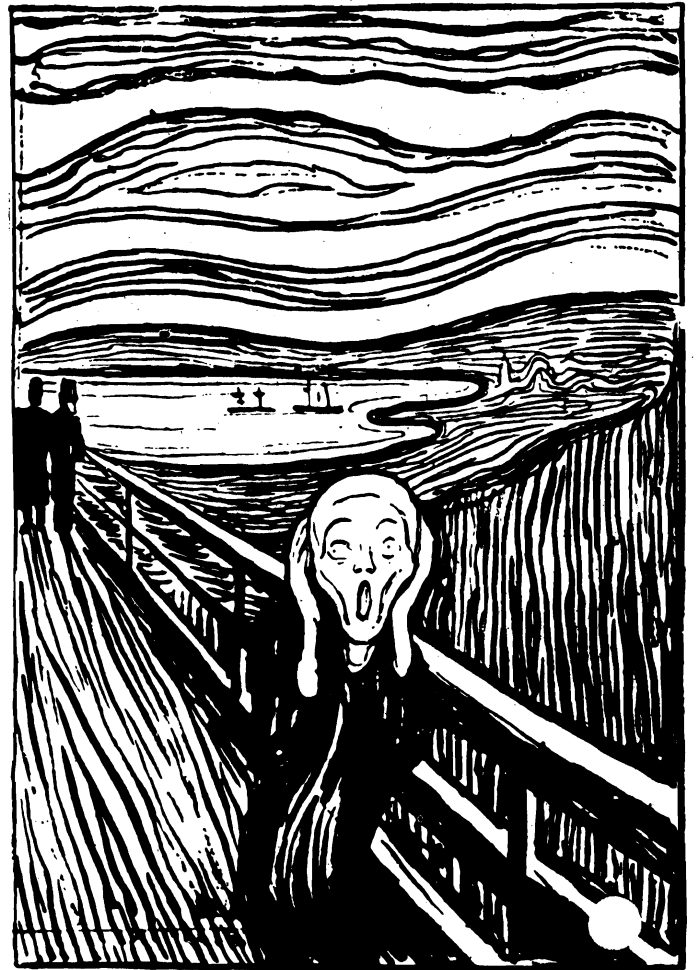
```
>@SECURE,ILF
SEC19R4 S73Q10 12/08/78 13:31:06
>REVERT FILE RUN*MINFIL.
1. REVERT FILE RUN*MINFIL.
>@EOF

SECURE SUMMARY LISTING:
RUN*MINFIL(1) UNLOADED AT 13:31:41 TRK SIZE = 6 UEF
= 63

END OF SECURE - TIME 0.133 SECONDS
>@ASG,A MINFIL.
WAITING ON FACILITY
@X OT
*EXECUTION TERMINATED*
Prøver igjen en halvtime senere:
>@ASG,A MINFIL.
READY
```

operativsystemet må først sørge for at filen leses inn fra magnetbånd.

Hvis heller ikke dette går, kan en søke hjelp hos driftsassistent Reimers, Lerken- dal, tlf. 3022.



FORTRAN-programmerere! Fortvil ikke!

Det finnes nå et hjelpemiddel for å tyde feilmeldingene du får fra FORTRAN-programmene dine.

"1001 måter å feile med FORTRAN" fra Tapir forlag - kr. 20,-.

Den boka behandler de fleste feilmeldingene du kan få når du kjører datamaskinprogram på UNIVACs 1100 serie.

Litt teori og bakgrunn, men først og fremst blir feilmeldingene forklart. (Boka har indeksregister!) Dessuten blir det gitt en innføring i de feilfinningshjelpemidler som finnes på datamaskinen.

A PROGRAMMER'S LAMENT

I really hate this damned machine;
I wish that they would sell it.
It never does quite what I want,
But only what I tell it.

HVOR ER ASCII-TERMINALENE ?

Av terminaler med fullt ASCII tegnsett til rådighet for våre brukere, har vi foreløpig en teksts skjerm av typen B-100. Denne er plassert i terminalrom for brukere i 2.etg., SB-II.

Det finnes også en skrivende terminal av typen Diablo 1620. Denne egner seg spesielt til skriving hvor en har behov for ekstra god skrift. Terminalen står i RUNITs lokaler i Kjemihallen. De som måtte ønske å benytte denne må kontakte A. Asphjell, tlf. (9)2997.

Ellers finnes en linjeskriver med mulighet for fullt ASCII tegnsett. Denne står i datamaskinhallen på Lerkendal. Linjeskrivoren kan brukes både på U 1108 og U 1100/21.

Brukere må benytte følgende oppsett:

```
@SYM PRINT$, ,ASCPR
```

Utskriftsfilen vil normalt bli skrevet ut etter kl. 16.00. Det blir da montert skrivekjede med store og små bokstaver. Dessuten vil papiret bli snudd, slik at teksten bli skrevet på hvitt papir og ikke på gråstripet sebrapapir.

Forøvrig vises til RUN-nytt nr. 8, 1977 "Tekstredigering med DOC".

Snorre Torgnes

STANSING AV HULLKORT PÅ 1100/21

UNIVAC-anleggene ved RUNIT er utstyrt med en hullkort-stans som er tilkoblet UNIVAC 1108. Filer på UNIVAC 1100/21 som skal stanses på hullkort overføres automatisk via ICC-enheten. Under overføringen mistes informasjonen om hvilken bruker kortene tilhører. Kortene blir derfor lagt ut umerket. Dette er ofte lite tilfredsstillende, så vi anbefaler brukerne å bruke et @MSG-kort i slike tilfeller, f.eks.:

```
@MSG,W 250 KORT UT, TEKSTES .
```

Dette gir operatøren anledning til å merke kortstokken med brukernummer.

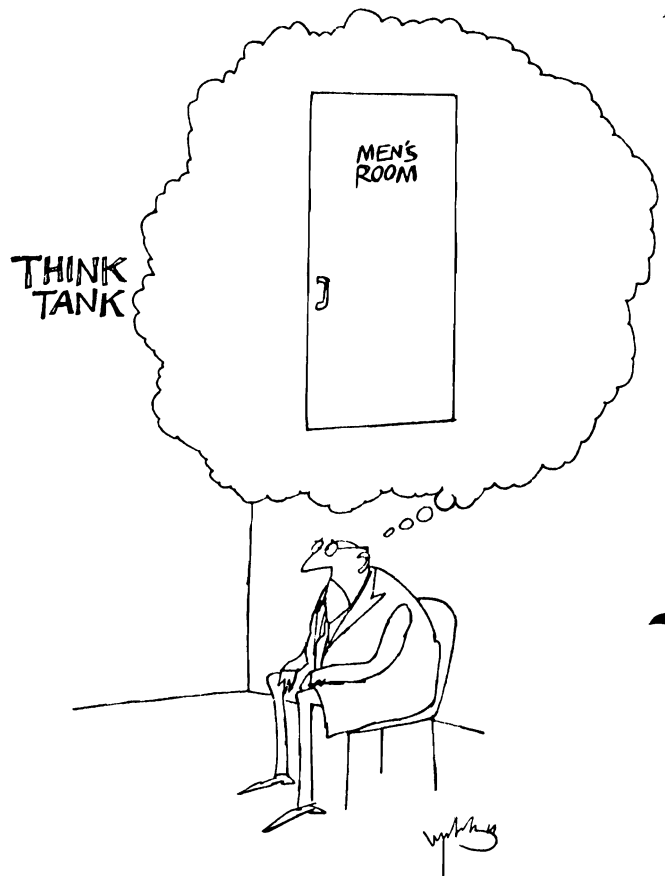
Per Ditlef Øvrebø

Enklere tilknytning av avanserte terminaler

En SM-4 maskin i Sentralbygg 2 vil nå bli klargjort for tilkobling av Tandberg-skjermer av type TDV 2115. FBT-systemet (FBT: Fleksibel Bruker-Terminal) i denne maskinen tillater Tandberg-skjermene å kjøre mot Univac som UNISCOPE 200 terminaler.

Blant egenskapene til UNISCOPE nevnes: Full skjerm modus, gode editeringsmuligheter på skjermen og beskyttede felt. Full skjerm modus betyr at en kan bearbeide et helt skjerm bilde lokalt før det oversendes til hovedmaskinen.

Kontakt: Trond Skjesol, tlf. 2983



MAGNETBÅND FRA ANDRE INSTALLASJONER

Skal du bestille program fra andre installasjoner og få disse oversendt på magnetbånd, bør du før bestilling kontakte:

Per Balstad
RUNIT - Kjemihallen
tlf. 2992

for å få rede på formater og lignende.

NY KJØREPLAN

Det er innført ny kjøreplan for UNITS data-maskinanlegg fra 1978-11-01, se Rinfo nr. 1.01.

De viktigste endringene er:

- Det er nå mulig å kjøre mot ubetjent anlegg.
- Hullbånd kan leses inn til og punches ut fra masselagerfil.
- Studentkjøring er tillatt fra interaktiv terminal, med visse begrensninger, når 1. del av grunnkurset er fullført.
- Antall jobbgrupper reduseres til det som kan kjøres i ordinær kontortid, og det som må kjøres utenom.
- Begrensningene for brukerkategorier forenkles.
 - 1.-4. årskurs har samme begrensninger.
 - Alle betalende brukere har åpne grenser.
 - Påbudte termineringsopsjoner for samtlige brukere unntatt de med typekode M.

NB! Korreksjoner til Rinfo nr. 1.01.

side 5: pkt 3.3 Begrensninger for jobber som skal kjøres mellom kl. 08.30-17.00:
Ikke over 4 min. CPU-tid

side 7: pkt 3.7 Studentkjøringer via interaktiv terminal: maksimal programstørrelse 15K ord før kl. 17.00, 30K ord etter kl. 17.00.

side 8: Navn på print/punch-grupper for NTR-terminaler:
Lade er tilkopleet 1108
Dragvoll er tilkopleet 1108

DRODLESVAR

Drodlene står på side 20.

- A) Hvis du gjettet at det var en mann som spiller trombone i en telefonkiosk, gjettet du riktig. Ved å dreie bildet 90 grader mot venstre, blir det : en avdød trombonespiller.
- B) Mann i festantrekk som har stått for nær heisdøren.
- C) Berømt vitenskapsmann sett gjennom mikroskop.
- D) Siden det er en datadrodle er vel denne ganske åpenbar: nærbilde av et hullkort.

ASCII COBOL LEVEL 4R1

Fra 1 januar 1979 tas ASCII COBOL Level 4R1 i bruk. Brukere som kompilerer med @ACOB og har LIB ACOB*LIB i sine MAP-program, får da denne versjonen som er basert på ANS-standard av 1974.

Nye brukermanualer er beskrevet i RUN-NYTT nr. 3, 9 oktober 1978.

Vil videre vil Level 3R1 av kompilatoren (nåværende versjon) ligge på filen ASC*II. Denne kalles ved @ASC*II.COB. Tilsvarende runtime-system vil ligge på filen ASC*LIB.

Det anbefales å konvertere til Level 4R1 så snart som mulig. ASC*II og ASC*LIB blir slettet etter nærmere kunngjøring.

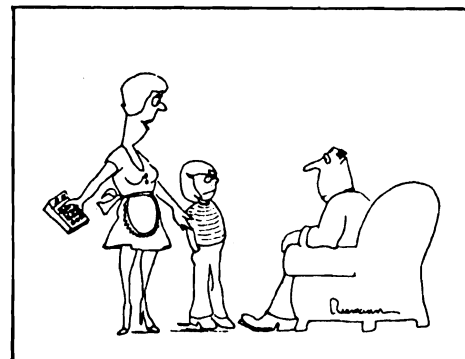
Arild Laugen

RUNIT er perfekt !

I løpet av ett år har vi fått bare et par lapper i våre forslagskasser. Det skulle vel tyde på at det er intet å utsette på våre tjenester, og at alle våre brukere er såre fornøyd?

NESTE RUN-NYTT :

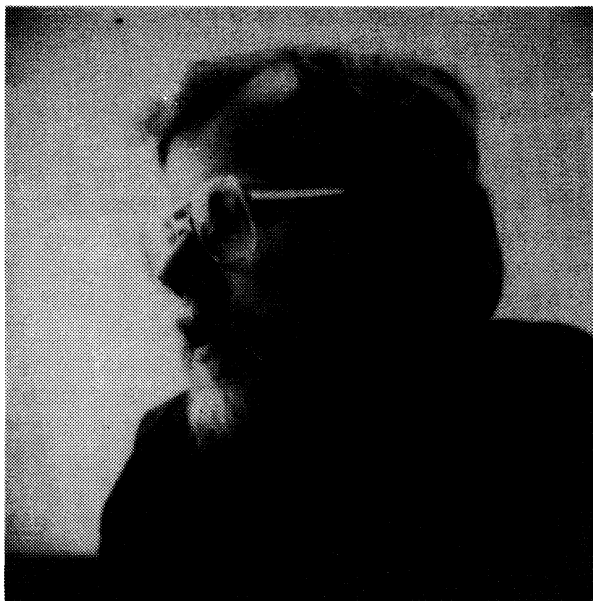
SPECIALNUMMER OM RUNITS PROFIL I 1980-ÅRENE



'He Flunked Math. His Batteries Failed.'

Språkreklamen:

MARY - et maskinorientert programmeringsspråk



Ole Solberg

Programmeringsspråket MARY - navnet er ingen forkortelse, det er virkelig oppkalt etter en dame - er RUNITs eget verk. Språket ble definert og den første oversetter for språket ble laget ved RUNIT. Dette arbeidet begynte i 1971.

Ildsjelen og den ustoppelige drivkraft i den første fase av prosjektet var en amerikaner ved navn Mark Rain. Denne vandrende programmerer dukket opp ved RUNIT i 1971 og målbandt alle med sine talegaver. Uten hans utrettelige optimisme og pågangsmot ville nok prosjektet avgått ved en stille død ganske raskt. Det viste seg nemlig at norsk data-industri ikke våget å støtte prosjektet. Etter fem magre år har vi nå etter hvert fått en viss støtte.

Et ALGOL-liknende språk

Ved første øyekast kan et enkelt MARY-program minne meget om et ALGOL-program. En viktig forskjell vil være at alle uttrykk i MARY beregnes konsekvent fra venstre mot høyre. Alle operatorer har samme prioritet. Tilordning ser ut som en helt vanlig operator. For eksempel vil en ALGOL-setning:

```
i=(i+2)*j
```

i MARY uttrykkes som:

```
i+2*j=:i;
```

Kan det være enklere?

Den som kan ALGOL (el. SIMULA) vil derfor etter, skulle jeg tro, en times opplæring i MARY kunne skrive enkle programmer.

Men om dette var hele forskjellen mellom MARY og ALGOL, da kunne vi jo klart oss uten MARY.

MARY er et "maskinorientert høynivå programmeringsspråk". Det er beregnet på programmeringsoppgaver der assembler tidligere var eneste mulighet. Det skal være mulig for programmereren å formulere problemet sitt ved hjelp av høynivå mekanismer for programflyt og for databeskrivelse (data typer). Samtidig skal han/hun ha full adgang til f.eks. å benytte alle registre på maskinen. Eksempler på høynivå mekanismer er IF-, DO-, CASE-setninger, rutiner, datamoder (data typer) (INTEger, REAL, BOOLEan, pekere osv.). I MARY kan en i tillegg til dette erklære nye datamoder (typer):

```
MODE LEDD = (REF LEDD NESTE-LEDD, INT
VERDI); (Ledd inneholder peker til neste
ledd og en tallverdi.)
```

En kan selvfølgelig også definere nye operatorer som virker på de nye datamodene:

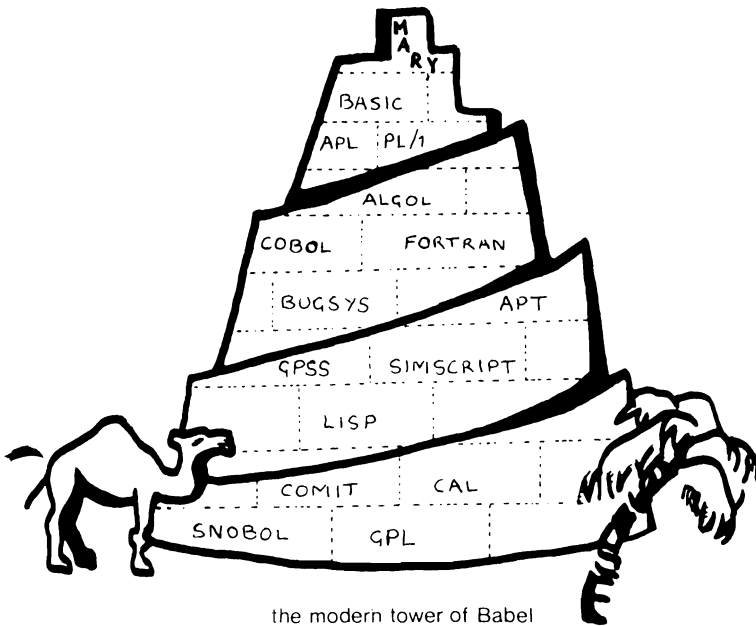
```
OP SETT-INN-LEDD (REF LEDD NYTT, KJEDE)=
(KJEDE =: NYTT.NESTE-LEDD; NYTT)$;
(Setter et nytt ledd først i en kjede og
gir den nye kjede som resultat.)
```

Programmereren kan på denne måten bygge opp et begrepsapparat for løsning av sitt problem.

I forbindelse med maskinær programmering kan man plassere variable (objekter) i registre eller på bestemte adresser i data-maskinens lager. Man kan direkte fra MARY programmere behandling av f.eks. I/O og avbrudd. Dette har man tidligere vært nødt til å gjøre i assemblyspråk.

I RUNIT-miljøet er oversettere for MARY tilgjengelig på NORD-10, SM4, KS500 og UNIVAC. De oversatte program kan kjøres på NORD-10, SM4 og KS500. For INTEL 8080 (MYCRO) finnes en eksperimentell oversetter som kjører på SM4. For den som måtte ønske å vite mer om programmeringsspråket MARY anbefales MARY textbook skrevet av Reidar Conardi og Per Holager.

MARY forts.



Ved Institutt for databehandling foreleses i høstsemestret kurset "Maskinær programmering" hvor MARY benyttes som verktøy.

Er MARY lett å lære?

Det viser seg i praksis at 2-3 måneders opplæring er tilstrekkelig for å utdanne "profesjonelle" MARY-programmerere. Det ser også ut til at MARY-programmerere blir minst like overbevist om at MARY er det perfekte språk, som andre om FORTRAN eller APL. Det har blitt hevdet at språket er for stort og komplisert og at det er vanskelig å lære, men da helst av de som ikke har forsøkt det.

MARY brukes flere steder i Norge

Noen vil kanskje tro at MARY er et språk som bare brukes som et leketøy av noen forskere ved RUNIT. Så er ikke tilfelle: Det benyttes eller har blitt benyttet ved Forsvarets Forskningsinstitutt, Televerkets Forskningsinstitutt, Computas, Universitetet i Tromsø, Kongsberg Våpenfabrikk og mange studenter ved Institutt for databehandling.

Vi bør spesielt nevne at Kongsberg Våpenfabrikk har valgt MARY som sitt språk for systemimplementasjon.

RUNIT har holdt flere kurs i MARY ved FFI, TF og KV. Ved KV er mellom 30 og 40 programmerere gitt opplæring i MARY de to siste årene.

Produktiviteten øker 3-4 ganger

Man har oppnådd å øke produktiviteten i programmeringen med en faktor 3-4 i forhold til bruk av assembler. Dette er tall målt ikke bare ved RUNIT, men også f.eks. ved Kongsberg Våpenfabrikk.

MARY er internasjonalt kjent

MARY har nådd ut over RUNIT-miljøet:

I 1973 holdt the International Federation of Information Processing, Technical Committee 2 (IFIP/TC-2) en arbeidskonferanse om maskinorienterte høynivåspråk i Trondheim, med RUNIT som lokal arrangør. At konferansen ble lagt til RUNIT var en anerkjennelse av MARY-arbeidet. Etter denne konferansen ble det opprettet en ny arbeidsgruppe innen IFIP (IFIP/TC-2 Working Group 2.4). Mark Rain, Per Holager og Reidar Conradi fra MARY-prosjektet er faste medlemmer av denne gruppen. "Machine Oriented Languages Bulletin" (MOLB) som ble startet av RUNIT ble også overtatt av IFIP Wg.2.4.

Ved Chalmers Tekniska Høgskola i Gøteborg er det laget oversetter for IBM 360/370.

I boken "Programming electronic switching systems" av Hills & Kano (1976) blir det hevdet at MARY "in many ways represents the state of the art in programming language design".

Da den internasjonale konsultative komite for telefon og telegraf (CCITT) i 1975-76 skulle bestemme seg for standard programmeringsspråk for datamaskinstyrte telefon-sentraler, var MARY blant de 8 språk som ble valgt ut av en gruppe på 27 for videre evaluering. Av politiske årsaker ble ingen av disse valgt, derimot ble det nedsatt en komite som definerte et nytt språk, CHILL. På grunn av vårt MARY-arbeid ble RUNIT engasjert av de nordiske teleadministrasjonene til å delta i denne komiteen. Vi har også blitt engasjert for å lage oversetter for dette språket.

TEKNOLOGISKE UTVIKLINGSTENDENSER

Innen feltet datamaskiner skjer utviklingen raskt, og spesielt på maskinvaresiden dukker det stadig opp mindre og billigere komponenter. Bjørn Ranum, som er leder for ADAB-80 prosjektet (Anskaffelse av databehandlingsutstyr for 80-årene), har den siste tiden studert teknologiske utviklingstendenser. Han vil i dette og kommende nummer presentere sine oppsummeringer for RUN-NYTT's lesere.

KOMPONENTTEKNOLOGI

Når en skal beskrive den teknologiutviklingen en i dag har innenfor databehandlingsfeltet, er det naturlig å starte med utviklingen av den grunnleggende komponentteknologi. For brukeren av et datamaskinsystem er det i prinsippet uinteressant hvilken teknologi som benyttes, men da den revolusjonerende utvikling på komponentsiden stort sett er grunnlaget for det som skjer innenfor hele databehandlingsfeltet, trenger den en nærmere beskrivelse.

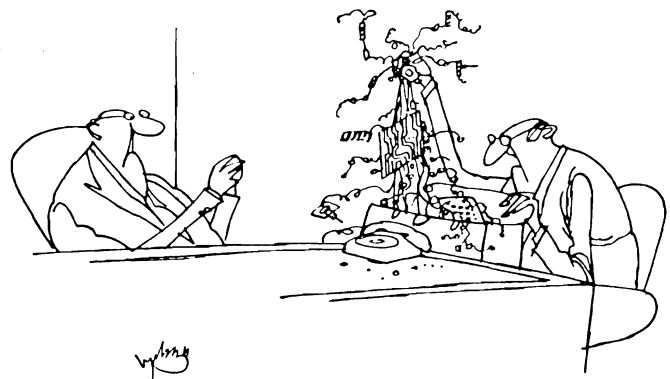
Logiske kretser

Flere av de datamaskinsystemer som vi i dag kjenner, f.eks. UNIVAC 1108, ble laget i 60-årene. Til konstruksjon av logiske kretser i disse maskinene ble det benyttet transistorer som hver representerer en

logisk port. I en maskin som UNIVAC 1108 er det mer enn 100.000 logiske portkretser. Den halvleder-teknologi som benyttes til fremstilling av storskala integrerte kretser (LSI), gjør det i dag mulig å legge ca. 50 000 logiske portkretser på en enkelt komponent av størrelsen 4 x 4 mm. Fram til 1982 forventer en at dette tallet vil øke til 250 000 eller mer.

Utviklingen av logiske kretser skjer hovedsaklig i to forskjellige former. Den ene er utvikling av logiske kretselementer (eng. "gate arrays") tilpasset spesielle behov. Disse benyttes særlig ved konstruksjon av større datamaskin-systemer hvor det settes store krav til hastighet. Den andre hovedretning er generelle programmerbare komponenter som betegnes mikroprosessorer. En mikroprosessor kan funksjonelt sett sammenlignes med sentralenheten (CPU) i en datamaskin.

De første mikroprosessorene kom på markedet omkring 1974. Disse var langsomme og opererte kun på 4 bits enheter. Siden da er det stadig kommet forbedrede og raskere versjoner. I løpet av 1978 er det annonsert en serie nye mikroprosessorer som opererer på 16 bits enheter. Instruksjonstider og adresseringsmuligheter til flere av disse er bedre enn det en finner i et minimaskin-system av typen NORD-10. I tillegg til selve mikroprosessorbrikkene finnes det komponentbrikker for inn/ut-behandling, lageradministrasjon osv. Ved hjelp av disse kan det lages komplette mikrodatamaskin-systemer. Det finnes også komponentbrikker som i seg selv er komplette mikromaskiner med CPU, lager og inn/ut-kontroll. Slike komponenter vil i fremtiden erstatte det meste av den diskrete logikk som tidligere ble benyttet.



"We don't have your system designed yet, but I brought along what we have."

Lagerteknologi

Ved siden av CPU, er lageret den viktigste delen av et datamaskinsystem. Ideelt sett ønskes en homogen, etnivås lagerstruktur, men på grunn av at kostnadene ved bruk av en bestemt lagerteknikk er høyere jo lavere adgangstiden til lageret er, ser en i dagens datamaskinsystemer et hierarki av lagersystemer.

Hovedlageret (primærlageret) i et datamaskinsystem ble frem til ca. 1975 laget vha. ferittkjerner. Produksjon av slike lagersystemer var spesielt personellkrevende, og dermed også kostbar. Ferittkjernelager krever dessuten stor fysisk plass og krever mye elektrisk effekt. I dag benyttes praktisk talt bare hovedlager basert på såkalt MOS halvleder-teknikk. På de første lagerbrikkene som ble benyttet kunne det lagres

forts.

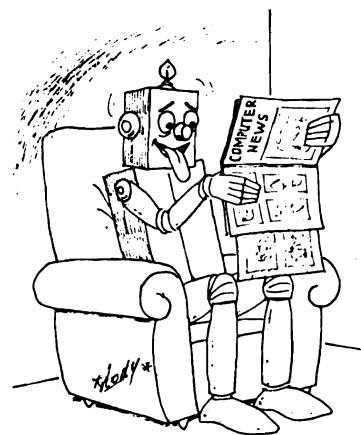
1000 (1K) informasjonsenheter (bits). I dag finnes det lagerbrikker som har en kapasitet på 64K bits, og en regner med at i 1982 vil en ha 256K-1M(10^6) bits lagerbrikker. Et lager som i 1970 besto av et kabinett med et volum på 1 m^3 , og som krevde flere kilowatt effekt, vil dermed få plass i en sigaretteske og vil kunne drives av et 5 volts batteri.

Selv med størrelsen til fremtidige hovedlager, vil det være nødvendig med andre lagerteknikker for å tilfredsstille totalbehovet for lagringskapasitet i et datamaskinsystem. I dag benyttes først og fremst platelager (disk) til å dekke behovet for lagring av store datamengder. Mens adkomst-tiden til et hovedlager ligger i området 0.1 - 1.0 mikrosekunder, ligger adkomsttiden til et platelager på ca. 10-30 millisekunder. Den enorme forskjellen i adkomsttid mellom disse to lagersystemene har alltid vært et problem for datamaskin-konstruktørene.

To nye lagerteknikker som i fremtiden vil kunne fjerne dette gapet er imidlertid nå i rask utvikling. Det er de såkalte "boblelager" og CCD ("charge coupled device") -lager. Bruken av "boblelager" er i dag stort sett begrenset til bruk i avanserte terminaler og i mikromaskinsystemer. CCD-lager er også tatt i bruk i større systemer. Det eksisterer i dag enheter basert på CCD-teknikk som erstatter konvensjonelle tromler og fasthodedisker i stormaskinsystemer.

Dagens platelagersystemer vil utvikles videre. De platepakker UNIT i dag har på sitt UNIVAC-anlegg, har en lagerkapasitet på ca. 200M tegn. Det finnes imidlertid allerede annonserte platelagersystemer hvor en kan lagre 635M tegn pr. pakke.

Siste nivå i lagerhierarkiet dekkes i dag av magnetbånd. Det er ingen begrensninger på hvor mye informasjon en på denne måten kan lagre. Problemet med bruk av magnetbånd er imidlertid at de krever manuelle operative inngrep. Det finnes i dag systemer som løser også dette problemet. Ett av disse systemene består av 9500 magnetbåndpatroner, hvor hver patron har en lagringskapasitet på 50M tegn, noe som gir en totalkapasitet på knapt $5 \cdot 10^{11}$ tegn. Adkomsttiden til en patron er på 15 sek. Slike systemer har imidlertid foreløpig en meget begrenset utbredelse.



Dataforkortelsen: OCR

Dette er en forkortelse som dukker opp stadig oftere, f.eks. snakkes det for tiden mye om OCR-utstyr. Når det dukker opp en forkortelse en er litt usikker på, vil en forsøke å sette fornuftige ord bak bokstavene. For vår egen del har vi lett for å la OCR bli til "Optical Card Reader", men det er feil. Siden vi er i ferd med å forlate hullkortalderen bør man like godt først som sist lære seg at OCR står for "Optical Character Recognition", som i praksis vil si optisk lesing av skrifttegn. F.eks. leverer IBM til sine skrivemaskiner et spesielt kulehode som gir en skrift som kan leses både av maskiner og mennesker. Dette gjør at bokstavene får et spesielt kantete datapreg, bl.a. er den Letraset skrifttype som brukes i overskriften på denne spalten inspirert av slike tegn, derfor heter også skrifttypen "Data".

Vi har også registrert at papirlappene i postkassen som avkrever oss penger etter-hvert er blitt temmelig upersonlige. Nota-bene-notiser om forbud mot endring av beløp, bretteing eller meldinger til adressat er obligatorisk på slik ubehagelig, overraskelsespost, fordi de skal leses av maskiner.

Det forskes og utvikles stadig på dette feltet, så vi må forvente at det blir stadig flere lesende maskiner rundt oss. Som omtalt i siste nummer av RUN.NYTT i forbindelse med humanistisk databehandling kan vi vente oss en liten revolusjon i informasjonsbehandling den dagen vi bare kan putte en vanlig bok i en lesemaskin og dermed få innholdet overført til datamaskinell form. I fremtiden vil vi kanskje også få maskiner som kan lese vanlig håndskrift - kanskje til og med legeresepter!

RUNIT's nye numeriske programbibliotek NAG

RUNIT vil i løpet av desember få sitt nye numeriske programbibliotek NAG. Dette vil komme i tillegg til det biblioteket RUNIT nå har.

Følgende språkversjoner er bestilt:

FORTRAN FIELD-DATA og ASCII versjon - begge i både enkel- og dobbelpresisjon.

Biblioteket består av 345 numeriske og statistiske rutiner. Det er meget godt strukturert. Det er inndelt i 27 kapitler. Disse er:

A02 - COMPLEX ARITHMETIC
 C02 - ZEROS OF POLYNOMIALS
 C05 - ROOTS OF TRANSCENDENTAL EQUATIONS
 C06 - SUMMATION OF SERIES
 D01 - QUADRATURE
 D02 - ORDINARY DIFFERENTIAL EQUATIONS
 D04 - NUMERICAL DIFFERENTIATION
 D05 - INTEGRAL EQUATIONS
 E01 - INTERPOLATION
 E02 - CURVE AND SURFACE FITTING
 E04 - MINIMIZING OR MAXIMIZING A FUNCTION
 F01 - MATRIX OPERATIONS, INCLUDING INVERSION
 F02 - EIGENVALUES AND EIGENVECTORS
 F03 - DETERMINANTS
 F04 - SIMULTANEOUS LINEAR EQUATIONS
 F05 - ORTHOGONALISATION
 G01 - SIMPLE CALCULATIONS ON STATISTICAL DATA
 G02 - CORRELATION AND REGRESSION ANALYSIS
 G04 - ANALYSIS OF VARIANCE

G05 - RANDOM NUMBER GENERATORS
 H - OPERATIONS RESEARCH
 M01 - SORTING
 P01 - ERROR TRAPPING
 S - APPROXIMATIONS OF SPECIAL FUNCTIONS
 X01 - MATHEMATICAL CONSTANTS
 X02 - MACHINE CONSTANTS
 X03 - INNERPRODUCTS

Hvert kapittel er systematisk inndelt i underkapitler etter problemtyper. Kapittel E04 er for eksempel inndelt slik:

E04A - one dimensional, function-only subroutines
 E04B - one dimensional, first derivative subroutines
 E04C - n-dimensional, function-only subroutines
 E04D - n-dimensional, first derivative subroutines
 E04E - n-dimensional, first and second derivative subroutines
 E04F - non-linear, least squares, function-only subroutines
 E04G - non-linear, least squares, first derivative subroutines
 E04U - n-dimensional, non-linear constraints, function-only subroutines
 E04V - n-dimensional, non-linear constraints, first derivative subroutines
 E04W - n-dimensional, non-linear constraints, first and second derivative subroutines

Det er tre nivå av subrutiner - som vist i følgende tabell:

Type	Purpose	Example
Problem solvers	One subroutine call to solve the problem	Solution of set of simultaneous real linear equations
Primary subroutine	Each subroutine implements a major algorithm	LU factorization
Basic module	Basic utility	Extended precision inner-product

(NAG: Numerical Algorithms Group)

NAG forts.

Det er lagt stor vekt på å gi en god dokumentasjon av biblioteket - både for hjelp til å finne den rette rutinen og for å bruke rutinen.

Hvert av kapitlene foran starter med en meget god kapittelintroduksjon. Denne gir først en generell informasjon om problemfeltet, og etterpå anbefalinger for valg og bruk av rutinene. Hver introduksjon slutter med et flytskjema som raskt kan gi beskjed om den riktige rutinen ut fra beregningsmetode og data. Et eksempel på slikt skjema vises:

rogrambeskrivelsen av de enkelte rutinene har til slutt listing av et eget programeksempel med inngangsdata og resultat.

Når biblioteket er mottatt og lagt inn i maskinen, vil nærmere beskrivelse av hvilke programmer som er i biblioteket og hvordan de vil være tilgjengelige, komme i et eget nummer av RUN-NYTT.

Knut L. Vik
RUNIT



Det er ikke uvanlig å gi sine forretningsforbindelser enkle julepresanger, og i år har RUNIT tenkt å gi brukerne en julegave! Det blir en liten lomnehåndbok med tittelen:

SLIK BRUKER DU INTERAKTIV TERMINAL

Som tittelen sier er det en innføring i hvordan en interaktiv bordterminal brukes - innloggingsprosedyre, kontrolltegn, de viktigste EDITOR-kommandoer, feilutskrifter o.l. Vi regner med at boken blir lagt ut for gratis avhenting i ekspedisjonene like før jul. Den kan også bestilles over telefon (9)3027.

Billigere kjøring i desember og januar!

RUNIT vil nå gjenta sin suksess fra sommermånedene, og tilbyr forandringer i prismekanismen for CPU-krevende jobber i tiden 1978-12-15--1979-01-21.

Denne gang gjelder det både typekode F og K, dvs. offentlig finansiert forskning, men ikke andre typer kjøring.

Følgende regler vil gjelde:

- 1) CPU-tid (pr. XQT) utover 8 min. er gratis, max tid pr. RUN er normalt 30 min., men kan vurderes av Snorre Torgnes. Det skal benyttes T-opsjon på RUN-kortet for automatisk kutting på max tid. Bruk av masselager, utskrift m.m. betales etter vanlige satser.
- 2) Jobbene blir utført på kvelds-/nattid (automatisk, slik som nå for mer enn 4 min. CPU), og bør være lagt opp slik at de kan kjøres på ubetjent anlegg.
- 3) Jobbene skal meldes til Snorre Torgnes, tlf. (9)3021, som også skal ha opplysninger om særegenheter ved jobben, som primær- og masselagerbehov og antatt klokkeid aktiv; om det er ting operatørene bør vite av hensyn til samtidig utførelse av flere slike jobber e.l.
- 4) Jobbene må vike for viktige produksjonsjobber (fakturering og regnskap), men vil normalt være ferdige "over natten".

RUNIT håper dette vil være et attraktivt tilbud, selv om kanskje ikke så mange har slike jobber ferdige nå. RUNIT vil forsøke å gjøre sommer- og julesalg til en fast ordning for ettertiden!

FORSIDE OG BAKSIDE

denne gang er tegnet av den kjente amerikanske "cartoonist" Ron Cobb. Cobb er opprinnelig en såkalt "underground" artist, men har etterhvert begynt å publisere sine arbeider i vanlige aviser. Hans skisser går ofte på temaet økologi, og han har en egen evne til å understreke farene ved at mennesket ødelegger sine omgivelser.

LURT - OG SURT MED SPSS

Reaksjonene på forrige nummers oppfordring har ikke vært overstrømmende - så vi forsøker en gang til: Brukere som har gode ideer, spørsmål eller liknende angående SPSS og mener at dette er av felles interesse for flere - skriv eller ring til

Paul Rusten, RUNIT
Kjemihallen (tlf. 2998)

og du blir kjendis i denne spalte.

DAGENS TEMA: HVORDAN FÅ NED KJØREUTGIFTENE PÅ SPSS-KJØRINGENE

1) Benytt mulighetene til å lagre data og spesifikasjoner på fil ved hjelp av SAVE FILE- og GET FILE-kommandoene. Se kap. 7 i håndboka.

2) Dersom en SPSS-fil inneholder store datamengder svarer det seg å kontrollere at et kjøreooppsett er korrekt i en egen kjøring ved bruk av EDIT-kommandoen, (se kap. 13 i håndboka).

3) Benytt "Integer modus" i rutinene FREQUENCIES, CROSSTABS og BREAKDOWN der dette er praktisk mulig. Dette gir langt raskere bearbeiding enn "General modus". I tillegg vil SPSS i "integer modus" oppdage at en kjøring som:

```
CROSSTABS INTEGER=A B(1,100)/TABLES=A BY B
ikke er mulig med mindre "workspace" utvides. Kjøringen vil bli avbrutt med en gang dersom dette ikke er ordnet. Dersom formen:
```

```
CROSSTABS TABLES=A BY B
ble benyttet ville betydelige mengder "CPU"-tid (og penger) brukes før SPSS oppdager at tabellen sprenger alle rammer.
```

4) Dersom de variable som inngår i en analyse ikke inneholder "missing values" kan "OPTION 1" benyttes og en sløser ikke bort datamaskintid med unødvendige kontroller.

5) Dersom det er mulig - erstatt IF-setninger med COMPUTE- og/eller RECODE-setninger. Disse krever mye mindre datamaskintid.

Et lite eksempel:

Anta at vi ønsker å gruppere våre individer i tre inntektsgrupper (lav, middels og høy) avhengig av verdien på variabelen INNTKT. Dette kan gjøres slik:

```
IF (INNTKT LE 50000) INTGRP=1
IF (INNTKT GT 50000 AND LE 90000)
INTGRP=2
IF (INNTKT GT 90000) INTGRP=3
```

Samme resultat kan oppnås på en langt billigere måte ved:

```
COMPUTE INTGRP=INNTKT
RECODE INTGRP (LOWEST THRU 50000=1)
(50001 THRU 90000=2)
(90001 THRU HIGHEST=3)
```

HVOR STORE BESPARELSER ER DET MULIG Å OPPNÅ?

Det er umulig å gi noe eksakt svar. Det avhenger av antall enheter, bruk av "labels", spesifikasjoner av "options" og "statistics" for den enkelte analyse, osv. Mer et eksempel som er kjørt viser at:

Ved utkjøring av en 20x300 krysstabell fra ei fil med ca. 5000 enheter ble tidsforbruket redusert med 33% når en benyttet "integer modus" og "OPTIONS 1" i CROSSTABS i forhold til "general modus" uten "OPTIONS 1".

NY SPSS-VERSJON

I det disse linjer leses er versjon 7.0 (eller fullstendig: versjon 7.0-UW2.2) tilgjengelig ved RUNIT. Den kjøres ved å bruke styrekortet

```
@SPSS*SPSS.SPSS7
```

Det er en fullstendig revidert versjon i forhold til tidligere versjoner, og flere endringer/tillegg er kommet:

1) Det er ikke lenger mulig å benytte '.' istedet for hopp til kolonne 16 i styrekortene.

2) Nye tester:

- MULT RESPONSE for tabulering av variable med mer enn ett svaralternativ.

- NPAR TESTS et sett tester for ikkeparametrisk statistikk. Inneholder det meste av det som finnes i NPAR.

- RELIABILITY for beregning av reliabilitetskoeffisienter.

En "UPDATE-MANUAL" på ca. 130 sider pluss div. annen informasjon om SPSS-versjon 7.0 fås ved henvendelse til:

Paul Rusten, RUNIT
Kjemihallen, tlf. 2998

ORAKEL~ TJENESTEN

RUNITs orakeltjeneste holder til i Sentralbygg II, 2. etg. "Bortgjemt" på et kontor sitter nemlig 4 studenter og gir orakelsvar.

Ora'kel, et, lat., (i oldtiden) guddomsvar; dunkelt og tvetydig svar som prestene på gudenes vegne meddelte menneskene; sted el. institusjon hvor slike svar blev gitt (f.eks. i Delfi); gåtefullt el. tvetydig svar el. råd; dyspsindig svar; klok og ansett rådgiver.

Vi som er orakler vil vel helst bli kjent for å tilhøre den siste gruppen, men det er vel neppe til å komme forbi at noen mener at vi gir dunkle og tvetydige svar.

Det som kreves for å bli ORAKEL er gode FORTRAN og EXEC 8 kunnskaper. Imidlertid har oraklene som pasjonerte datamaskinbrukere, rotet seg borti litt av hvert av problemer og problemløsninger rundt i miljøet. Kunnskapsmengden er derfor atskillig mer omfattende enn kravene (heldigvis) og vi kan derfor også løse ditt problem (SIC!).

Tilsammen mestrer vi de fleste sprog som benyttes i miljøet (f.eks. ALGOL, COBOL, SIMULA og FORTRAN) i tillegg til terminalbruk, editor, programbiblioteket, database-systemer og mye mer. Nå er det imidlertid galt å trekke den slutning at oraklene kan og vet alt. Det har vel mange etterhvert erfart. Til hjelp har vi derfor en hærskare av oppslagsverk og manualer, og en av de viktigste oppgavene våre er å kunne formidle kontakt mellom frustrerte brukere og veiledere som er eksperter på brukernes problemer.

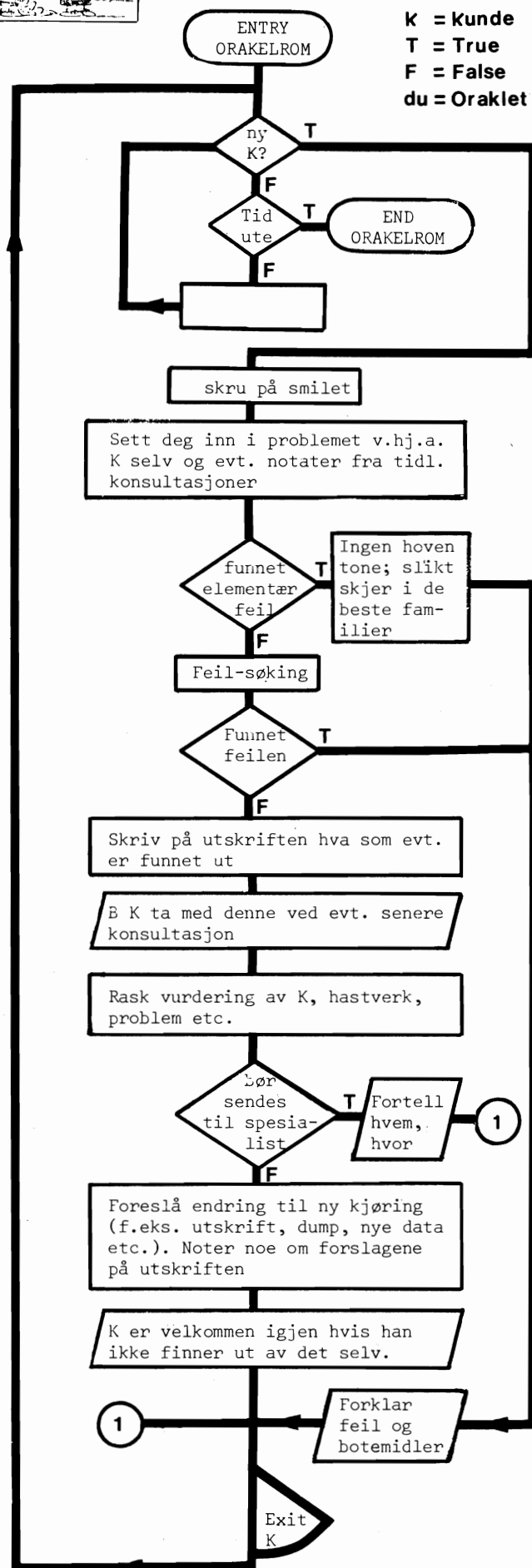
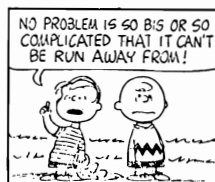
Oraklene er glade over å få besøk (se flytskjemaet), men vi foretrekker at problemet først er grundig gjennomtenkt.

MOTTO: Problemer løses mens de venter, mirakler tar litt lenger tid.

HVOR: SB-II, 2.etg. (ORAKEL-kontoret)

NÅR: Mandag - fredag kl. 11-15

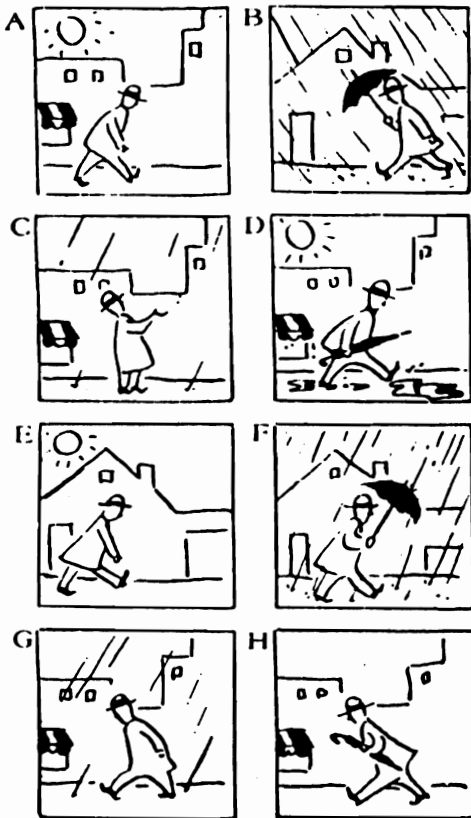
HVORDAN: Ved personlig fremmøte medbringende "listing" og/eller problem-spesifisering, eller over telefon 3004.



JULENØTTER

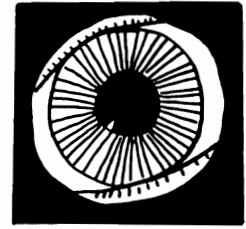
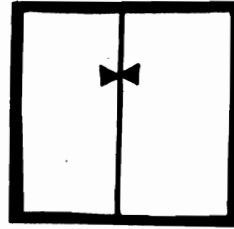
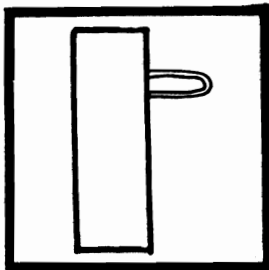
SEKVENSTEST

Studer serien av bilder nedenfor. Start med bilde E, og forsøk å arranger bildene slik at de representerer en logisk sekvens av hendelser. Klarer du det, tyder det på at du er flink til å kombinere observasjonsevne med evne til å resonnerer.

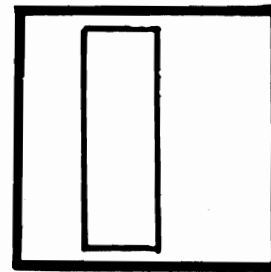


DRODLER

Drodlesporten går ut på å finne ut hva noen svært stiliserte tegninger forestiller. Nedenfor er vist noen eksempler, svar står på side 11.



Den første drodlen overfor kan lett modifiseres til en "data-drodle":



Har du ideer til flere data-drodler? Send dem til RUN NYTT!

ER DET SÅ KJEDELIG ?

Nedenstående klipp fra boken "Book of lists" skulle tyde på at programmerer-yrket ikke gir de helt store opplevelser. Professoryrket derimot byr på mer fart og spenning!

23 TYPICAL JOBS RATED ACCORDING TO BOREDOM

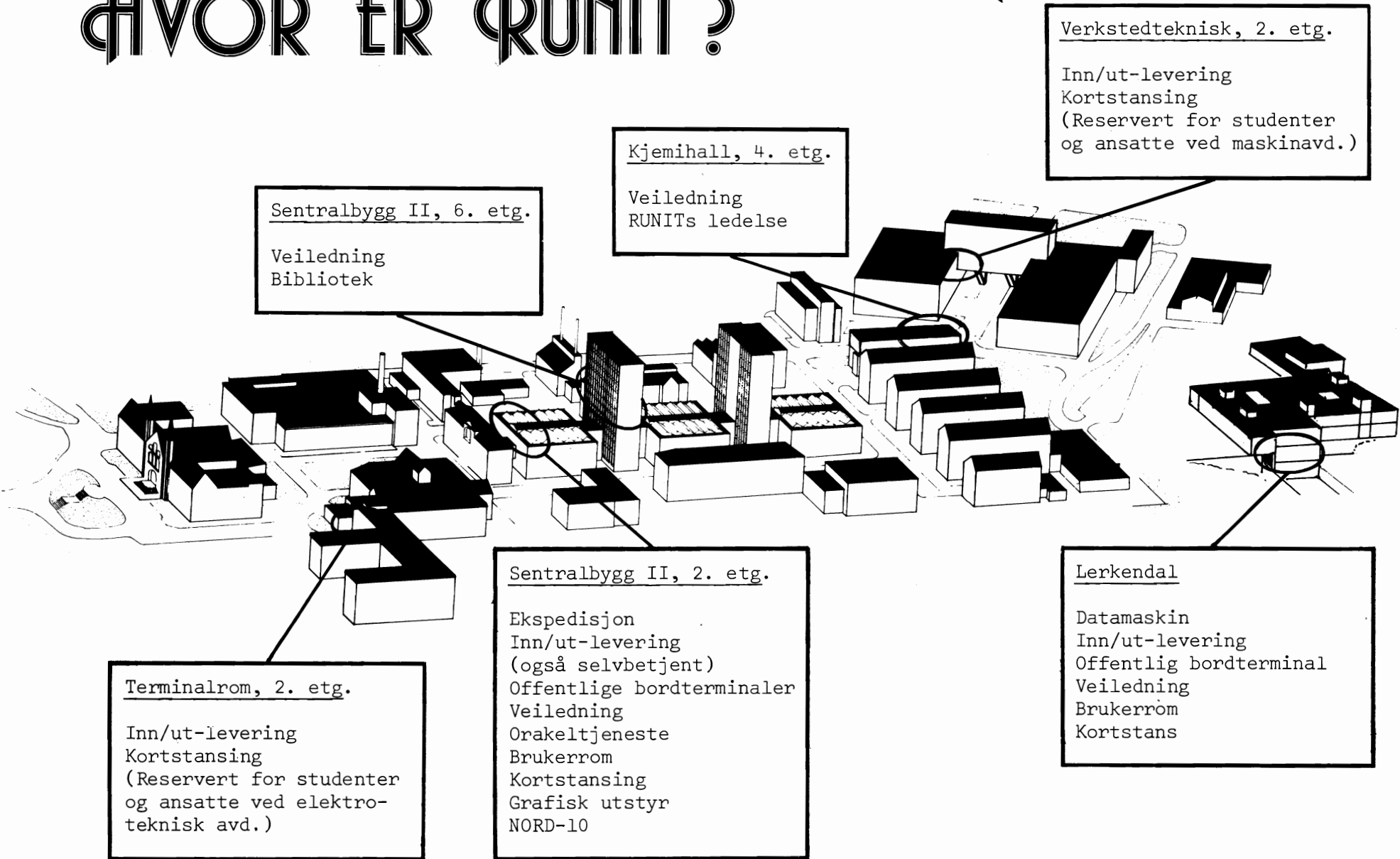
Based on interviews with 2,010 workers performing 23 different jobs, the Institute for Social Research at the University of Michigan drew up "boredom factors" for each occupation. The average was considered to be 100, and the higher the rating the more boring the job.

	Boredom Rating
1. Assembler (work paced by machine)	207
2. Relief worker on assembly line	175
3. Forklift-truck driver	170
4. Machine tender	169
5. Assembler (working at own pace)	160
6. Monitor of continuous flow goods	122
7. Accountant	107
8. Engineer	100
9. Tool- and diemaker	96
→ 9. Computer programmer	96
11. Electronic technician	87
12. Delivery service courier	86
13. Blue-collar supervisor	85
14. White-collar supervisor	72
15. Scientist	66
15. Administrator	66
17. Train dispatcher	64
18. Policeman	63
19. Air traffic controller (large airport)	59
20. Air traffic controller (small airport)	52
21. Professor with administrative duties	51
22. Professor	49
23. Physician	48

SPACHMO-KRYSSSET

Han hjelper til	Merke over- gang		Pron.	1 Kebne- Kajise	Eng.ut- rop Harald Tusberg	Hele	Flytbar Svar betalt	Skrute
Tenk								
Organ			-			Alike- vel (dial.)	Tone SKredk	
Krigs- område		2 x	Joker		Språk			
				Liv Rund- berget	Flaske Sjef			
H.stad Omtrent	Reining		H. Plagg Vokaler					Utrop
							Gud Fugl	
				Signal		Part- nere	Redsk. fabr. Tilgang	
Smådyr Finn reiningen	Lever Sjarm		H.stad Fugl		Nasj. merke			
			Tudele ELV					
Kulle		Bønn			Gmlt. parti			By i Italia Pap mange- rinne
	Uro Batch	Spør Bare			Like 1b. pøtt. (forn.)		Kraft forster- ker Pattedyr	
Tau opp Seisk.			Medar- beider Kode					
						Stropp Dantam.		
stat opp- noid				Toveis Hast				
H.stad						Ubun- det		Line
Plast fabrikk						Hotell Kjede		
Uttale leppe korn			Like		16. vin stat			
		Språk		1 ceci- lie Org. Like			Huske	Behand- ling
	Neste		Tegne redsk. prep.					Ledd
						Institu- sjon Fase		
Land M.navn	Brutal Fisk		Mynt Sluttord			Gass		
				Slafs?		Forskjel- lige Stoff		
Kommun- ikasjons- enhet						= ø Uro		Høyde
			Grunn- stoff		ytte		strø	
Pike navn			Under- lag					

HVOR ER RUNIT ?



QUNIT'S VEILEDNINGSTJENESTE

Generell veiledning:

- Oraklet.: Rom 255/3, bak terminalrom i Sentralbygg II, 2.etg. kl. 11.15 - 15.15 mandag - fredag, tlf. 3004 (bare i semestret)
- Ekspedisjonen: 2.etg. Sentralbygg II, kl. 08.00 - 16.00, tlf. 3026. Her tegner du deg som bruker. Kjørenr., brukernr. og skriftlig informasjon utleveres. Henvisning til rette vedkommende.

Ansvarlig for brukertjenesten: Arne Asphjell, Kjemihall, tlf. 2997

Spesielle veiledere:

OMRÅDE	VEILEDER	PLASSERING	TLF.
Bruk av bordterminal	Knut Ragnar Holm	Kjemihall	2998
Databaser, filsystemer	Steinar H. Kvitsand	SB-II, 6.etg.	2969
Datanett (lokalt og eksternt)	Alf Engdal	SB-II, 6.etg.	2979
Grafisk databehandling	Morten Zachrisen	Kjemihall	2996
Håndbøker (UNIVAC og NORD)	Per Balstad	Kjemihall	2992
Katalogiserte filer og magnetbånd	Steinar Reimers	Lerkendal	3022
Kjøp av mindre dataanlegg	Trond Børsting	SB-II, 6.etg.	2972
Kjøp av terminalutstyr	Trond Børsting	SB-II, 6.etg.	2972
	Bjørn Myrstad	SB-II, 2.etg.	2974
	Hans G. Endresen	SB-II, 2.etg.	2974
Kjøring på andre dataanlegg	Snorre Torgnes	Lerkendal	3021
Konvertering av magnetbånd fra andre anlegg	Per Balstad	Kjemihall	2992
Konvertering av program IBM UNIVAC			
CD UNIVAC			
Honeywell-Bull UNIVAC	Trond Johansen	SB-II, 6.etg.	2962
NORD-10	Geir Moe	SB-II, 6.etg.	2991
Priser, avregning	Bjørn Gifstad	SB-II, 6.etg.	2966
Programbibliotek	Per Balstad	Kjemihall	2992
Programmeringsspråk ALGOL	Tor Stålhane	Kjemihall	3014
	Kjell Sundnes	Kjemihall	3005
	Harald Oftedal	SB-II, 7.etg.	3446
APL	Per Balstad	Kjemihall	2992
BASIC	Greta Aas	SB-II, 5.etg.	2954
	Arild Laugen	SB-II, 6.etg.	2970
COBOL	Geir Skylstad	SB-II, 6.etg.	2986
ED	Trond Johansen	SB-II, 6.etg.	2962
FORTRAN	Geir Skylstad	SB-II, 6.etg.	2986
	Rune Østlyng	SB-II, 6.etg.	2961
	Carsten Myhre	SB-II, 6.etg.	2957
	Arve Dispen	Lerkendal	2989
	Knut Ragnar Holm	Kjemihall	2998
	Ole Solberg	Kjemihall	3017
MARY	Geir Green	Kjemihall	3013
	Paul Rusten	Kjemihall	2998
SIMULA	Per D. Øvrebø	Lerkendal	2975
Andre	Snorre Torgnes	Lerkendal	3021
Reklamasjoner	Steinar Reimers	Lerkendal	3022
	Ragnhild Heggglund	Eksp. SB-II	3027
SPSS	Paul Rusten	Kjemihall	2998
	Per Balstad	Kjemihall	2992
Statistikkprogram	Paul Rusten	Kjemihall	2998
Styrespråk, EXEC 8	Per D. Øvrebø	Lerkendal	2975
	Bjørn Berg	Lerkendal	2975
	Bjørn Ranum	SB-II, 6.etg.	2980
	Knut Ragnar Holm	Kjemihall	2998
Tildeling av brukernr.,kjørenr.	Ragnhild Heggglund	Eksp.SB-II	3027
Tildeling av magnetbånd	Steinar Reimers	Lerkendal	3022
Tilknytning av terminalutstyr	Snorre Torgnes	Lerkendal	3021
Vedlikehold av Kongsberg-utstyr	Arvid Grande	SB-II, 2.etg.	2973
	Ola Staveli	SB-II, 2.etg.	2973
Vedlikehold av utstyr fra Norsk Data	Erik Hansen	SB-II, 2.etg.	3036

Veiledere i andre miljøer:

Dragvoll:	Eirik Lien	Bygning 3, nivå 5	6718
Lade:	Ola Kindseth	Blokk C, 3. etg.	tlf. 16900/282
Rosenborg:	Kolbjørn Hagen	Paviljong B	6223
Museet: Veiledningstjeneste inntil 1 time pr. uke	Arild Laugen	SB-II, 6.etg.	2970

EKSTRA BONUS!

RUN-NYTT mini-poster:

