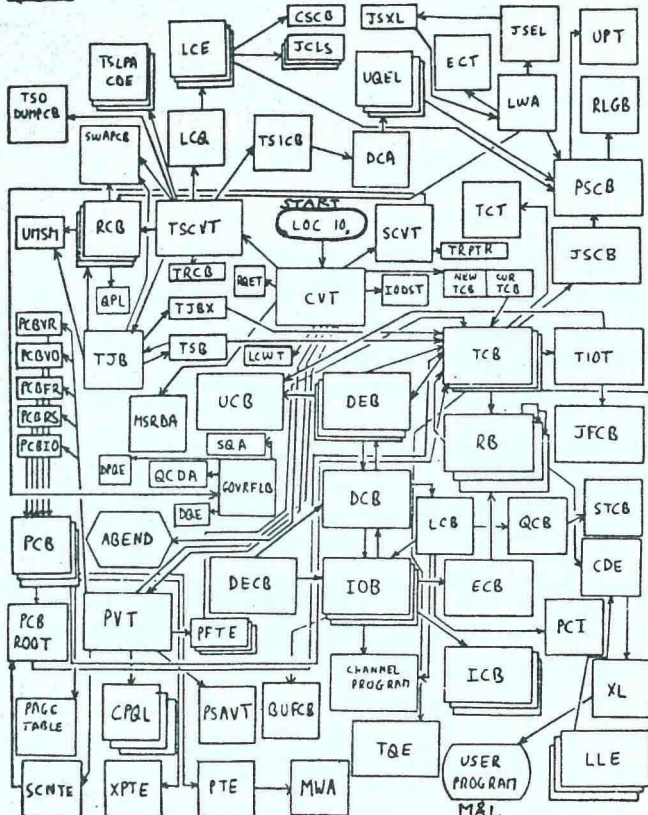


1-1984

QUICK COURSE IN OS/VS CONTROLBLOCKS

Behövet av djupare insikt i relationerna och kommunikationen mellan olika kontrollblock i IBM:s VS-system har länge varit ett känt faktum. För att råda bot mot denna brist har BRA i samarbete med Boole & Babbage tagit fram en snabbkurs i ämnet som vi härmed bjuder på. Målsättningen med kursen är att eleven efter någon timmes studie självständigt skall kunna beskriva samtliga kontrollblock och deras inbördes förhållande till varandra. Lämplig start är LOC 10_w



Datorföreningen STACKEN

Föreningen, som bildades 1978, är en kårförening, med runt 150 medlemmar. Föreningens målsättning är att genom diverse aktiviteter ge medlemmarna möjlighet öka sina kunskaper och sitt intresse inom dataområdet.

Vad aktiviteterna beträffar, är de beroende av vad medlemmarna engagerar sig i. Vi har haft studiebesök hos olika leverantörer och några installationer. Vi har, begränsad, tillgång till NADAs DEC-2020, Nadja. Där vi kan lära oss programmera i ett flertal programmeringsspråk. Vi kan även lära oss texthantering. Dessutom har vi tillgång till konferenssystemet KOM, utvecklat vid FOA. För en tid sedan fick vi en avancerad dator av DEC. Vår stora aktivitet är nu att finna en lokal till datorn och få igång den. Vi har även föreläsare, som vill dela med sig av sina erfarenheter.

Medlemmarna samlas klockan 19, första helgfria torsdagen i varje månad, i sal E7 på KTH.

Vi vill gärna ha fler medlemmar och om du är intresserad - kom på ett möte.

STACKPOINTER

STACKPOINTER är organ för Datorföreningen STACKEN. STACKPOINTER utkommer när material i tillräcklig mängd finns, förhoppningsvis 4-5 gånger per år.

Ansvarig utgivare: Jan Michael Rynning
Redaktör: Hans Nordström
I redaktionen: Mats O Jansson
Danny Kohn
Jan Michael Rynning
Leif Eurén
Göran Wallberg

Datorföreningen STACKEN
c/o NADA KTH
100 44 STOCKHOLM

Postgiro 433 01 15 - 9

Medlemskap i STACKEN kan beviljas efter ansökan till föreningens styrelse. Medlemskapet kostar f n SEK 84 per år och inbetalas på STACKENS postgirokonto.

Återgivande av delar av innehållet är tillåtet om källan anges.

Innehåll

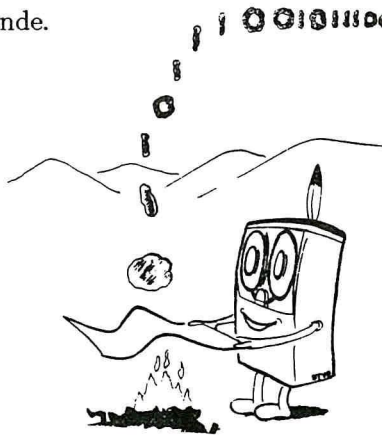
Artikel	sida
Framsidan	1
Datorföreningen STACKEN	2
STACKPOINTER	2
Innehåll	3
Vårmöte	4
Motioner	5
Magic!?.	6
Recension av "Samkörarna"	7
Hur du får ett konto på Nadja	8
Hur du loggar in på Nadja	10
Hur du hanterar filer på Nadja	13
Hur du kör program på Nadja	18
KATIA	19
PL/1, vad är det?	21
Hur Murphy's lag kom till	23
En titt på TV-spelen i TV-spelens förlovade land	24
Fadderverksamheten	25
E de någe å ha?	26
Real Programmers Don't Write Specs	27
Baksidan	28

Vårmöte

Förslag till föredragningslista för Datorföreningen STACKENS vårmöte, torsdagen 1984-02-02 klockan 19 på KTH, E7.

- §1. Mötets öppnande.
- §2. Val av justeringsmän.
- §3. Val av ordförande för mötet.
- §4. Val av sekreterare för mötet.
- §5. Tillkännagivande av uppgjord röstlängd.
- §6. Frågan om mötet var stadgeenligt utlyst.
- §7. Frågan om dagordningens godkännande.
- §8. Verksamhetsberättelse.
- §9. Revisionsberättelse.
- §10. Balansräkning.
- §11. Ansvarsfrihet för avgående styrelse.
- §12. Presentation av budget för 1984.
- §13. Motioner.
- §14. Övriga frågor.
- §15. Mötets avslutande.

/gw



Motioner

Motion angående STACKENS medlemskap i Unga Forskare.

Eftersom STACKEN inte har någon nytta av medlemskapet i Unga Forskare, så yrkar jag på att STACKEN utträder ur Unga Forskare. Ytterligare en anledning är att jag ogillar kollektivt medlemskap i en förening.

Upsala 1983-12-02 01:12:53

Per Danielsson

Motion angående hedersmedlemskap åt Alexander.

Jag tycker att den framlidne mögelsvampen Alexander (må han vila i frid) skall återfå sitt hedersmedlemskap i STACKEN som han så skändligen fråntogs för en tid sedan.

Upsala 1983-12-02 01:13:12

Per Danielsson

Motion angående hedersmedlemskap för Nils Söderman.

STACKEN hade säkerligen inte blivit till om inte Nisse Söderman haft de idéer han hade. Som ett litet bevis på uppskattning för NS initiativ så tycker jag att vi gör NS till hedersmedlem.

Plats 1984-01-08 18:12:44

Hans Nordström

Motion angående medlemsfördelningen.

Ont om kvinnliga medlemmar i STACKEN, är det gott om. En av orsakerna kan vara någon slags tröskeffekt. Är det få kvinnliga medlemmar förut, så krävs det mycket för att komma. Skall vi inte försöka göra något åt det?

Mitt förslag är att det skall vara kostnadsfritt, för damer, att ansluta sig 1984. I förhoppning att många vågar testa, när det ekonomiskt inte gör något.

Plats 1984-01-08 18:18:53

Hans Nordström

Magic !?

“ Some years ago I was snooping around in the cabinets that housed the MIT AI lab’s PDP-10, and I noticed a little switch glued to the frame of one cabinet. It was obviously a homebrew job added by one of the lab’s hardware hackers (no one knows who).

You don’t touch an unknown switch on a computer without knowing what it does, because you might *crash* it. The switch was labeled in a most unhelpful way. It had two positions, and scrawled in pencil on the metal switch body were the words “magic” and “more magic”. The switch was in the “more magic” position.

I called another hacker over to look at it. He had never seen the switch before, either. Closer examination revealed that the switch only had one wire running to it! The other end of the wire did disappear into the maze of wires inside the computer, but it’s a basic fact of electricity that a switch can’t do anything unless there are two wires connected to it. This switch had a wire connected on one side and no wire on it’s other side.

It was clear that this switch was someone’s idea of a silly joke. Convinced by our reasoning that the switch was inoperative, we flipped it. The computer instantly crashed!

Imagine our utter astonishment. We wrote it off as coincidence, but nevertheless restored the switch to the “more magic” position before reviving the computer.

A year later, I told this story to yet another hacker – David Moon, as I recall. He clearly doubted my sanity, or suspected me of a supernatural belief in the power of this switch, or perhaps thought I was fooling him with a *bogus saga*. To prove it to him, I showed him the very switch, still glued to the cabinet frame with only one wire connected to it. It was still in the “more magic” position. We scrutinized the switch and its lone connection and found that the other end of the wire, though connected to the computer wiring, was connected to a ground pin. That clearly made the switch doubly useless: not only was it electrically nonoperative, but it was connected to a place that couldn’t affect anything anyway. So we flipped the switch.

The computer promptly crashed.

This time we ran for Richard Greenblatt, a long-time MIT hacker who was close at hand. He had never noticed the switch before, either. He inspected it, concluded it was useless, got some diagonal cutters and *diked it out*. We then revived the computer, and it has run fine ever since.

We still don’t know how the switch crashed the machine. There is a theory that some circuit near the ground pin was *marginal*, and flipping

the switch changed the electrical capacitance enough to upset the circuit as millionth-of-a-second pulses went through it. But we'll never know for sure. All we can really say is that the switch was magic.

I still have the switch in my basement. Maybe I'm silly, but I usually keep it set on "more magic". "

Det ser intressant ut? Stycket är hämtat ur en bok som kan vara trevlig att läsa. (Tack, PL.) Det är filen GAM:HACKER.JAR i tryck, med mycket tillägg och lite borttag. Boken har fått titeln **The Hacker's Dictionary** — **A guide to the World of Computer Wizards** på förlaget Harper & Row, New York. Författare finns det flera, men den som har © är Guy L. Steele Jr. Ett välkänt namn, bland hackers. Boken kostar USD 5.95 och är registrerad som ISBN 0-06-091082-8. Någon intresserad för ett co-köp?

/hn

Recension av "Samkörarna"

Boken är en s.k. "datathriller" och är bitvis riktigt rolig. Ändå tas några rätt allvarliga punkter om dagens dator-samhälle upp. Den varnar för vad som kan hända om myndigheter eller andra får samköra dataregister hur som helst. Något annat som tas upp, är vad vi riskerar i händelse av krig; vilka faror dagens dator-samhälle innebär för oss medborgare, ifall vi skulle bli invaderade.

Samkörarna innehåller även ett exempel på hur fredsaktivister skulle kunna använda KOM för att nå sina syften och vilka konsekvenser det skulle få för politikerna. Boken är en bra tankeställare för oss, som i framtiden (och idag) kommer att vara beroende av datorer.

Samkörarna av Jan-Jörgen Stenhagen
Wahlström & Widstrand
ISBN 91-46-14425-0

MOJ

Hur du får ett konto på Nadja

Nadja är en DEC-2020 minidator, som används av NADA (Institutionen för Numerisk Analys och Datalogi på Kungliga Tekniska Högskolan) för undervisning, forskning och administration.

Nadja står i QZs (Stockholms Datorcentrals) lokaler längst ner i huset med klocktornet. Det är QZs operatörer och systemprogrammerare som ansvarar för driften av datorn.

Nadja har 512K 36-bitars ord primärminne, 2 skivminnen om vardera 38M 36-bitars ord, 1 bandstation, som klarar 800 bpi och 1600 bpi, 48 asynkrona terminallinjer, som klarar hastigheter mellan 50 baud och 9600 baud, samt en synkronlinje för hopkoppling med övriga datorer i ANF-10-nätet.

Eftersom Nadjas centralenhet (KS-10) fungerar på samma sätt som den på Oden (KL-10), Katia (KA-10) och andra DEC-10-datorer och eftersom de även har samma operativsystem (TOPS-10) så kan samma program köras på alla sådana datorer.

På Nadja finns bland annat programspråken ALGOL 68, APL, BASIC, BLISS 10, C, COBOL 68, COBOL 74, FOCAL, FORTH, FORTRAN, LISP, MACRO 10, MIDAS, PASCAL, PROLOG, SAIL, SIMULA 67, SNOBOL och TECO, korsassemblers för PDP-11, 1802, 6502, 6800, 6809, 8008, 8080, 8085, Z80 och 9900 samt ett flertal editorer och andra användbara program.

Du som är medlem i STACKEN kan få ett personligt konto på Nadja, som du får använda för studier och personlig utveckling. Du får alltså inte låna ut kontot ut till andra eller använda det för komersiell verksamhet eller till att störa övrig verksamhet på Nadja. Om du vill ha ett kontonummer att köra på ska du vända dig till någon i STACKENS styrelse.

Nadja är i första hand avsedd för NADAs undervisning. Därför är STACKENS konton avstängda mellan klockan 8 och 20 på vardagar under terminerna, när det normalt pågår undervisning. Även efter klockan 20 kan det hända att Nadja är hårt belastad eller att terminalsalarna är bokade för övningar. Då får du inte sitta där. Vilka tider terminalsalarna är bokade kan du få reda på genom att läsa anslaget utanför handledarbåset vid terminalsalarna eller bredvid den torrlagda fontänen på bottenvåningen eller genom att ge kommandot

.HELP TIDER

Särskilt under höstterminen brukar Nadja vara ganska hårt belastad även sent på kvällen och på helgerna. Om du märker att det är fullt i

terminalsallarna eller att datorn är hårt belastad bör du vänta med att köra. Även när datorn inte är så hårt belastad bör du undvika att köra program, som använder mycket minne eller mycket CPU-tid. KOM är tyvärr ett sådant program, som lastar ner datorn märkbart. För att se hur många som kör på Nadja kan du ge kommandot

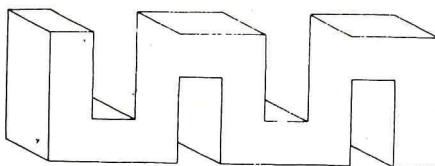
.SYSTAT Z

Om du får ut mer än 15 stycken så är datorn **hårt** belastad och då **ska** du inte köra.

Tänk på att **eleverna kommer i första hand** och att det är de som har **rätt att använda datorn!** Vi i STACKEN får hålla tillgodo med den datortid som blir över, vilket inte är så illa det heller.

För att återgälda att STACKEN får gratis datortid av NADA och för att skaffa oss ett gott rykte på KTH bör vi föregå med gott exempel i terminalsallarna genom att inte störa andra genom att föra oväsen, inte skräpa ner, utan isället städa om någon annan har skräpat ner, och hjälpa de elever som sitter där om de frågar om någonting.

Jan Michael Rynning



Redaktörn

Nu syns det att STACKPOINTER kommit igång igen. Men det kan bara fortsätta om det kommer in material. 'Aha', tänker du. 'Jag förstod att det fanns en hake gömd.' Ja, det stämmer. STACKPOINTER är inte någon Perpetum Mobile. Det behövs tillförsel av energi. I detta fallet din energi för att plita ihop en artikel om något som intresserar dig. Bara genom denna ständiga tillförsel kan vi hålla STACKPOINTER gående. GMEI.

I denna utgåva finner du lite om de nya tv-spelen. Några recensioner av intressant litteratur. Varför inte läsa lite om olika textbehandlingsverktyg.

Attans otur att inte BD hann med fortsättningen av Lisp-sticket.

/hn

Hur du loggar in på Nadja

På de flesta datorer, som mer än en person kan köra på samtidigt, måste man uppge ett kontonummer eller kontonamn och ett hemligt lösenord, innan man får börja köra. Kontonumret berättar för datorn vem man är och det hemliga lösenordet, som är olika för olika kontonummer, är till för att ingen obehörig ska kunna ta sig in på datorn genom att hitta på ett kontonummer.

På Nadja består kontonumren av två tal med komma mellan. Lösenorden ska vara 5 eller 6 tecken långa och får innehålla bokstäver (dock inte ÅÄÖ) och siffror. Ditt kontonummer och lösenord står på lappen du får när du får ut ett kontonummer att köra på.

Terminalsalar

För att kunna köra på Nadja måste du ha en terminal, som är kopplad till Nadja. Högst upp i huset med klocktornet på KTH finns det sex terminalsalar. I två av dem, kallade bruna och orange salen, finns det terminaler, som du kan använda för att köra på Nadja. När du slår på terminalen och trycker på **(RETURN)** så kopplas du antingen direkt till Nadja eller till NADAs terminalväxel KNUT. Följ anvisningarna från KNUT, så kopplas du till Nadja!

Välkommen till NADAs terminalväxel KNUT.

För hjälp, skriv ? följt av vagnretur.

NADA-KNUT>?

Dina alternativ är:

1 Termövn. Nadja 2 Nadja fria linj

Skriv alternativets nummer eller namn följt av vagnretur.

NADA-KNUT>1

Det finns ingen ledig linje till Termövn. Nadja

NADA-KNUT>2

Du är kopplad till Nadja fria linj

Modem

Om du har ett 1200/75-110 bauds modem kan du ringa till QZs UPNOD och därifrån koppla dig till Oden och sedan vidare till Nadja. Telefonnummer

är 22 81 30. När du har ringt till UPNODen trycker du på **(RETURN)** tills UPNODen svarar med annat än frågetecken. När du har gett kommandot för att koppla dig till Oden trycker du på **(RETURN)** ett par gånger.

< QZ-3/L89 Terminalväxel >

<Dator ? (? ger hjälp) > Q,D

Om du har ett 110-300 bauds modem kan du ringa direkt till Oden, genom att slå 23 86 60. Har du ett 1200 bauds modem är numret 14 35 00 eller 24 61 20. Efter du har ringt upp måste du trycka på **(RETURN)** en eller ett par gånger. När du har fått en punkt i vänstermarginalen säger du åt datorn att koppla dig vidare till Nadja.

CONNECTING TO HOST SYSTEM.

Oden/QZ Stockholm 7.01A 15:17:00 TTY25, line BALDER_77

Please LOGIN or KOM

.SET HOST NADJA

Det är QZ, alltså indirekt QZs betalande kunder, som betalar för telefonerna och modemerna, som du ringer till. För att de ska ha tillgång till alla linjer de behöver är de flesta konton på Nadja spärrade, så det inte går att logga in på dem om man tagit sig till Nadja via QZs modem på dagtid på vardagar. Eftersom STACKENS konton brukar vara spärrade på de tiderna är det normalt inget problem.

Själva inloggningen

När du har kopplat dig till Nadja, på något sätt, och ska logga in så skriver du LOGIN och ditt kontonummer, som står på lappen, och trycker på **(RETURN)**. När datorn frågar efter Password: skriver du in ditt hemliga lösenord och trycker på **(RETURN)**. Lösenordet syns inte på skärmen.

NADJA KTH DEC-2020 7.01A 15:18:59 TTY17, line NADJA_17

Please LOGIN

.LOGIN 105,2001

JOB 25 NADJA KTH DEC-2020 7.01A TTY17

Password:

Saturday 84-01-07 15:19



Hur du loggar ut

När du har kört färdigt måste du säga åt datorn att du inte ska köra längre. På Nadja gör du det med kommandot

.LOGOUT

Om du ringde upp datorn med modem är det inte säkert att telefonlinjen kopplas ner. Kolla för säkerhets skull och koppla ner telefonlinjen själv om inte datorn gjorde det. Annars kan du få höga teleräkningar!

Om förbindelsen bryts

Om förbindelsen mellan din terminal och Nadja bryts, genom att växeln kopplar bort dig eller telefonlinjen bryts, så har du 20 minuter på dig att ta dig fram till datorn och försöka logga in igen. När du loggar in på nytt får du veta att du redan är inloggad och en fråga om du vill fortsätta med det jobbet.

```
NADJA KTH DEC-2020 7.01A 15:18:59 TTY17, line NADJA_17
Please LOGIN
```

```
.LOGIN 105,2001
```

```
JOB 25 NADJA KTH DEC-2020 7.01A TTY17
```

```
Password:
```

```
Other jobs detached with same PPN:
```

```
Job 26 KOM stopped
```

```
Do you want to ATTACH to this job? (Y)
```

```
.ATTACH 25A105,2001A
```

Om du väntar mer än 20 minuter så loggas ditt jobb ut automatiskt, om du inte körde AMIS när förbindelsen bröts, för då brukar det inte loggas ut.

Om datorn stannar

Om det blir strömavbrott eller något fel i datorn, så att systemet kraschar, så är alla jobb utloggade, när datorn startas igen.

Ibland måste Nadja stängas av för serviceåtgärder. Då brukar man få ett meddelande om detta när man loggar in och med jämna mellanrum tills datorn stängs av

```
::OPR 16:31 OPSE: Timesharing ends in 30 min.
```

Efter man har fått det sista sådana meddelandet loggas alla jobb ut.

```
::OPR 17:01 OPSE: Timesharing is over.
```

Jan Michael Rynning

Hur du hanterar filer på Nadja

För att kunna bearbeta någon form av data måste du ha den i primärminnet, tillsammans med programmet som ska utföra bearbetningen. När du ska spara ett program, eller någon annan form av data, mer permanent måste du göra det på sekundärminne på de flesta datorer. Oftast använder man skivminne, eftersom det är enklast. Varje sådan mängd av undansparad information brukar kallas en fil. På nästan alla datorer kan man sätta namn på filer, så man själv slipper hålla reda på var på skivminnet den lagras och bara behöver komma ihåg namnet. Namnen på alla filer och information om var de lagras brukar finnas i en katalog på datorn, eller directory som det heter på engelska.

På Nadja består namnet på en fil av två delar. Gemensamt för delarna är att de bara får innehålla bokstäver (inte ÅÄÖ) och siffror. Den första delen, som är det egentliga namnet på filen, får innehålla 1 till 6 tecken. Den andra delen brukar användas för att ange vad filen innehåller och får vara 0 till 3 tecken lång. Den kallas namntillägg, eller extension på engelska. Mellan de två delarna har man en punkt. En fil kan exempelvis heta `ALLAN.PAS`, där `PAS` anger att den innehåller källkoden till ett PASCAL-program. I vissa kommandon kan du skriva en stjärna i stället för den första eller andra delen, för att ange att du menar alla filer med vilket namn eller vilken extension som helst, `*.PAS` till exempel. Om du vill veta vilka extensions man brukar använda och till vad kan du ge kommandot

`.HELP EXTENT`

Om du vill komma åt en fil, som inte ligger på ditt eget directory, måste du specificera på vilket directory den ligger, exempelvis `ALLAN.PASÄ105,2001A` eller `DSKC:FOOBAR.CÄ12,6666,CÄ`. Det som står mellan Ä och Ä är directorynamnet, som består av kontonumret, följt av namnen på eventuella underdirectories. Det som står före kolonet är namnet på den skivminnesenhet filen finns på. Nadja har två sådana, `DSKB:` och `DSKC:`.

Hur du skriver in en fil

För att knappa in en fil i datorn använder du ett speciellt program, en texteditor. Om du sedan vill ändra på någonting i filen, lägga till eller ta bort någonting, så använder du texteditorn till det också. En editor är även användbar om man vill leta reda på någonting i en textfil.

Det finns två editorer på Nadja, som vi kan rekommendera. Den ena är AMIS, en skärmorienterad editor med mycket kraftfulla kommandon,

skriven av medlemmar i STACKEN med en annan editor, EMACS, som förebild. AMIS visar hela tiden texten du håller på att editera på skärmen. För att den ska fungera måste datorn veta vilken typ av bildskärm du har. För att få veta vilka typer av bildskärmar datorn känner till skriver du

.TERMINAL TYPE ?

AMIS klarar inte av alla typer av terminaler, som datorn då räknar upp, fast den klarar de vanligaste. Om du inte förstår vad förkortningarna betyder kan du ge kommandot

.HELP TERMINAL

Då får du ut en lång lista med de fullständiga namnen på en massa terminaler och vilka förkortningar som används för dem. När du har hittat din terminaltyp kan du trycka två gånger på **(CONTROL-C)** för att slippa se resten av listan. Om du sitter i NADAs terminalsalar behöver du normalt inte bekymra dig. När du kopplar upp dig mot datorn antar den att du sitter vid en Datamedia Elite 1520 (EL1520). Om du skulle råka sitta vid en Datamedia DT80/1, som fungerar som en VT100, så får du ge kommandot

.TERMINAL TYPE VT100

Det finns ett särskilt program för att lära sig AMIS, som startas med

.R PUB:ATEACH

Om du ska knappa in en ny fil eller ändra i en fil du redan har, med AMIS, exempelvis filen **ALLAN.PAS**, så skriver du

.AMIS ALLAN.PAS

Det är bara första gången efter du har loggat in, som du måste skriva ut namnet på filen du vill editera. I fortsättningen får du alltid samma fil som förra gången, om du inte skriver något filnamn. Eftersom du får förkorta kommandon till datorn räcker det nästa gång med

.AM

Den andra editorn, som vi kan rekommendera, om du har en skrivande termina eller en skärmterminal som AMIS inte klarar av, är TECO. TECO har den fördelen framför AMIS att man kan skriva program i den. För att knappa in en ny fil respektive redigera en gammal använder du något av kommandona

.MAKE ALLAN.PAS

.TECO ALLAN.PAS

TECO kommer också ihåg vilken fil du arbetade med senast, på samma sätt som AMIS.

Hur du skriver ut en fil

Om du vill skriva ut en fil på terminalen, utan att gå in i en editor och titta på den, så använder du kommandot

.TYPE ALLAN.PAS

För att skriva ut filen på skrivaren i bruna eller orange salen använder du ett av följande kommandon, beroende på var du vill skriva ut den

```
.PRINT ALLAN.PAS/FORMS:BROWN  
.PRINT ALLAN.PAS/FORMS:ORANGE
```

Hur du ser vilka filer du har

Du kan få en lista på din terminal, med namnen på filerna i ditt directory, deras skyddskoder och datum du skrev in dem eller senast ändrade i dem med kommandot

```
.DIRECTORY
```

Ta bort filer

För att ta bort en fil använder du kommandot

```
.DELETE ALLAN.PAS
```

Om du vill ta bort alla filer, som heter ALLAN som första namn, gör du det enklast med kommandot

```
.DELETE ALLAN.*
```

Innan du loggar ut bör du ta bort alla filer du inte behöver, exempelvis sådana med extension BAK och REL, alltså

```
.DELETE *.BAK,*.REL
```

Kopiera filer

Om du vill ha en kopia av en fil, exempelvis för att kunna arbeta med den utan att förstöra den du hade från början, så använder du kommandot

```
.COPY KAKAN.PAS=ALLAN.PAS
```

som gör en kopia av ALLAN.PAS och kallar den KAKAN.PAS.

Byta namn på filer

För att ändra namnet på en fil, exempelvis ändra namnet KAKAN.PAS till TAR.PAS, skriver du

```
.RENAME TAR.PAS=KAKAN.PAS
```

Skyddskod på filer

Alla filer på Nadja har en skyddskod, som anger vad du och alla andra som kör på datorn får göra med den. Skyddskoden består av tre siffror med värden mellan 0 och 7, där den första siffran anger vad du själv får göra med filen och de två sista vad alla andra får göra med den. Normalt är första siffran 0, så du kan göra vad du vill med filen. Om du ändrar den till 3 så kan du inte ta bort eller ändra i filen, utan att först ändra skyddskod på

den. De två sista siffrorna ska vara 55, om du vill att alla andra ska kunna läsa vad som står i filen, eller 77, om bara du själv ska kunna komma åt den.

När du skapar en ny fil kan du ange skyddskoden mellan < och >

.AMIS ALLAN.PAS<055>

För att ändra skyddskod på en fil du redan har skriver du

.PROTECT ALLAN.PAS<355>

Om du vill att alla nya filer du skapar ska få en viss skyddskod, om du inte specificerar någon annan, ska du ha en fil som ska heta **SWITCH.INI**, som ska innehålla en rad med

LOGIN /DEFPROTECTION:055

Skriv

.HELP PROTECTION

om du vill veta mer om skyddskoder!

Om du vill låta vissa som kör på Nadja göra vissa saker med dina filer, men inte andra, så kan du ge filerna en skyddskod, vars första siffra är 4 eller högre, och lägga upp en fil med namnet **ACCESS.USR** på ditt directory. Om du vill veta mer om hur det fungerar kan du skriva

.HELP FILDAE

Begränsat skivminnesutrymme

Det finns två gränser för hur mycket skivminnesutrymme du får använda, en som gäller när du är inloggad och en som gäller när du är utloggad. Du får inte överskrida dessa gränser. Alla STACKEN-konton har gränserna 30 respektive 20 skivminnesblock från början. Det är ganska lite och om du behöver mer så får du ansöka hos styrelsen om att få mer, med motivering om hur mycket du behöver och vad du tänker använda det till.

För att få reda på vilka gränser som gäller för ditt konto och hur mycket av det tillåtna utrymmet du har utnyttjat kan du göra

.R QUOLST

Operativsystemet allokerar skivminnesutrymme i enheter om 5 block, så dina filer tar alltid upp 5, 10, 15, ... block, även om de inte fyller alla. För att få reda på hur stor plats dina filer egentligen tar ger du kommandot

.DIRECTORY/ALLOCATED

För att spara utrymme kan du packa ihop flera filer i en fil med programmet **LIBMAN**, som du startar med

.R LIBMAN

Om du har 4 filer på vardera 1 block, alltså i praktiken 5 block, var, så tar de upp 20 block på skivminnet. När du packar ihop dem i en biblioteksfil så kommer den att bli 8 block stor, summan av antalet filer och deras storlekar, och alltså bara uppta hälften så stor plats, 10 block, på skivminnet. Biblioteksfiler brukar ha extension **LIB**. De viktigaste kommandona i **LIBMAN**

är

LIBMAN>USE ALLAN.LIB	anger vilket bibliotek du vill använda.
LIBMAN>ADD ALLAN.PAS	lägger till en fil i biblioteket.
LIBMAN>DELETE ALLAN.PAS	tar bort en fil ur biblioteket.
LIBMAN>REPLACE ALLAN.PAS	byter ut en fil i biblioteket.
LIBMAN>COPY ALLAN.PAS	kopierar tillbaka en fil från biblioteket.
LIBMAN>DIRECTORY	visar vilka filer som finns i biblioteket.

För att du ska spara utrymme på skivminnet måste du ta bort filerna från skivminnet efter du har lagt in dem i biblioteket. Om du vill veta mer om LIBMAN kan du ge kommandot

.HELP LIBMAN

Filen SWITCH.INI

De flesta systemprogram läser av parametrar från en fil som heter SWITCH.INI, om du har en sådan fil. För varje program som läser den filen kan man ha en rad med programmets namn och vilka parametrar man vill ge till det. Filen SWITCH.INI kan exemlevis se ut på det här sättet

Login /DefProtection:055/TMP:KOM:"Vad-du-heter"

Directory /Width

Print /Preserve/File:ASCII

Den första raden betyder att LOGIN-programmet ska sätta upp vissa parametrar åt dig, nämligen att alla nya filer du skapar ska få skyddskoden 055 och att ditt namn ska läggas in i en temporärfil, som programmet KOM läser, så du inte behöver skriva in det själv, när du kör KOM. Nästa rad berättar för DIRECTORY-programmet att du vill utnyttja hela terminalbredden, så du får namnen på dina filer utskrivna i två kolumner. Om du inte har en DIRECTORY-rad med /WIDTH på, så skrivs filnamnen ut i en kolumn. Sista raden förhindrar att filer med extension LST tas bort efter utskrift på skrivaren och att filer med extension DAT skrivs ut som om de hade FORTRAN-styrtecken i första kolumnen, vilket de normalt inte har. Mer information om filen SWITCH.INI kan du få med kommandot

.HELP SWITCH

Hjälp

Med HELP-kommandot kan du få mer information om det mesta. För att få reda på hur HELP-kommandot fungerar skriver du bara

.HELP

Jan Michael Rynning

Hur du kör program på Nadja

Det första du måste bestämma dig för, när du ska skriva ett program, är vad du vill göra, hur det ska göras och vilket programspråk du ska använda.

Kompilerande språk

Om du väljer ett kompilerande språk, som ALGOL 68, BLISS 10, C, MIDAS, PASCAL, SAIL, SIMULA 67 eller SNOBOL, så måste du knappa in källkoden till programmet med en editor, innan du kan be datorn att köra det.

För att köra programmet måste du be datorn översätta källkoden till maskinkod, som består av ettor och nollor, länka ihop din maskinkod med färdiga rutiner, ladda in maskinkoden i minnet och starta programmet. Det gör du med kommandot

```
.EXECUTE ALLAN.PAS
```

Första gången du ger detta kommando får du, säkert som amen i kyrkan, en massa felmeddelanden på din terminal. Rätta felen i ditt program med editorn och försök igen!

När datorn kompilerar `ALLAN.PAS` läggs maskinkoden, som programmet översätts till, i en fil som heter `ALLAN.REL`. Den filen ska du ta bort innan du loggar ut, eftersom den tar plats och eftersom du kan få tillbaks den genom att kompilera om `ALLAN.PAS`.

I stället för att utföra alla stegen varje gång man ska köra programmet, så kan man spara den färdiglänkade maskinkoden på en fil, precis som den ser ut när den är inladdad i minnet. Det gör du med kommandona

```
.LOAD ALLAN.PAS
```

```
.SSAVE ALLAN.EXE
```

Då får du en fil, som heter `ALLAN.EXE` och är ännu större än `ALLAN.REL`.

Köra färdiga program

För att köra ett program, som är sparad som en `EXE`-fil skriver du

```
.RUN ALLAN
```

För att köra ett systemprogram, exempelvis BASIC, använder du kommandot

```
.R BASIC
```

Jan Michael Rynning

KATIA

Det här gör vi.

För att kunna köra **KATIA** måste **STACKEN** ha en lokal att ställa maskinen i. Några fundamentala saker krävs av lokalen nämligen:

- Relativt stort utrymme 45-55 kvm, helst uppdelat i ett större och ett mindre rum.
- Tillgång till 380V trefas el (ganska många ampere).
- Tillgång till vatten för luftkonditionering.

Det har visat sig ganska svårt att uppfylla all dessa krav samtidigt. Vi har dock hittat en lokal som skulle kunna vara möjlig (tillochmed bra) att använda. Svårigheten är bara att få loss den då annan verksamhet för närvarande pågår i dessa. Vi jobbar dock på det och hoppas väl att vi skall kunna ha fått en lokal till sommaren.

Under tiden sysselsätter vi oss med andra aktiviteter. Vi har tänkt oss att göra en dummyinstallation för att se om allt som utlovats finns med. Det har redan visat sig att det saknas en del dokumentation. Främst manualer. Dessa har vi nu påbörjat jakten efter. Vi jagar också ett ordentligt kylaggregat som kan kyla bort sådär en 15 - 20 KW plus sin egen uppvärmning. Är det någon som har kontakter inom denna branch så vore det behövligt med en insats.

Kort om hur en KA-10 fungerar.

Datorn är uppbyggd av sk FLICP CHIPS, ett modulsystem byggt med transistorer, där varje kort motsvarar en logisk enhet t ex en räknare eller ett antal grindar. Man kan säga att ett FLICP CHIP innehåller samma funktioner som dagens TTL-kretsar. Storleken skiljer dock en del för ett FLIP CHIP är ca 5x10 cm. Logiken är ickebottnad dvs spänningsskillnaderna mellan 0 och 1 är mindre än matningsspänningen. Matningsspanningen är ± 6 V, en logisk etta är - 3 V och en logisk nolla är + 3 V. Jfr TTL där matningsspänningen är +5 V och en nolla (i princip) är 0 V och en etta + 5 V.

En annan underlighet med **KATIA** är att den inte har en gemensam klocka som styr hela maskinen utan den gör allt med handskakning, sk Händelsestyrning och varje logisk enhet för sig håller med sin egen klocka. Jag skall förklara vad jag menar. Om CPU:n t ex vill göra en läsning från

kärnminnet går det till ungefär så här. CPU:n lägger ut adressen på en av de fyra minneskanalerna och "drar i ett snöre" för att tala om att den har gjort det. Det minnesskåp som känner sig adresserat läser in adressen och kvitterar genom att dra i ett annat snöre att den tagit emot adressen och CPU:n släpper då bussen. Denna kan då användas för andra ändamål t ex att göra DMA-accesser på en annan kanal. När så minnet har läst data från kärnminnet (görs egentligen med en skrivning i detsamma varvid innehållet i den accessade i minnescellen förstörs) lägger minnet ut datat på bussen och lyfter i ett tredje snöre. Under tiden skriver minnet tillbaka, det under läsningen förstörda minnesinnehållet. När CPU:n har läst in datat lyfter denna i ett fjärde snöre för att kvittera läsningen. Minnet släpper då bussen. Detta förfarande, kallas också handskakning, gör att CPU:n kan jobba så fort det går. I ett klokat system är ju klockan anpassad till någon sorts medelfart där t ex långsammare minne måste bromsa CPU:n. Om vi t ex får tag i snabbare minne kommer maskinen följaktligen att gå snabbare.

För den som är intresserad att lära sig mera om hur **KATIA** fungerar, både hårdvaru- och mjukvarumässigt, finns all dokumentation i ett av tråskåpen vid Undervisningsdatorn **Elmer**. Ta kontakt med Göran Wallberg så släpper han in Dig i skåpet. **Tyvärr kan Du inte ta med Dig något av dokumentationen utan får läsa den på plats. Dokumentationen är främst till för de som direkt arbetar med KATIA.**

Hur arbetet är organiserat.

Den gamla KATIA-gruppen upplöstes av den nya Styrelsen. En ny bildades med Mats Jansson och jag själv som organisatörer och totalt är vi idag av 6 personer. Alla har åtagit sig att utföra ett antal uppgifter och samordningen av detta arbete sker dels via KOM och dels via mer personliga möten. Det kommer naturligtvis att behövas mera folk så småningom och information om detta kommer i KOM och på månadmötena. Har Du förslag på hur specifika delar kan lösas är Du naturligtvis välkommen med synpunkter. Du kan t ex skriva ett brev till Mats Jansson eller mig eller, ännu bättre, skriv ett brev till **KATIA Planeringsmöte**. Är det något område Du speciellt vill ägna Dig åt eller kan särskilt bra så hör av Dig.

Danny Kohn

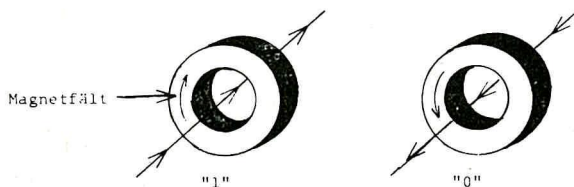


Fig. 2.17 Ferritkärnor i två olika tillstånd

PL/1, vad är det ?

PL/1 är IBMs stora Algol-språk och är en förkortning av "Programming Language One". Tyvärr är det nog inte det bästa språket i världen, vilken nog var skaparnas mening.

När PL/1 skapades i början av sextio-talet så fanns det vissa användargrupper i USA som ansåg sig behöva ett programmeringsspråk som både kunde klara vetenskapens och ekonomins behov. Elakt sagt en blandning mellan Fortran och Cobol. IBM satte sig ner med två sådana grupper (SHARE och GUIDE) och började skapa PL/1. SHARE var den gruppen som var intresserad av vetenskapliga- beräkningar medans GUIDE var intresserade av ekonomiska- beräkningar.

PL/1 har till viss del ärvt en del av Cobols "pratighet" men eftersom det är ett Algol liknande språk så har man för det mesta även en kortare variant. T.ex ordet **DECLARE** kan även skrivas **DCL** och **POINTER** kan förkortas till **PTR**.

När man första gången tittar i ett PL/1 program blir man bara förvirrad över alla olika variabel typer. Detta beror på att man själv bestämmer med vilken precision datorn skall räkna. Detta gör att man alltid får fram samma resultat oberoende av dator fabrikat. När jag i fortsättningen kommer att nämna PL/1 är det då iförsta hand den kompilator som finns till VAX/VMS. VAX-11 PL/1 (ver 1.2) är inte en komplett PL/1 utan ett subset av PL/1. Detta subset kallas för Subset G och är till för datorer som använder ASCII istället för EBCDIC.

PL/1 kräver inte att man deklarerar alla variabler om man inte gör det så kommer kompilatorn att själv sätta en standardtyp. Att PL/1 själv sätter variabeltyp på odeklarerade variabler är ett tydligt arv från Fortran. Jag tycker PL/1, tyvärr, bär allt för stora spår efter Cobol och Fortran.

T.ex. variabeltypen **PICTURE** finns nästan bara i Cobol och i Basics **PRINT USING**. Varför det är så förstår jag inte, för jag tycker att det är en användbar variabeltyp. Fördelen är att man kan använda den både som sträng och som numeriskvariabel.

Något som jag saknar hos många språk är en ordentlig fel-hantering. På den punkten är PL/1 riktigt starkt. Detta kallas för ON-enheter i PL/1. T.ex. kan man ta han om slut på fil,fil finns ej, division med noll m.m. Om man inte fångas upp av någon ON-enhet kan man alltid fånga upp "oväntade" fel med **ON ERROR**. Kan man då inte ta han om felet där heller så kan man låta programmet signalera felet.

VAX-11 PL/1 är ett fattigt språk om man tittar på vilka styrstrukturer som finns. I princip är det två typer **IF-THEN-ELSE** och **DO-WHILE**. **IF-THEN-ELSE** kan se ut på flera sätt då det finns två sätt att skapa samlingar av satser, man kan börja med **BEGIN** eller **DO** vilket man använder har ingen betydelse, om man bara skall utföra en sats så behöver man ingen **BEGIN**-eller **DO**-grupp. Till skillnad från Pascal så tillhör alltid senaste **ELSE** närmast föregående **THEN**-stats.

```
IF foo > 4711
THEN DO;
    s1; s2; s3; ... sn;
END;
ELSE BEGIN
    s1; s2; s3; ... sm;
END;
```

PL/1:s andra styrstruktur är **DO-WHILE**-satsen som har följande utseende:

```
DO i := 1 TO n WHILE (i < 100)
    s1; s2; s3; ... sn;
END;
```

PL/1:s motsvarighet till Pascals **FOR**-slinga är en variant av **DO-WHILE**-satsen där **WHILE** har plockats bort och endast **DO-TO** återstår. På samma sätt är motsvarigheten till Pascals **WHILE**-sats ett special fall av **DO-WHILE**, där det inte står något mellan **DO** och **WHILE**.

Vill man skapa en struktur i PL/1 måste man även ange ett nivå nummer framför variabeln för att kompilatorn skall kunna förstå att det är en struktur, t.ex

```
DCL 1 PERSON,
    2 FORNAMN CHAR (10),
    2 EFTERNAMN CHAR (20);.
```

Då det även är tillåtet att deklarerat på följande sätt

```
DCL FORNAMN CHAR (10),
    EFTERNAMN CHAR (20);
```

vilket är en helt annan sak.

PL/1 har även en motsvarighet till Pascals **CASE**-sats, den heter **SELECT**. Dock finns den inte i VAX-11 PL/1 för närvarande. Men den kommer i den nya version av VAX-11 PL/1 (ver 2.0) som kommer inom den närmsta tiden.

I PL/1 ingår ett stort antal s.k. "Built-In"-funktioner vilket gör att man har tillgång till ett stort antal användbara rutiner direkt i sitt program. Man kan t.ex. skapa en rutin med samma namn som en "built-in"-funktion.

I denna egna rutin kan man anropa "built-in"-funktionen bara genom att deklarerar den:

```
DCL DATE BUILTIN;
```

Men vad kan PL/1 som inte övriga Algol-språk kan? Det är nog inte så mycket. Den största fördelen med PL/1 är att man kan lätt gå förbi de kontroller som finns i kompilatorn. Detta är möjligt genom att man har möjligheten att lägga olika variabler "ovanpå" varandra. PL/1 har en relativt bra stränghantering.

Det finns dock inget i PL/1 som är så bra att jag vill rekommendera någon att överge Pascal, Simula eller Algol för PL/1s skull. Att jag programmerar en hel del i PL/1 beror bara på att jag gör det på dagarna. Men just för tillfället är det nog Modula-2 som intresserar mig mest.

MOJ

Hur Murphy's lag kom till

Det hände 1949 på Edwards flygbas i Muroc, Kalifornien, i samband med projekt MX981. Överste J. P. Stapp utförde krasch-försök på den norra basen. Försöken genomfördes av Northrop Aircraft på kontrakt av flygmedicinska labbet vid Wright Field.

Vid flygplanslabbet fanns det en kapten Ed Murphy. Besvärigheter med trådtöjningsgivarvästet för bälten gjorde att Murphy kom med kommentaren:

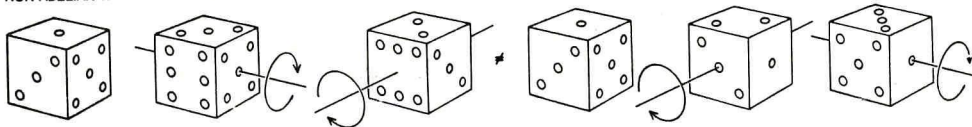
"Finns det en möjlighet att göra det fel, så gör han det" syftande på teknikern som utförde monteringen på labbet.

Ett par veckor senare, vid en presskonferens, förde överste Stapp fram, bl. a. de fina resultaten med avseende på lågt antal olyckor. "Det var," sade Stapp, "helt beroende på medarbetarnas verkliga tro på Murphy's lag." Efter ett par månader introducerades denna lag i fabrikantens annonser.

Det var födelsen till en av de mest kända lagarna. (Berättelsen kommer från George E. Nichols vid JPL-NASA. Vid tiden för försöken vid Wright Field var han ansvarig för N.A.'s kraschtesten.

/hn

NON-ABELIAN TRANSFORMATION



En titt på TV-spelen i TV-spelens förlovade land – Kiseldalen

Nyss hemkommen från Kiseldalen. Åker man till Kiseldalen bör man inte försumma att titta in i några av de TV-spelshallar som finns i området. Utvecklingen har ingalunda stagnerat, nya typer av penningsslukande maskiner kommer hela tiden. Det här är en sammanfattning av mina intryck från ett par sådana besök.

Den stora nyheten är videospel *med bilder lagrade på videoskiva* i stället för datorgrafik. Kombinationer finns även. Det finns spel där man blandar klassisk datorgrafik och filmade scener. Flera spel innehåller filmade flygscener och dogfights, där bakgrunden består av en videofilm, och förgrunden av en bild av den egna farkosten som man styr med en spak på vanligt sätt.

Det utan tvekan populäraste spelet är **Dragon's Lair**, ett videospel i fantasy-miljö. De rörliga bilderna består av en tecknad film med olika händelsegrenar. Spelet handlar om Dirk The Daring som ska ge sig in i ett spökslott för att rädda den fagra prinsessan och plocka upp lite skatter på vägen. Stället är nerlusat av monster, fällor, och elaka trollkarlar. Till sin hjälp har man en spak som kan be Dirk gå framåt, bakåt, till vänster och höger, samt en knapp som får honom att använda svärdet. Man tittar på filmen och styr Dirk genom olika hårresande situationer med reglagen – och någon betänketid ges inte!

När man valt en aktion växlas videoskivan till ett nytt spår där man får se hur det gick. Har man tur överlever Dirk, och kan vandra vidare till nästa knepiga situation. I annat fall... Man får 5 Dirkar för 50 cent.

Dragon's Lair är väldigt roligt, men blir snabbt dyrt eftersom man har fem valmöjligheter i varje situation, och man måste rycka i spaken inom ett mycket kort tidsintervall. Nybörjaren rekommenderas att först titta på när andra spelar ett tag, och lära sig åt vilket håll man ska vända sig i de första situationerna.

Ett problem med spel där man hämtar scener från en videoskiva är att man får en "blackout" då handlingen (och skivan) byter spår. "Dragon's Lair" lider av detta. Utveckling pågår dock för att eliminera denna paus, och många av videospelstillverkarna ligger nu i startgroparna för att lansera fler spel av detta slag, utan blackouts.

De vanliga videospelen med datorgrafik lever dock kvar, och förfinas. Star Wars (med scener ut filmen) är faktiskt mycket bra, till skillnad mot **TRON**, som ju annars är det uppenbara "spin-off"-spelet. Mina favoritspel (förutom "Dragon's Lair") är *Star Wars* och *Millipede*.

Videospel till hemdatorer har även börjat säljas i stor skala ute i butikerna. Dels i skivbutiker och dels i nyöppnade affärer som bara säljer programvara (dock inte bara spel). Även vissa bokhandlar har lite programvara på hyllorna för datorlitteratur. Marknaden för videospel är ännu mycket omogen, och många produkter och tillverkare tävlar om hyllutrymmet. Det påstås att spelen med de snyggaste förpackningarna säljer bäst. Ett spel kostar mellan 20 - 50 USD.

Per Linberg

Fadderverksamheten

Att kunna något är ofta svårt. Hur få reda på var du ska leta är svårare. Som nybliven medlem i föreningen är du ovan vid mycket. Vilken verksamhet som är på gång? Hur datorn Nadja hanteras? Var de olika lokalerna finns? Ja, det finns säkert många fler frågor.

För att smidigt kunna komma in i föreningen kan det behövas lite hjälp. Vi är några stycken som försöker stå till tjänst. Vi börjar i samband med att de nya personliga kontona utdelas. Det blir en liten genomgång av de grundläggande kommandona på TOPS-10. Dessutom är det en genomgång av de regler som gäller för användningen av Nadja och KOM. Alla frågor besvaras allteftersom de kommer fram. Fadderverksamheten är inte tänkt att ersätta egna studier. Snarare är det ett komplement och möjligtvis katalysator. Genom den här verksamheten har du också fått tillfälle att få ett namn på ett av ansiktena i föreningen. Det är en av personerna Mats Jansson, Danny Kohn, Joakim Langlet, Hans Nordström eller Göran Wallberg. Vår förhoppning är att du återkommer till oss med framtida frågor om det blir några. Vi skall se till att du får svar och hjälp.

/hn

E de någe å ha?

Chips kommer och går. Just nu skall det av någon anledning vara 16-bitars. Emedan alla inte är kopior av varandra, så uppkommer frågan: "Vilken är bäst?" Jag hittade följande jämförelse i en tidskrift, jag lånat. (Tack Lukas!)

Category	Z80	Z800	LSI11	Z8000	8086	80286	68000	16032
Multiple stacks	D	D	A	A	C	C	A	C
Hardware multiply/divide	D	C	A	A	B	B	A	A
Addressing modes	D	C	A	B	C	C	B	B
General purpose registers	D	C	B	A	C	C	A	B
String operations	B	B	C	B	B	B	C	A
Address space	64K	512K	4M	8M	1M	16M	16M	16M
Addressing implementation	lin	seg	seg	seg	seg	seg	lin	lin
Multiuser considerations	D	C	D	A	D	B	A	A
Small task support	A	A	A	A	A	A	C	C
Processor speed (MHz)	8	10	3	6	10	10	12	6
Processor pipelined?	n	y	n	n	y	y	n	y
Floating . processor	9511/ 9512	9511/ 9512	FIS/ FP11	8070	8087	80287	68881	16081
Basic FPU architecture	D/C	D/C	C/B	A	B	B	A	A
Transcendental functions	C/D	C/D	D	C	B	B	A	D
Conventional O/S support	B	B	A	D	B	B	C	D
Language/applications	A	A	A	D	B	B	C	D
Literature	A	B	B	B	A	B	B	D

A = Excellent B = Good C = Fair D = Poor

Vad är då grunden för jämförelsen? Jo, hur bra passar chipet för FORTH!

/hn



Real Programmers Don't Write Specs

Real Programmers don't write specs — users should consider themselves lucky to get any programs at all and take what they get.

Real Programmers don't comment their code. If it was hard to write, it should be hard to understand.

Real Programmers don't write application programs, they program right down on the bare metal. Application programming is for feebs who can't do systems programming.

Real Programmers don't eat quiche. They eat Twinkies, and Szechwan food.

Real Programmers don't write in COBOL. COBOL is for wimpy applications programmers.

Real Programmers' programs never work right the first time. But if you throw them on the machine they can be patched into working in only a few 30-hour debugging sessions.

Real Programmers don't write in FORTRAN. FORTRAN is for pipe stress freaks and crystallography weenies.

Real Programmers never work 9 to 5. If any real programmers are around at 9 AM. It's because they were up all night.

Real Programmers don't write in BASIC. Actually, no programmers write in BASIC, after the age of 12.

Real Programmers don't write in PL/I. PL/I is for programmers who can't decide whether to write in COBOL or FORTRAN.

Real Programmers don't play tennis, or any other sport that requires you to change clothes. Mountain climbing is OK, and real programmers wear their climbing boots to work in case a mountain should suddenly spring up in the middle of the machine room.

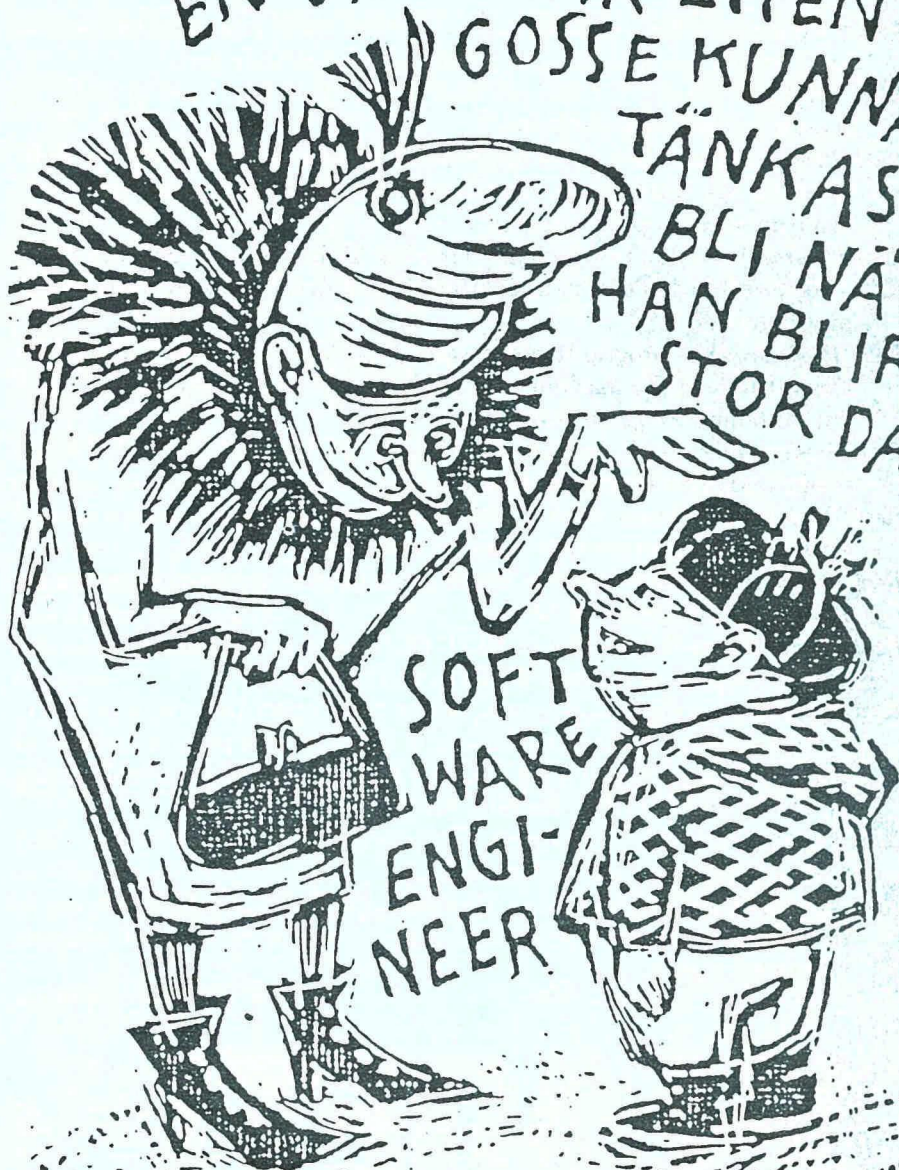
Real Programmers don't document. Documentation is for simps who can't read the listings or the object deck.

Real Programmers don't write in PASCAL, or BLISS, or ADA, or any of those pinko computer science languages. Strong typing is for people with weak memories.

trad.

A-LA-GET!

OCH VAD SKA
EN SÅN RAR LITEN
GOSSE KUNNA
TÄNKAS
BLI NÄR
HAN BLIR
STOR DÅ?



SOFTWARE
ENGINEER