Microcomputer User Group 1993:3

Der indkaldes til ordinær generalforsamling i MUG

onsdag den 27 okt. 1993, kl. 19.00,

Vesterbro Kulturhus, Lyrskovsgade 4, København V.
(Lokale 2+3)

Dagsorden:
1. **Valg af dirigent og referent.**
2. **Formandens beretning.**
3. **Kassererens beretning.**
4. **Bibliotekarernes beretning.**
5. **Indkomne forslag.**
6. **Valg af formand.**
7. **Valg af bestyrelsesmedlemmer.**
8. **Valg af 2 revisorer.**
9. **Eventuelt.**

Bestyrelsen.

I dagens anledning trækkes lod om diverse objekter, doneret af forskellige sponsorer. Medl. får udleveret et lod ved indgangen.

Næste møde er:

tirsdag **23 nov** **Kamin-komsammen/emner medbringes *****

se også næste nr. af bladet - kik på BBS'et - spørg en medlems-kollega

***** Bidrag/Indlæg/Forslag til møder og blad er velkomne!**

Har du en god ide, et praktisk tip, som andre måske kan bruge eventuelt et forslag til noget, som du gerne vil vide mere om så lad os høre nærmere.

Som (symbolsk) belønning donerer foreningen en diskette-check til indløsning i et af bibliotekerne.

Møderne afholdes på adressen:

Vesterbro Kulturhus, Lyrskovsgade 4, København V.

Kl. 19. Lokale_nummer fremgår af opslag ved indgangen.

INDMELDELSE OG KONTINGENT

Indmeldelse i MUG Denmark foretages ved indbetaling af kontingent (225 kr. årligt) på Giro 5 68 65 12, MUG Denmark /v Lars Gråbæk

Oplag: 300
Tryk: Dansk Tidsskrifts Tryk.
Redaktion: Viggo Jørgensen.
Redaktionen afsluttet primo oktober 1993.

EDB-sprog og manual-sprog.

Klumme
EDBPROG2.393

RAID = FENABDED

Det er en naturlig ting, at et fagområde opdyrker sine egne fagudtryk, der på en præcis måde skal beskrive de ideer og elementer, som man arbejder med. Det letter samtale mellem fagets udøvere, og dermed fordeling af opgaver, beskrivelse af fejl mm.

Ordet *system* anvendes om mange forskellige emner, fordi det er så praktisk: Det fortæller, at her er der tale om noget, der fungerer som en enhed, fx. en samling programmer, som tilsammen er et tekstbehandlingssystem, eller en disk-controller, som sammen med nogle harddiske udgør et disk-system, en enhed, der opfylder den funktion, at maskinen har et sted at gøre af sine data, når den vil gemme dem af vejen for kortere eller længere tid.

I forbindelse med en ny serie netserver computere fra IBM så jeg følgende beskrivelse af, hvordan flere "små" harddiske bliver sat sammen, så de for maskinen ser ud som én stor disk. Derved kan man fx. opnå at gemme fx. 10 store billedfiler på hver 108 MB på 4 diske à 270 MB uden at skulle spekulere på, hvilken disk, der nu skulle bruges. Der kan i eksemplet kun være 2 x 108 MB = 216 på én disk, som kan rumme 270 MB, de sidste 64 MB ville da ikke kunne udnyttes.

Hvis flere diske "ser ud" som en stor disk, kaldet en "logisk" disk, så ville maskinen kunne lægge 64 MB af det tredie billede på den første disk, og de sidste 44 MB på den næste disk. (Har nogen forresten hørt om en "ulogisk disk"?)

Diske, der er sammensat på denne måde, kaldes et *disk array*. Ærligt talt, jeg ved ikke, hvordan man normalt oversætter det engelske array til dansk, men det svarer til ordet område; men bedre end "disk-område" vil det nok være med betegnelser som disk-sammenkobling, disksøjle, diskbatteri, diskregiment, disk-hær, lagringsarsenal, diskgruppe.

Nej, der er noget i vejen med alle de betegnelser, ikke sandt?

Derimod kan ordet "disk system" fungere indenfor vores vante edb-sprogvaner, man kan næsten gætte sig til, at det er nogle enheder, som tilsammen fungerer som en lagringsenhed.

I den før omtalte skrivelse forklares lidt om de forskellige måder, hvorpå et sådant disk system kan fungere:

> »RAID 0 (Redundant Array of Inexpensive Disks, level 0) er Data Striping without Parity (DSA), d.v.s. "fordeling" over flere fysiske diske. Alle diske betragtes som én stor, logisk disk.
> Dette giver en utrolig god perfformance, da der er flere disk-arme til rådighed for læs/skriv og samtidig mulighed for meget store filsystemer, men da der ikke er hverken dublikering eller fejlkorrigering, er sikkerheden minimal, derfor kombineres RAID 0 oftest med RAID 1« *

De mange anglicismer (performance) og fejlstavninger (dublikering) har ikke **kun** noget at gøre med "fagets udtryk", men skal snarere forklares med, at de danske bøjningsformer virker ulogiske, når man lige har læst eller "tænkt" noget på det fremmede sprog.

Med andre ord samme afsmitning, som man oplever mellem sprogene i grænseegne; samme sprogforvirring, som udvandrere eller rejsende kan opvise, når de i længere tid ikke har brugt deres modersmål.

Hvis man ikke helt vil gå over til det udenlandske sprog, hvor skal man så sætte

---

* Citeret fra reklamebrev til almindelige virksomhedsledere fra WRP Data, Fakta og Forlydender, 9 sept 1993. Redaktionens fremhævelser.

grænsen? Skal RAID oversættes til "Flere End det Nødvendige Antal Billige Diske konfigureret som En Disk"? Ved udvælgelse af de store bogstaver får jeg FENABDED, men det kan måske gøres bedre. Pointen er selvfølgelig, at ingen uden for en radius af 3.75 i porto vil kunne fatte betydningen af den forkortelse, og heller ikke mange inden for dette område.

Nå. Vi beholder altså RAID. Ud fra denne tanke bevares også "Data Striping without Parity", eller snarere "Data Striping Array", DSA.
For min skyld kan man også bevare performance (hastighed), fil (kartotek eller dataregister), dublering (men ikke dublikering eller duplikering, please!). I stedet for fejl-korrigering ville jeg foretrække fejlkorrektion.

Kort sagt, det er svært at anføre alvorlige indvendinger mod de enkelte dele af ovenstående citat, men tilsammen bliver effekten, at det lyder for meget som en engelsk-dansk blanding. Det er der nok ikke meget at gøre ved, med mindre man foretrækker et fagsprog, som er helt fordansket og som vil gøre det endnu mere vanskeligt at forstå manualer og artikler udefra.

Et andet problem er, at de, der skulle have glæde af ovenstående RAID-beskrivelse, fx. regnskabsmanden i et firma med 40 pc'er i netværk, måske ikke ligefrem fatter tillid til en leverandør, som udtrykker sig på datadansk eller kaudervælsk edb-sprog. Eller er meningen mon den, at de mange fine ord skal virke imponerende, duperende?

DISK ARRAYS

Emnet DISK ARRAYS er ellers interessant nok. Tidligere var store disksystemer forbeholdt kunder med et meget stort edb-budget.

Sikkerheden bestod i den daglige backup, hvis man overhovedet foretog en sådan.

Med faldende hard-warepriser er situationen i dag en anden, selv om der er tale om pebrede priser i forhold til almindelige kloner og bordmodeller. Men skal vi af en eller anden grund op i datamængder på mere end et par gigabyte, så er disksystemer stadig dyre, og det er da nærliggende at forestille sig et system, hvor flere billigere diske tilsammen gør det ud for én, derfor er RAID 0 et interessant bud på en løsning. Med to diske på 500 MB ville man kunne rumme en fil på op til 1000 MB i modsætning til almindelige systemer, hvor man måtte splitte op i 2 filer på hver 500 MB.

Gad i øvrigt vide, hvor stor Fil Allokerings Tabellen, FAT, bliver for så store diske, og hvordan operativsystemet klarer det.

På en "lille" 100 MB disk er FAT tabellen allerede oppe på 192 KB, så på en 500 MB disk må den altså være på 960 KB, hvis man da ikke kan ændre blokstørrelsen (egtl. cluster-størrelsen, den plads en fil på én byte optager på disken).

Af de andre RAID - niveauer (levels) er det interessante især RAID 3, hvor en checksum af de til diskene skrevne data nedfældes på en anden harddisk. Denne validering giver større sikkerhed for, at de læste data er korrekte, noget man sikkert gerne vil betale for i en virksomhed med flere tusinde transaktioner hver dag.

Det skaber imidlertid en flaskehals når man skal læse data, alle skal da bruge disken med checksummen. Dette løses med RAID 5, der kan det samme som RAID 3, men fordeler checksummen på flere diske, for derved at opnå større hastighed.

Desværre var prisen på maskiner med denne type disksystem ikke opgivet i nyhedsbrevet, (de kan heller ikke leveres endnu), men det er nok ikke helt forkert deraf at slutte, at de ligger fra 150.000 kr og opefter (IBM 95A-3NG med max. 5.6 GB disk-kapacitet, 3NT med max 7GB, ja: 7000 MB!)

SLANGEN I PARADIS

På maskiner med mulighed for RAID vil der formentlig være problemer med kompatibilitet, som får betydning for, hvilke operativsystemer og netværkssystemer, som kan benyttes. Der vil altid være mulighed for at benytte et netværk, som ikke går direkte i hardware (fx. visse

versioner af Lantastic), men spørgsmålet er så, hvor meget man derved taber i hastighed, i forhold til et optimeret eller optimerbart system.

RING PÅ 9988 7766 OG TRYK DERES KONTONUMMER

Man kan forestille sig, at denne forrygende udvikling af billig, pålidelig maskineri inden længe vil muliggøre en masse administration, som man før mente var for besværlig.

Forestil Dem fx., at interessegrupper som fx. Dansk Naturfredning vil kunne overkomme administration i en helt anden størrelsesorden, end den vi kender i dag.

I en kampagne for fredning af orkideer vil man måske bede alle interesserede om at indsætte et beløb på en bestemt konto og give svar på et spørgeskema, pr. tlf. med afstemnings knapper eller fra en TV-computer, og uden en hær af interviewere har man dagen efter en meningsmåling, som vil blive taget alvorligt.

Det forhindrer ikke saglig information og debat.

Dette er naturligvis et uskyldigt eksempel. Jeg tør slet ikke tænke på indsamlinger til flygtninge og krigsinvalider.

Hvis der nu fx. kan foretages skattefradrag for støttebeløb, og hvis man vil forbedre indsamlingsresultatet ved at hjælpe folk, så de ikke behøver at holde regnskab med fradragets størrelse, så kunne et pålideligt datasystem være af særlig stor betydning.

"Bare ring på 9988 7766 og tryk Deres kontonummer eller Dan-kort og beløbets størrelse, så sørger vi for, at Deres støttebeløb bliver fratrukket på selvangivelsen".

Andre eksempler på områder, hvor decentral eller folkelig edb kan have betydning som alternative kræfter er mange; i de første år er det nok først og fremmest administration og pengeoverførsler, man vil se en udvikling på.

ShareWare programmer, som kan interface til mere eller mindre genialt udtænkte betalingssystemer, vil begynde at komme frem.

Efter nogle år vil også andre områder, som ikke giver mulighed for direkte indtjening, nyde godt af pålidelige edb-systemer, som kører i døgndrift og hurtigt kan aflevere information i store mængder. Tænk fx. på, hvordan vores FIDO-net længe har været et alternativ til det exclusive unix-net i universitetsmiljø, og tænk på, hvor dejligt det ville være, hvis det altid virkede...

Hvis det er rigtigt, at man om et halvt års tid vil kunne få '38.000' modemer, så vil det tage 4-5 minutter at hente en fil på 1 MB fra et BBS eller lign., noget, der med gammeldags modemer ville vare over en time. Der skal nok blive noget at bruge disk-pladsen til!

D. A.

```
The trouble with some people is
that they won't admit their faults,
I'd admit mine - if I had any.
```

Resultater fra test af tapestreamer software.

TAPEPRGW.393

Det tyske datatidskrift **MC**, har lavet en del test af tapestreamere, og den tilhørende software. MC's redaktion fik et ret nedslående resultat af de test de havde lavet.
Resultaterne til denne artikel er hentet i *MC nr 2/92, 2/93 & 10/93*

Når bladet, MC, tester tapestreamere, testes ikke blot hardware, men også softwaren. Ofte havde hardwaren en udmærket kvalitet, hvorimod det samme ikke kunne siges om softwaren.

Ofte er softwarens kvalitet meget ringe, selvom meget af softwaren tit har fine menuer, der sælger programmet godt.
Men en test af backup-softwarens hovedopgave, – nemlig det at kunne lave sikre backup, klares meget dårligt.

Hovedproblemet er, at kun sålænge der ikke opstår fejl på båndet, fungerer softwaren. Så snart der er blot den mindste fejl på båndet opgiver mange programmer.
Ofte er det ikke muligt at læse efterfølgende volumes (f.eks. ved *incremental backup*) på båndet, og for de fleste programmer er det i hvertfald ikke muligt at læse efterfølgende filer.

Udfra de test der er lavet i MC, har jeg kun fokuseret på sikkerheden af programmerne. Vil man se andre resultater, som f.eks. hastighed ved backup og restore, om programmerne kun kan lave backup af DOS-filer, eller også af hele harddiske eller partitions, eller om programmet kan være på en BOOT-diskette, så henvises til de nævnte numre af bladet MC.
Noget andet man bør være opmærksom på, er at det er sjældent at backup-software kan læse bånd fra andre backup-programmer.

MC's havde testet softwaren ved at spole et stykke ind på båndet, og så slette lidt af båndet med en magnet.
Derefter undersøgte man så hvordan backup-softwaren reagerede. En del software **går i stå !** ved første fejl der findes på båndet, og resten af båndet er således ulæseligt **!**

Forklaring til tabellen:
ECC angiver at programmet kan benytte fejlkorrigerende metoder, så at data ved små fejl kan genskabes (båndet kan ikke repareres).
For at kunne teste programmernes reaktion ved båndfejl, blev en så stor del af båndet slettet, at fejlkorrigeringen ikke kunne reparere fejlen.

Næste kolonne angiver om man kan læse filerne fra de efterfølgende volumes på båndet.

Sidste kolonne er den sværeste at opfylde, idet kun få backup-programmer kan læse alle filer pånær de ødelagte. ⟶

Frank Damgaard

```
The bitterness of poor quality
remains lomg after the sweetness
   of low price is forgotten
```

Program navn:	har ECC	læser volumes efter fejl	læser filer efter fejl
Arcserve/Solo 2.11	?	Ja	Ja/nogle gange
DOS Arch 2.00	?	Ja	Ja/*1
DOS Arch 2.20	?	Ja	Ja/*1
EZ-tape 2.22	Ja	Ja	Ja
File-Secure 1.55	?	Ja	Ja
Maynstream 3.1	?	Nej	Nej
QIC-Stream 2.01	?	Nej	Nej
QFA Tape-System 2.45	?	Nej	Nej
Sixtus 1.39	?	Ja	Nej
Sytos Plus 1.21	Ja	Ja	Nej
Sytos Plus 1.32	Ja	Ja	Ja
Central-Point Backup (PC-tools 8-0)	Ja	Nej	Nej

Bemærkninger:
? Det vides ikke om programmet har ECC
*1 Hvis programmet benytter ECC, er der i denne test ikke undersøgt hvad der sker hvis fejlen på båndet er for stor til at ECC kan reparere skaden.

Test af Norton Commander version 4.0

COMNDR4.393

Jeg har nu i ca 14 dage haft lejlighed til at prøve NC Commander ver. 4.0, og er meget tilfreds med dette program.

Nu kan et helt directory med alle underkataloger, kopieres i én arbejdsgang. Ligeledes kan man flytte og slette på samme måde. Det er ganske rart.

Man kan pakke- og udpakke filer med flere forskellige metoder, og en god detalje er det at man direkte kan se hvad de forskellige filer indeholder.

Det er ikke nødvendigt at have diverse ekstra programmer til det. Der er også mulighed for at læse andre filformater, selv Windowsfiler. Ved hjælp af F-taster kan man nu også få sorteret som det passer bedst til ens temperament.

Der er også en udmærket editor, som er brugt til at skrive denne lille anmeldelse. Den kan alt hvad der er behov for til mindre opgaver og kan nu også printe ud , blot ved at taste F9, istedet for som før at kopiere til LPTx eller PRN.

Der er også en søg- & erstat funktion i editoren.

En fin og forbedret detalje er filefind, hvor man kan søge efter en tekststreng, uden at vide hvilken fil den ligger i.

— — —

Det er kort sagt et program der er meget anvendeligt, kun synes jeg at prisen er alt for høj, men den skavank er der desværre mange andre programmer som lider under.

Erk Martinsen.

MASKINEN

TRACYW.393

Tracy Kidder, det lyder som et dæknavn. En, der er "kidding you" er en, der "narrer dig" eller driver gæk med dig. Navnet "Tracy" er nok mest kendt fra detektivserien "Dick Tracy", og "trace" betyder jo også at spore. Ikke desto mindre er det iflg. Nyt Nordisk Forlag navnet på den unge journalist, som i slutningen af 70'erne påtog sig opgaven at skrive om et udviklingsprojekt på Data Generals hovedsæde i USA, Westborough, og som vandt en Pulitzer Prize på bogen. Det er sjældent nødvendigt at bruge pseudonym, når man har vundet en sådan påskønnelse, men ok, det kan jo være, at forfatteren har et "rigtigt" navn, som han synes lyder så forfærdeligt, at han hellere vil kaldes noget andet; eller det kan være et morsomt sammenfald, at én, der på glimrende vis efterspørger alle de mange facetter i udviklingsarbejdet på et kompliceret, højteknologisk projekt, hedder "Tracy".

Denne spændende og lidt mærkværdige bog blev "bestilt" af projektlederen Tom West, der havde lige så meget forstand på psykologien i en arbejdsgruppe som på datamater. Han mente, at den opmærksomhed, som de unge ingeniører oplevede ved at blive interviewet og få lejlighed til at fortælle om deres syn på projektet, ville være med til at skabe arbejdsglæde og give en stærkere oplevelse af projektets betydning.
Projektet var et lavpris projekt, som ingen i den øverste ledelse rigtigt troede på; den egentlige, langsigtede forskning i nye CPU'er skulle foregå i en forsknings"park" i North Carolina, hvor et hold eliteforskere skulle arbejde koncentreret og uforstyrret mod en ny generations maskiner.

Konkurrenten, Digital Equipment Corporation, i daglig tale kaldet DEC, var allerede begyndt at udspy 32-bit maskiner "som vingummier", den legendariske VAX11/780. Det gjaldt derfor om at Data General så hurtigt som muligt skulle komme på gaden med en 32-bit maskine, for ikke at tabe markedsandele. Data Generals grundlægger De Castro var begyndt som ingeniør hos DEC, men efter nogle år dér mente han, at han kunne gøre det bedre og billigere og startede med et par andre for sig selv, i begyndelsen ganske beskedent, men de fik hurtigt succes. For at holde markedsandele var det nødvendigt at holde trit med DEC.

Tracy Kidder fortæller således om Tom West:

> Inderst inde var West bange for VAX. DEC havde offentliggjort en hel del teknisk litteratur, som gav en beskrivelse af VAX, og West havde læst det hele. Og intet af dette materiale havde fået ham til at føle, at hans teams måde at angribe sagen på stod tilbage for DEC's. Men for nogle ingeniører fører læsning ikke til nogen holdbar viden. For dem er berøringen den vigtigste sans. Så derfor tog West fri en dag i 1978, og mens hans team allerede var godt i gang med arbejdet på deres egen maskine, drog West af sted fra Westborough for selv at kaste et blik på VAX.
>
> Han rejste til en by, som han kun ville oplyse lå et sted i Amerika. Han gik ind i en bygning, som om han havde sin vante gang der, fortsatte ned ad en korridor og lukkede sig forsigtigt ind i et rum uden vinduer. Gulvet var brækket op; en form for rende fyldt med tykke elektriske kabler var lagt ind under det. Nede ved den fjerneste væg, for enden af renden, stod et splinternyt eksemplar af DEC's VAX indbygget i flere store kabinetter, som kunne have en svag lighed med køleskabe. Men til West's overraskelse stod et af kabinetterne åbent, og en mand med værktøj i hænderne stod foran det. West tænkte, at det nok var en af DEC's teknikere, som var i gang med at installere maskinen.
>
> West's hensigter var ikke ulovlige, men de tålte på den anden side heller ikke nogen nærmere efterforskning, og han

ønskede på ingen måde at bringe den ven, der havde givet ham tilladelse til at opsøge dette lokale, i en vanskelig situation. Hvis teknikeren havde bedt West om at legitimere sig, ville han ikke have løjet, men han ville heller ikke have besvaret spørgsmålet. Men det problem opstod ikke, for teknikeren spurgte ham ikke om noget. West stod og betragtede hans arbejde, og kort efter pakkede teknikeren sit værktøj sammen og gik.

Så lukkede West døren, gik gennem lokalet tilbage til datamaten, som nu næsten var helt samlet, og begyndte at skille den ad.

Det kabinet, han åbnede, indeholdt VAX'ens centralenhed, den såkaldte CPU, – den fysiske maskines hjerte. I VAX bestod denne ædle del af 27 printplader, der var placeret som bøger på en hylde. West tilbragte det meste af formiddagen med at trække pladerne ud, han ville undersøge hver enkelt plade og derefter sætte dem på plads igen.

Han undersøgte VAX'ens chips udenpå – nogle af dem var forsynet med numre, som han kendte i forvejen – og talte, hvor mange der var af hver enkelt type. Bagefter så han på andre dele af maskinen. Også her bemærkede han, hvilke typer af komponenter der var brugt, og talte dem op.

Og da han var færdig med dette, lagde han det hele sammen og beregnede, at det sandsynligvis kostede 22.000 dollars at fremstille det essentielle maskinel i en VAX (hvilket DEC solgte for lidt mere end 100.000 dollars).

Han forlod maskinen i præcis samme tilstand, som han havde fundet den.

"Jeg havde levet i frygt for VAX et helt år", sagde West, da vi en aften kørte ned ad motorvej 495. "Jeg var egentlig ikke ude på noget kriminelt. VAX var ikke beskyttet af patentretten, og jeg ville se, hvor stor skade, der var sket (i forhold til Data Generals konkurrenceevne). Jeg tror, jeg blev høj da jeg så på den og opdagede, hvor kompliceret og kostbar den var. Det fik mig til at føle, at nogle af de beslutninger, vi har truffet, er rigtige."

DeCastro mente, at den egentlige forskning skulle foregå i North Carolina. Man havde ikke ressourcer nok til også at satse fuldt ud på et lignende projekt; og der var heller ikke nogen i virksomheden, som for alvor troede på, at man kunne lave et ordentligt produkt i kælderen i Westborough, men på den anden side skadede det jo ikke med lidt konkurrence mellem to hold. Det er sådan noget, som holder folk til ilden. Så projektet "i kælderen" kommer i gang.

En af de mere gyselige detaljer er beskrivelsen af, hvordan West og hans kollega Alsing opbygger et team, som skal være mere end begejstrede for arbejdet. De unge uerfarne kaldes "børn"; de ved ikke, hvor hårdt det kan slide på venner, familie og den almindelige sindstilstand.

Alle cheferne erkender, at det er lidt groft at udnytte børnearbejdskraft på den måde, men det er nødvendigt for at gennemføre projektet.

Man forsøger så til gengæld at give de unge mennesker en oplevelse af, at de har været med til noget betydningsfuldt; der er meget prestige forbundet med at fuldføre et projekt som dette, set fra et ingeniørmæssigt synspunkt.

I en vis forstand er det en lige så stor præstation at udforme projektet, således at unge, begavede og veluddannede ingeniører vælger ansættelse i gruppen i "kælderen", især fordi efterspørgselen efter unge edb-specialister var langt større end udbudet.

En af de besynderligste beskrivelser i bogen er en beskrivelse af, hvordan maskinen udfører en instruktion, altså en "assembler" instruktion, maskinens lav-niveau indbyggede kommandoer, som fx. addér 2 til indholdet i register A.

Et register er så at sige maskinens "tælleværker", altså et sted, hvor tallene opbevares

mens de bearbejdes, og inden de gemmes i hukommelsen eller på disk/diskette o.lign.

En assembler instruktion er altså den mindste kommando, som det er relevant for en programmør at interessere sig for, men når man bygger computere, skal der jo ske adskillige ting før og efter instruktionen kan udføres; men også selve additionen (eller hvad det nu er) kan have flere trin, som udføres ved hjælp af microcode. Der er m.a.o. en "computer i computeren", og derved kan man ved hjælp af ganske simple instruktioner sammensætte mere nyttige operationer, fx. addition og division. Her kommer beskrivelsen af en nedstigning i computerens indre.

Blau boede i et boligkompleks nær Natick i Massachussetts. Hans værelser var nydeligt, men sparsomt møbleret: der var en gammel divan, en sækkestol og en velassorteret bogreol – hans intellektuelle føde syntes at bestå af science fiction, romaner, der havde fået gode anmeldelser, filosofi, fysik og matematik. En cykel med ti gear stod lænet op ad væggen i soveværelset. Hans seng var en madras, der lå på gulvet. Han undskyldte det tomme køleskab. Det var en lejlighed, som tilhørte en ung mand, der var på vej et eller andet sted hen, og det fik mig til at føle mig gammel.

Sammen med Blau steg jeg så at sige ned i Eagles maskinrum. Han pegede på Eagles hoveddele og omtalte dem, som om de var levende væsener, der stillede spørgsmål, ledte efter svar og sendte og modtog meddelelser. Nogle mennesker kan blive nervøse, når man taler om maskiner på denne måde, men det var en af de måder Blau anskuede maskinen på, mens han arbejdede på den.

Blau forklarede, at maskinen for at udføre en ordre i et brugerprogram først er nødt til at gøre en masse andet arbejde. Lad os antage, at et særligt program – vi kan for eksempel kalde det programmernes program – allerede kører.

Dette program kommunikerer med personer, der bruger datamaten, og det udfører opgaver for dem. Det planlægger og kontrollerer alle de brugerprogrammer, der kører i et givet øjeblik. Det finder også et program til brugeren.

Dette programmernes program indeholder inddata- og uddata-assemblerordrer, som fortæller inddata/uddata-kontrolenheden – den såkaldte IOC – hvordan informationerne skal sendes frem og tilbage mellem maskinen og brugerterminalen. (IOCen gør det muligt for datamaten at kommunikere med den ydre verden, groft sagt har den samme funktion som en oversætter, den behersker mange sprog. Den kan klare såvel "transporten" af informationer ved høj hastighed som kommunikation med brugerudstyr, der arbejder ved relativ lav hastighed. Dét er kompliceret.)

Godt, sagde Blau. Lad os sige, at en eller anden ude ved en terminal, en bruger, ønsker at køre et program; vi kan for eksempel kalde det program "FOOBAR". Via en terminal fortæller denne bruger programmernes program: "Kør program FOOBAR". Programmernes program fortæller IOC, at den skal flytte en del af dette program fra et pladelager uden for CPUen og ind i CPUens "hovedlager", og at den skal gøre dette med stor hastighed. Når dette er effektueret, overlader programmernes program kontrollen til program FOOBARs ordre, den ene efter den anden.

Før maskinen kan udføre en ordre, er den naturligvis nødt til at finde den. Derefter skal den hente den ind og afkode den. Således nåede vi frem til en printplade, der kaldes ordreprocessoren eller IP, Instruction Processor, og det var en temmelig kompliceret indretning.

IPen er selv kun i besiddelse af et ret begrænset lager. På en vis måde kan man sige, at den gør overslag over, hvilke ordrer i brugerprogrammet der vil blive de næste, og disse holder den så parat i sit lager.

På grundlag af disse overslag finder, henter og afkoder IPen ordrer på forhånd. Derfor kaldes IPen også en accelerator; den udfører arbejde, som skal udføres i alle datamater, men den gør det på forskud.

Men lad os for eksempel antage, sagde Blau, at program FOOBAR har kørt et stykke tid, og IPen opdager – hvilket nu og da sker – at den ikke har den næste ordre i sit eget lager. Hvis dette sker, sender den en meddelelse til adresseoversætterenheden -- ATU (Address Translation Unit) -- som blandt meget andet indeholder et kort over CPU'ens hovedlager. I denne meddelelse spørger IPen ATUen, hvor den næste blok af ordrer i brugerprogrammet FOOBAR er lokaliseret.

Vi kan nu forestille os, sagde Blau, at ATUen finder denne næste blok på sit kort. Dette betyder, at de ønskede ordrer befinder sig i hovedlageret. ATUen ved hvor og sender nyheden videre til IPen. Derpå sender IPen meddelelse til en anden accelerator, systemlageret, og beder om at få disse ordrer.

Måske er systemlageret i besiddelse af denne næste blok af ordrer. Hvis dette er tilfældet, behøver den ikke at få dem udleveret andre steder fra, men kan sende dem direkte til IPen, hvorved der spares tid. Men hvis systemlageret ikke kan finde disse ordrer hos sig selv, henter det dem frem fra hovedlageret og viderebringer dem til IPen.

"Men", sagde Blau, "det kan tænkes, at denne næste blok af instruktioner heller ikke befinder sig i hovedlageret." Andre personer bruger datamaten, kører andre programmer. Nok er hovedlageret stort, men det er ikke stort nok til at kunne indeholde alle dele af alle disse programmer på samme tid.

Almindeligvis indeholder det kun nogle dele af hvert program. Så lad os antage, sagde Blau, at den næste afdeling af program FOOBAR befinder sig uden for CPUen, at den stadig ligger i et ydre lager, på en plade. I dette tilfælde vil ATUen ikke kunne finde adressen på den næste blok af ordrer på sit kort. Derfor sender ATUen en meddelelse til *microsequenceren.* Denne besvarer ATUens meddelelse ved at udsende et bestemt microprogram[1], som overgiver systemprogrammellet kontrollen med eftersøgningen. Systemprogrammellet beordrer derpå noget, der kaldes et "fejlsøgningsprogram" [2] kaldt frem; dette indeholder mange assemblerordrer og som følge heraf mange mikroprogrammer. Dette fejlsøgningsprogram skulle kunne fortælle maskinen, hvordan den skal bære sig ad med at finde den næste blok af ordrer i program FOOBAR, og hvordan den fører dem ind i lagersystemet.

Det burde ikke kunne ske – hvis det gjorde, sagde Blau, ville det være ensbetydende med, at der var sket en fejl ved udarbejdelsen af programmellet, – men lad os forestille os, at ordrerne til fejlsøgningsprogrammet ikke befinder sig i lagersystemet, men i stedet ligger et eller andet sted på en plade. For at gøre det muligt at hente ordrerne til fejlsøgningsprogrammet på pladen, skulle Eagle selv udføre et fejlsøgningsprogram, men den ville ikke være i stand til at udføre et fejlsøgningsprogram til at få fat i ordrerne, før den allerede var i besiddelse af dem. Blau sagde, at situationen ville være den samme, som hvis man låste et skab og lod nøglen ligge inden i det. "Hvis dette sker, vil maskinen begynde at spejle sig selv i en uendelig række spejle."

Fejl af denne type er så berygtede, at de nu næsten er et ukendt fænomen; men for at være på den sikre side tog Eagles konstruktører alligevel højde for dem. Via styrepanelet står Eagle i forbindelse med en mikrodatamat, en af Data Generals almindelige mikronovaer, der til dels fungerer som Eagles terapeut. Hvis der går fisk i Eagle, er det meningen, at mikronovaen stadig skal fungere; den kan køre fejldiagnoseprogrammer og derved finde ud af, hvad der er galt med den store maskine, selv om denne er død. Mikronovaen "overgramser" også konstant den store maskine for hele tiden at have føling med, om der opstår dødsensfarlige situationer som for eksempel en uendelig regres af fejl. Hvis den opdager en, vil styrepanelet

[1] ved at starte et interrupt, som man ville sige i dag for en I386'er, der også kan reagere på denne måde hvis den ønskede instruktion eller data-byte ikke kan findes gennem adressetabellerne.

[2] Kaldes også Interrupthandler, Interrupt Service Routine.

blive alarmeret, og dette vil sende en meddelelse tilbage til mikronovaen, som igen vil sende en advarsel til konsollen, den store skrivemaskine, som er placeret ved siden af Eagles CPU. Denne maskine vil da udskrive følgende tekst:

JEG ER GÅET I SPIN. CPU STANDSET.

Hvis man er operatør på dette system og ser denne meddelelse, sørger man omgående for at få en direkte samtale igennem til Data General.
Men lad os nu gå ud fra, at intet af dette skete, sagde Blau. Lad os antage, at den næste ordre-blok, der skal bruges i programmet FOOBAR, let lader sig spore. Systemlageret sender blokken videre til IPen. Denne kaster en anden blok af informationer bort for at gøre plads til denne, og så er eftersøgningen forbi. Eagle kan begynde at udføre den næste ordre i program FOOBAR.

IPen undersøger den indkodede ordre og identificerer den som en "Skip On Equal", hop hvis lighedstegn, – en af de ordrer, der fortæller datamaten, at den skal sammenligne to værdier og, afhængig af udfaldet, følge en af to veje.
I assembler skrives dette som "WSEQ" (hvilket, hvis det udtales, nærmest lyder som en rusten havelåge, der åbnes).

WSEQ giver maskinen instruks om, at den skal sammenligne to værdier, og hvis de er ens springe den næste assemblerordre i programmet FOOBAR over og gå videre til den ordre, som kommer efter den oversprungne[3]. Antag, sagde Blau, at de to værdier – naturligvis i elektrisk kode – har fundet vej ind i regneenheden eller ALUen (Arithmetic Logic Unit – somme tider også kaldt "talknuseren"), hjertet i enhver datamat.

IPen har ud fra WSEQ-ordren bestemt flere forskellige informationer. Den vigtigste af disse er adressen på det mikroprogram, der kan fortælle Eagle præcis, hvad den skal gøre i Skip On Equal-situationen.

Microsequenceren udbeder sig adressen på det næste mikroprogram. IPen sender nu denne adresse til sequenceren, hvorefter denne begynder at køre mikroprogrammet. IPen har på dette tidspunkt også udledt, på hvilken plads de to værdier, som skal sammenlignes, befinder sig. Disse pakker befinder sig allerede i ALUen. IPen fortæller ALUen præcis, hvor de er.

"Nu bevæger vi os ned på et lavere abstraktionsniveau", sagde Blau. "Vi skal derned, hvor mikrokoden befinder sig."

Der er et ur inden i Eagle. Det tikker, hver gang der er gået 220 milliarddele af et sekund. Mellem hvert tik udfører Eagle en mikroordre.

»Tik«, sagde Blau.

Sequenceren udsender den første af WSEQ-mikroprogrammets mikroordrer.
Mikroordren består af 75 bit, 75 enheder af høje og lave elektriske spændinger. De spredes ud i kredsløbene. Nogle går til ALUen og fortæller den, at den skal trække den ene af de to værdier fra den anden. Nogle af disse mikrobit går til IPen, andre til ATUen og IOCen, og atter andre går direkte tilbage til sequenceren og giver den adressen på den næste mikroordre i WSEQ-mikroprogrammet.

»Tik«.

Sequenceren udsender den næste mikroordre. En del af den går til ALUen, hvor den fortæller denne, at den skal undersøge resultatet af den netop foretagne subtraktion. Hvis resultatet af subtraktionen ikke er nul, – hvis med andre ord de to værdier ikke er identiske – sender ALUen en lav elektrisk spænding gennem en bestemt ledning. Mikroordren har fortalt *microsequenceren,* at den skal overvåge denne ledning. Hvis *sequenceren* opdager en lav spænding i ledningen, bringer den WSEQ-programmet til afslutning. Den sender en meddelelse til IPen, som i alt væsentligt lyder: "Ikke lige store, ingen

[3] En sådan skip instruktion er meget økonomisk, idet den ikke har en adresse, der skal hoppes til.
I Intels processorer er hop instruktioner altid forsynet med en adresse hvorhen, som dog kan være på kun én byte. Den betyder: hop så og så mange bytes frem fra det sted, hvor vi er nu.

Skip On Equal denne gang – hent den næste assemblerordre i brugerprogrammet."

Men hvis resultatet af ALUens subtraktion er nul – hvis de to værdier *er* lige store – så sender ALUen en høj elektrisk spænding gennem den særlige ledning, som *microsequenceren* overvåger. *Sequenceren* tolker dette og udsender den tredje og sidste mikroordre i WSEQ-mikroprogrammet.

»Tik«.

Så går den sidste linie mikrokode ud og fortæller alle printpladerne – med undtagelse af IPen – at de skal vente. IPen får at vide, at den skal springe den næste assemblerordre i brugerprogrammet over, men genoptage operationerne, når den næste ordre efter den oversprungne kommer. Det er på denne måde, Eagle hopper over en korsvej i et program og sætter retning ned ad en anden vej.

West havde sagt, at det var et "tankespil" at konstruere en datamat. Jeg spurgte Blau, om det var samme form for tankespil, de spillede under konstruktionen af Eagle, som vi havde praktiseret ved at følge en ordre rundt i maskinen. "Det kan du lige tro!" sagde han. Men de spillede hundreder af sådanne tankespil; og blot det at beregne, hvordan Eagle skulle indrettes med hensyn til WSEQ, var i sig selv så kompliceret, at Blau ikke kunne forklare det fyldestgørende for mig på en enkelt aften.

Sådanne logiske spil kan, især hvis de spilles under tidspres – mens man flyver med hovedet nedad – helt besætte edb-specialistens tankeverden. Efter at have spillet på denne måde et stykke tid ser man på et træ, og, aha, pludselig står det helt klart, at et træ har samme struktur som en datamat; og en hovedgade med sidegader – hvad er det andet end en form for dataprogram? Chuck Holland sagde, at denne ubehagelige oplevelse af at være låst fast inden i maskinen almindeligvis holdt sig i tre dage – ved de sjældne lejligheder, hvor han var borte fra kælderen så længe ad gangen.

Er det Computerarkæologi at interessere sig for sådan en beskrivelse? Nejda. Det er en ganske udmærket måde at levendegøre de principper, som gør det muligt at bygge hurtige maskiner.

Og ærligt talt, der er også nogle menneskelige aspekter, som man kan blive klog på. Tracy Kidder forsøger ikke kun at give os et indblik, men også et overblik. Og det er meget, meget sundt.

At forsøge at se sig selv "lidt udefra", når man arbejder intenst med et spændende program eller hvadsomhelst, som optager én dybt, det kan måske give en smule ubehag i starten "uha, er jeg sådan en kedelig fyr, der sidder med næsen i skærmen og tankerne fastlåst på denne lille bitte detalje af den store verden..."; men når man så er kommet over dét stadium, kan det hænde, at man får inspiration og større arbejdsglæde ved at se tingene på en lidt anden måde.

Det er interessant at se en så særpræget beskrivelse af forudhentning af instruktioner, noget som allerede 8-bits Intel8085 kunne gøre.

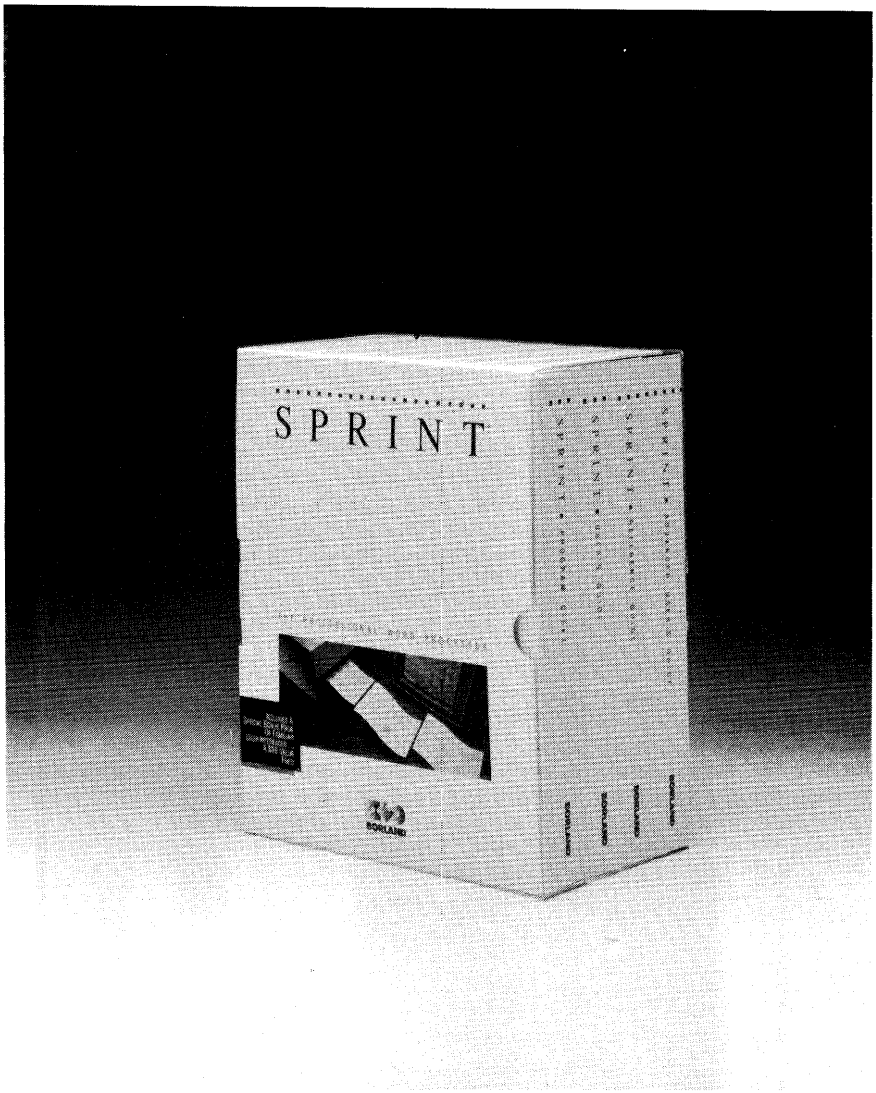
Og beskrivelsen af en hukommelses-husholdning, "memory management unit" (ATU) er også god at blive klog på, der er mange ting, som svarer til det, en 386'er er i stand til at gøre.

Hvis der er nogle afsnit i ovenstående, som virker lidt tågede, så er det nok fordi ideen med denne komplicerede maskine faktisk er meget sammensatte, ud over forudhentning af instruktioner er det jo en beskrivelse af 1) virtuelt lager, 2) interrupts og endelig 3) MICROCODE:

Microsequenceren er så at sige en computer i computeren, en kontrollant, der udsender ordrer til de forskellige enheder, som CPUen er sammensat af, for at få dem til at arbejde sammen.

Ved hjælp af en microsequencer kan man opbygge komplicerede instruktioner, som på de tidligste CPUer svarede til længere programstumper. Derved spares der på det antal bytes, som programmet på disk eller diskette består af, men til gengæld bliver de enkelte instruktioner lidt langsommere. En microsequencer er "intern"

forts. side 20



(se anmeldelse i nr. 2, 1993, side 18)
SPRINT Editoren koster kr. 281 incl. moms & forsendelse, og kan bestilles hos:

BORLAND SCANDINAVIA A/S

BOKS 236 • Gydevang 12 • 3450 Allerød

Telefon 42 27 64 55 • Telefax 42 27 16 77 • Giro 7 29 51 03

HVAD KOSTER OBJEKT ORIENTERET PROGRAMMERING ?

OBJECTSW.393

Objekt orienteret programmering er lidt mere ressourcekrævende, ikke noget, som betyder noget i et stort projekt, men nok til, at det kan berettige lidt opmærksomhed.

Hvis man inkluderer alt det, som normalt "hører med", kan de kompilerede, færdige programmer blive lidt mere omfattende, end de behøver at være. Der kan med andre ord komme noget ekstra bagage med, som ikke tjener noget formål.
For at illustrere forskellen kompilerede jeg et meget simpelt program, som kan trække på include filerne til Turbo C++ eller lade være. Programmet fylder ca. 18 kb med C++ streams m.v. includerede, men kun 4.5 kb uden. Det er uvæsentligt for store projekter og store programmer, men kan have betydning for små hjælpeprogrammer.

```
PROGRAM1.EXE     18620
PROGRAM1.OBJ       568
PROGRAM2.EXE      4692
PROGRAM2.OBJ       392
```

Som man kan se, er program1 væsentlig større end program2, selv om de foretager sig det samme. Der er kun én forskel, nemlig den, at program2 ikke inkluderer iostream.h

Programmets kerne er udskrift af 3 linier, som fortæller med hvilke definitioner, det er kompileret. Det er nok noget man sjældent vil gøre, men det kan ind i mellem være nødvendigt at foretage en sådan test, hvis man skal finde ud af, hvordan compileren opfører sig.

```
1    #include "iostream.h"     /* dette er med i program 1 men ikke i 2.
2                               /* og det behøver altså ikke at være med.
3    #ifndef __STDIO_H
4    #define __STDIO_H

5    /* her indsættes eventuelle nødvendige definitioner.  */
6    /* i dette program er der kun brug for én definition. */
7    extern "C" {
8    int cdecl puts(const char *__s);
9    }

10   #define test "Der blev anvendt lokale io definitioner."
11   #else
12   #define test "Der blev ikke anvendt lokale io definitioner"
13   #endif

14   #ifdef _Cdecl
15   #define t2 "Cdeclarationer er defineret"
16   #else
17   #define t2 "Cdeclarationer er ikke defineret"
18   #endif

19   #ifdef __STDC__
20   #define t3 "Standard C definitioner benyttes"
21   #else
22   #define t3 "Standard C definitioner benyttes ikke"
23   #endif

24   main()
25   {
26     puts(test);              /* bliver lokale definitioner benyttet? */
27     puts(t2);                /* bliver cdeclarations benyttet? */
28     puts(t3);                /* bliver stdc benyttet? */
29     return 0;
30   }
```

Forskellen er, at der includeres en masse iostream funktioner hvis man medtager iostream.h; deri erklæres (eller deklareres) cin, cout, cerr og clog, 4 objekter, som kan skrive de

mest almindelige dataformater til skærm eller fil efter at have konverteret dem til læselig tekst.

For mange er dette en væsentlig del af C++ programmeringen, men det er ikke nødvendigt at includere og benytte disse stream input/output funktioner. Det kan imidlertid meget vel tænkes, at man benytter et andet IO library, hvorved disse cin/cout bliver overflødige. Hvis man så alligevel medtager iostream.h, vil programmerne blive fyldt op med kode til disse stream funktioner (eller objekter).

ET RASK LILLE SIDESPRING

Dette er imidlertid ikke det eneste, som demonstreres af ovennævnte eksempel. Det fortæller, når det er kompileret, om nøgleordene __STDIO_H, _Cdecl og __STDC__ er defineret. At de er defineret vil sige, at der har været en linie med

```
#define __STDC__
```

et eller andet sted i programmet, inden det sted, hvor man med #ifdef tester på, om det er defineret. At det er defineret til en tom linie er i denne sammenhæng underordnet, men ordet står i compilerens tabel over definerede ord.

Det første ord defineres, hvis headerfilen stdio.h er inkluderet. Det er meget praktisk, at compileren gør det, det sparer lidt tid. Ofte er en includefil nemlig nævnt flere steder i headerfilerne, og hvis denne har været tygget igennem én gang sparer man compileren for at gentage dette.

Det andet ord defineres for at compileren, når den kører som "gammeldags" C compiler, kan finde ud af at "annullere" nøgleordet cdecl. Det findes i masser af funktions - deklarationer, og Turbo C++ forventer det, hvis den "får lov" til at benytte sine "extensions", d.v.s. hvis den skal foretage syntax check på funktions prototype deklarationer. Hvis man forlanger af compileren, at den **skal** benytte standard C, "gammeldags C", så vil den protestere mod nøgleordet cdecl i funktions prototyper. Hvis man ikke forlanger dette, og den støder på ordet cdecl, så ***afdefinerer*** den straks ordet __STDC__ og vil altså forvente sin egen, nyere ANSI C inspirerede syntax. Jaha, den kan også "afdefinere", *undefine*, og det kan man med tcc også gøre fra kommandolinie med -U optionen.
Fx. kan man skrive -U__STDC__, hvis man er i tvivl om hvilken funktion det har.

Der er noget mere i det, nemlig en erklæring af en extern "C" funktion. Dette "C" er et nøgleord for Turbo C++, og det fortæller, at alt i den efterfølgende blok skal opfattes som "C" library funktioner.

Man kan sagtens definere sin puts uden først at skrive extern "C". Compileren vil i så fald addere en kode for, at dette er en af brugeren defineret funktion, så man ikke ved en fejltagelse kommer til at omdefinere en library funktion.

Med et object listing program, som fx. tdump.exe, kan man se, at den externe funktion i .obj filen er defineret således:

```
- - -
a2 EXTDEF
    1:  "@puts$qpxzc"  <--OBS HER!!!
b3 PUBDEF  GroupIx=0  SegIx=1  "_main"  Offset=0
c2 LEDATA  SegIX = 1  Offset = 0  BufLen = 1e
- - - osv.
```

(Mine fremhævelser. Det kan godt være svært at se, hvad der er hvad i en større .obj file listning.)

Der er altså tilføjet noget inden puts og efter puts; compileren fortæller derved sin efterfølger, linkeren, at dette er en funktion, som brugeren har defineret, ikke en, der skal findes i standard library. Derfor vil linkeren, tlink, som danner det færdige program, ikke kunne finde funktionen `@puts$qpxzc` i `c*.lib`, og følgelig vil programmet ikke kunne fungere. Man **skal** altså skrive:

```
extern "C"  {
int cdecl puts(const char *__s);
}
```

hvis man vil have fornøjelsen af **selv** at erklære en standard library function i TurboC++.

Man får med andre ord nemt problemer med sine C++ programmer, hvis man benytter funktioner fra bibliotekerne, men ikke de tilhørende include filer. Det tilrådes derfor, at man benytter producentens includefiler, hvor de er nødvendige; på den anden side er det jo ikke forbudt at sætte sig ind i, hvornår de med fordel kan undværes.

ET EKSEMPEL PÅ PRISEN PÅ OBJEKTER

For at se nøjagtigt hvor meget anvendelse af objekter koster, vises her et program i 2 udgaver, en almindelig C - version og en C++ version. Programmet beregner højden af en flagstang, når man indtaster afstand og vinkel; flagstangen formodes at være vinkelret på jordoverfladen, eller lad os sige det sådan, at målingen formodes at benytte en vandret grundlinie. Hvis der ikke er tale om en flagstang eller trætop, bliver problemet med at måle afstanden til flagstangens fod af helt andre dimensioner.
Matematikken er simpel og velkendt for gamle 4- mellem elever, spejdere m.fl.; højden er lig med tangens til vinkelen gange med afstanden.
Programmerne fylder:

`FLAG1V0.EXE`	`24494`	almindelig C version
`FLAG1V1.EXE`	`24590`	Turbo C++ version.

Det er kun 100 bytes, vi betaler for den ekstra C++ service, og det må siges at være en beskeden pris. Bemærk imidlertid, at begge programmer er uden C++ streams, som ellers ville have kostet en del kilobyte mere. Det, der fylder mest, er floating point funktionerne.

I en version med `#include <iostreams.h>` kom programmet til at hedde flag1v2, og fyldte 38kb, altså stadig ca. 14kb ekstra, i dette tilfælde uden nogen anvendelse.

Til slut skal det lige siges, at én af de væsentlige fordele ved cin og cout er deres evne til "selv" at finde ud af, hvilket format de data har, som man skal skrive til skærmen, derved kan man somme tider spare megen tid på fejlfinding.

Artiklen her skal på ingen måde forstås som en generel advarsel mod at benytte C++ iostream.h, med cin og cout << float << tekst << integer osv., det er kun et forsigtigt skøn over udgifterne derved.

Her kommer programmerne i kildetekst med kommentarer:

```
1    /* Program FLAG1V0.C, almindelig C syntax for en simpel beregning */

2    #include <stdio.h>
3    #include <math.h>
4    /* hvis vi ikke includerer math h, får vi alvorlige fejl, fordi
5      compileren tror, at tan() returnerer et heltal, en integer,
6      MEN den returnerer en dobbelt længde floating point værdi. */

7    char *progname;

8    void usage(int erro) {
9     printf("Programmet %s anvendes ved at indtaste afstand og vinkel\n", progname);
10    printf("Fx. %s 12m 45°, hvor 0<grader<89\n",progname);
11    if (erro==1)
12      printf("Fejl i antal parametre\n");
13    if (erro==2)
14      printf("Fejl i første parameter\n");
15    if (erro==3)
16      printf("Fejl i andet parameter\n");
17    if (erro==4)
18      printf("Fejl i grad angivelse, skal være mellem 0 og 89.99\n");
19    exit(erro);
20   }
```

```
21  main (argc,argv)
22  int argc;
23  char **argv;
24  {
25   double afstand,vinkel,rvinkel;
26   char enhed[80];
27   progname=argv[0];                    /* en pointer til navnet
28                                           på dette program, sådan
29                                           som det er kaldt fra
30                                           kommandolinie eller billedshell */

31   if (argc != 3) usage(1);
32   if ((sscanf(argv[1]," %lf%s",&afstand,enhed))!=2)
33     usage(2);
34   if ((sscanf(argv[2]," %lf%s",&vinkel,enhed))!=2)
35     usage(3);
36        /* tangens til 90 grader er ikke defineret,*/
37        /* vi må sikre os mod tåbelige fejl */

38   if (vinkel>=89.99) usage(4);
39   rvinkel=vinkel/360 * 2 * M_PI;       /* function tan() forventer at
40                                           vinkelen er udtrykt i radianer,
41                                           som er buelængden for vinkelen
42                                           i en cirkel med radius 1.    */

43   printf("Højde = %lf\n",tan(rvinkel)*afstand);
44   return 0;
45  }
```

Nu det samme program i en lidt mere objekt-orienteret version:

```
2   #include <stdio.h>
3   #include <math.h>
4   // hvis vi ikke includerer math, får vi masser af kompileringsfejl,
5   // vi bliver advaret om, at funktionen tan() ikke er "declared"/erklæret

6   char *progname;

7   extern "C" {
8    void cdecl exit(int i);        // man kunne includere dos.h i stedet
9   }

10  void usage(int erro) {
11   printf("Programmet %s anvendes ved at indtaste afstand og vinkel\n",progname);
12   printf("Fx. %s 12m 45°, hvor 0<grader<89\n",progname);
13   if (erro==1)
14     printf("Fejl i antal parametre\n");
15   if (erro==2)
16     printf("Fejl i første parameter\n");
17   if (erro==3)
18     printf("Fejl i andet parameter\n");
19   if (erro==4)
20     printf("Fejl i grad angivelse, skal være mellem 0 og 89.99\n");
21   exit(erro);
22  }

23  class skal_maales {
24   double vinkel,afstand;         // disse variable kan KUN
25                                  // berøres gennem funktionerne
26                                  // som erklæres i public-delen:
27   public:
28   void set_vinkel(double grader);
29   void set_afstand(double afstand);
```

```
30    double hoyde(void);
31   } dims;

32   void skal_maales::set_vinkel(double grader) {
33    if (grader>=89.99)                  // fejl-behandling 4 er flyttet
34                                        // ind i objektet.
35      usage(4);
36    vinkel = grader;                    // vinkel er objektets variabel.
37   }

38   void skal_maales::set_afstand(double m) {
39    afstand = m;                        // afstand er objektets variabel.
40   }

41   double skal_maales::hoyde(void) {
42    return tan(vinkel/360 * 2 * M_PI) * afstand;
43   }


44   main (int argc, char **argv) {
45    double afstand,vinkel;
46    char enhed[80];
47    class skal_maales hustag;           // eller flagstang, trætop, dog husk:
48                                        // ikke ææå i variabelnavne
49    progname=argv[0];                   // pointer til programmets kaldenavn
50    if (argc != 3) usage(1);            // vi skal have 2 parametre

51    if ((sscanf(argv[1]," %lf%s",&afstand,enhed))!=2)
52      usage(2);
53    hustag.set_afstand(afstand);        // vi sætter afstand ved at
54                                        // sende et message til vores objekt,
55                                        // her sendes 'besked' til hustaget
56    if ((sscanf(argv[2]," %lf%s",&vinkel,enhed))!=2)
57      usage(3);
58    hustag.set_vinkel(vinkel);          // her sendes 'besked' til
59                                        // hustag om vinkelen

60    printf
61    ("Højde = %lf\n",hustag.hoyde());   // vi beder "hustag" om at
62                                        // fortælle sin højde
63    return 0;
64   }

65                                        // end of program kildetxt.
```

Bemærk, at der er variable i MAIN funktionen, som hedder afstand og vinkel, det samme som objektets "beskyttede" variable, men det gør ikke noget, da man ikke uden videre kan få fat i de private variable, hustag.afstand, hustag.vinkel.
Hvis man ellers har fantasi til det, er det en udmærket idé at benytte forskellige navne, for ikke selv at rode rundt imellem dem.

Program inputsiden er svag, man kan let rode rundt mellem afstanden og vinkelen.
Programmet kan imidlertid bringes til at kontrollere om tallene er angivet med rigtig "enhed".

Kontrollér programmernes funktion ved at give vinkelen 45 grader, derved skal højden være den samme som afstanden.

D. A.

i CPUen og kan derfor køres meget hurtigere end et RAM-program. Det er imidlertid langsommere end "hardcodede" instruktioner, d.v.s. instruktioner, hvor hver enkelt bit i instruktionskoden udløser en begivenhed i en bestemt del af CPUen.

Virtuel hukommelse skal for at fungere bestå af både hardware og software, som skal være skræddersyet til hinanden. Det er kort sagt en måde at bruge RAM på, at hvert enkelt program tror, at det har en stor, tom maskine til rådighed. Hvis der ikke er nok plads på de RAM-chips, man har købt, kan harddisken bruges om ekstra plads.

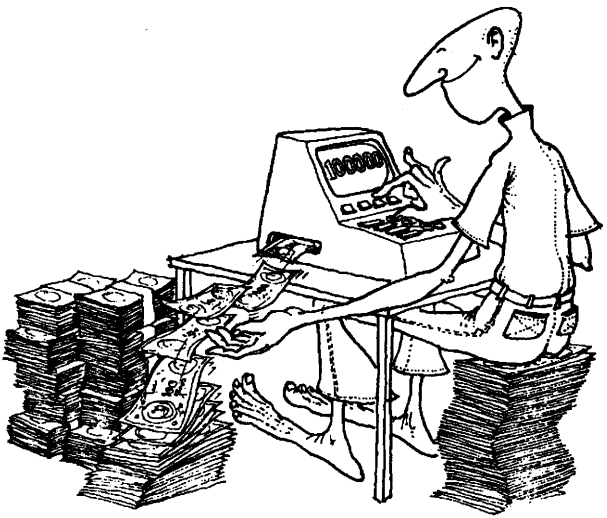
For nogle måneder siden bad en kollega mig om råd til at køre windows 3.0 på en 386 maskine med "kun" 1 MB RAM, og jeg måtte så til at læse nærmere om dette systems opsætningsmuligheder. Ud fra den viden, jeg har fra læsning af forskellige beskrivelser af edb-maskiners opbygning, som fx. den ovenstående, skønnede jeg straks, at det eneste acceptable var at køre programmet Windows 3.0 i 386 virtuel memory mode, hvor en disk fil, helst stor og i en enkelt blok, benyttes af systemet som ekstra memory, derved ville alle større programmer kunne få plads nok; selv om det gik langsomt, ville det virke, og der ville ikke kunne komme meddelelser om system failure, fatal error osv.

Det virkede, efter et par timer kunne vi køre en kæmpe LOTUS demonstration med masser af grafik og "levende billeder".

Det var jo langt fra den verden, som Tracy Kidder beskriver i bogen "MASKINEN", men principperne for anvendelse af hukommelse i Windows 3 er altså ikke så forskellige fra de principper, Jon Blau, West og alle de andre benyttede ved konstruktionen af MASKINEN i 1979.

D. A.

Ref.: John Tracy Kidder:
The Soul of the New Machine ©1981, oversat 1983 af Torben Nilsson. ISBN 87-17-05108-8



NY BOXER EDITOR VERSION 5A

BXR5.393

Programmør-Editor for DOS.
(version 3.03 omtalt i MUG 1993:1, er på disk: MEDL9303)

Hvad der under Windows er blevet næsten en selvfølge finder man under DOS kun efter lang tids søgen:
En programmør-Editor med farver til syntax-fremhævelse, understøttelse af flere compilere, forskellige tastatur-opsætninger, makroer...

Editorer findes - og ikke bare i Shareware - som sandskorn ved havet. Denne mangfoldighed af editorer har selvfølgelig hver deres særpræg og mange forskellige tilpasningsmuligheder, der er dog ikke mange der kan konfigureres så frit som Boxer.

Her har du f.eks. mulighed for at vælge et tastatur-layout, der svarer til det du er vant til d.v.s. arbejder du normalt med Borlands IDE, Word, WordPerfect eller en helt anden editor/-tekstbehandler kan du vælge en tastatur-udlægning, der svarer helt til det du allerede kender.

Således at du kan bruge de vante kommandoer til at markere blokke, slette en linie, loade en ny fil osv. Er det en af de gængse editorer/-tekstbehandlere du helst bruger er det bare at vælge blandt de medleverede tastatur definitioner ellers må du selv i gang med at definere det som du vil have det ved hjælp af den medfølgende RECONFIG utility.

Med Boxer kan man editere flere filer i forskellige vinduer. Eller hvis man har to næsten ens filer kan man loade dem i hvert sit vindue på skærmen og så aktivere 'synkron scrollning', således kan man synkront bladre igennem de to filer til man finder det/de sted(er), hvor de er forskellige. Der er også syntaksstyret belysning, det fungerer på den måde at indlæser man en fil med extension PAS, vil den blive behandlet som en PASCAL fil, og reserverede ord, konstanter og kommentarer fremhæves med forskellige farver.

Når man modtager Boxer er den allerede sat op med de nødvendige informationer hertil for programmeringssprog som f.eks. Turbo Pascal, C, ASM eller DOS-Batch. Finder man ikke udvalget tilstrækkeligt kan man selv tilføje nye eller man kan ændre i de definitioner, der allerede er tilstede. I denne mode er der selvfølgelig også en farve til parenteser, således at man nemt kan se om man har lige mange højre og venstre parenteser i et givet udtryk.

Er man vant til at arbejde med sin compiler på kommandolinie-niveau, vil Boxer også være et interessant bekendtskab idet den kan sættes op til, udfra fil extention (PAS, ASM osv.), at kalde den relevante compiler, vise compilerens fejlmeldinger og markere linier med fejl.

Der er også mulighed for markering både i hele linier og i spalter, Cut/Copy /Paste og en indtil 4096 gange! gennemførlig UnDo operation.

Boxer glimrer også med nogle små nyttige ting til hjælp i det daglige tastearbejde:
Heltals lommeregner, ASCII-tabel, Lineal samt et tegne-modul for blokgrafik. Det sidste giver mulighed for nemt at indramme et afsnit, lave en lille tabel eller lign.

Boxer genererer som de fleste andre editorer en backup fil når man gemmer en ny version af sin fil; men den gemmer alle backup filer i et separat directory, således at man kun skal rydde op i sine backup filer et sted! Og ikke som det normalt er tilfældet rundt omkring i alle de directories hvor man har ASCII filer man arbejder med.

Boxer kan sortere linier, kopiere vilkårlig ofte, ombytte, finde-og-erstatte (også over flere filer), behersker flere video-modi (bl.a. 28, 30 eller 50 linier) og giver On-line hjælp om alle options, endvidere kan den betjenes med mus.

Den husker også hvilken fil man sidst har arbejdet med, - så starter man Boxer uden at angive et filnavn vil den starte med den fil indlæst som man arbejdede med sidst man forlod editoren og med cursoren der hvor den stod da man gemte dokumentet.

Det er meget smart; men giver lidt problemer, hvis den sidste fil man arbejdede med blev læst fra en diskette og den rigtige diskette så ikke sidder i diskdrevet; men det får dog ikke editoren til at gå ned.

Boxer 5a skal bruge 384 Kbyte RAM og DOS 2.1 eller højere. Hvis man installerer og beholder alle filer der er med i 'Boxer pakken' fylder det omkring 750 kbyte på harddisken. Og det kan uden problemer skæres ned til omkring 500 kbyte, så man får en virkelig potent editor uden af den grund at fylde sin harddisk

---

Boxer er en gennemtænkt, hurtig og komfortabel editor. Selvom jeg ikke har nået at prøve alle de features der remses op ovenfor tør jeg godt anbefale den. Den har bestemt været en positiv oplevelse i den korte tid jeg har arbejdet med den. Om jeg ender med at bruge den i det lange løb tør jeg ikke sige, jeg er nok lidt konservativ mht. editor, så jeg har lidt svært ved at forlade den editor jeg normalt bruger både hjemme og på arbejde.

L.G.

Registrering: 35 $ + 6 $ til forsendelse. Forfatter er *David R. Hamel.*
Sender man 50 $ + 8 $ til Amerika får man en håndbog og en gratis Update.

Diskette med **Boxer 5a** kan efter aftale fås hos: Frank Damgaard.

Sælges: AD.393

Laser printer Canon LBP-8A1, 128 k ram, Courier10R + Landscape, DK manual, kan bruges som FAX. 1400 kr.

Matrix printer, Digital LA75, seriel u/manual, tastatur-indstill. 600 kr.

Matrix printer, DataSouth, DS-220, par. & seriel, programmérbar, IBM kompatibel, A3 & A4 format, ca. 300 tegn/s, US-manual. 500 kr.

Philips GP 300, seriel, aut. A3/A4, incl. tractor, (evt. arkføder) manual i låg, skriver flot. 300 kr.

Printer IBM Quietwriter III/5202, parallel, DK manual. 250 kr.

VT 220 Terminal kompl. m. tastatur, com-port, prn-port 9-pol 'D' 400 kr.

Nokia extern data modem DS 3648, 4800/2400 baud, 2&4-tråds, V27bis/V25/CCIT Rec v54, P&T godk. med manual, køreklar. 300 kr.

Vagn Nielsen. 3128 2154

ADRESSER, SOFTWARE & DISKETTER

CP/M-volumes bestilles ved CP/M-bibliotekaren.
Husk, ved bestilling af CP/M-volumes, at oplyse om diskformat!

PC-volumes bestilles ved PC-bibliotekaren.

Volume fra bibliotek (5.25") incl. disk & forsendelse 20,- kr.

Bestyrelsen:

Formand:

Donald Axel
Saxenkolvej 20
3210 Vejby
4870 6913

Frank Damgaard
Kastebjergvej 26A
2750 Ballerup
4497 3747

Anders Otte
Grønnevej 261, 13
2830 Virum
4285 1645

John B. Jacobsen
Lyshøj Allé 20, 3th.
2500 Valby
3116 1393
e-mail:
johbjbj@inet.uni-c.dk

Steen Weidner
Rådmandsg. 40-C, L-146
2200 København N.
3181 5753

Kasserer:

Lars Gråbæk
Esbern Snaresgade 6
1725 København V.
3123 9236

Vagn Nielsen
Klintevej 33
2700 Brønshøj
3128 2154

Viggo Jørgensen
Fensmarks Allé 6
3520 Farum

CP/M Bibliotek:

Frank Damgaard
Kastebjergvej 26A
2750 Ballerup
4497 3747
(man-tor 1730-1830)
Giro 1 92 80 66
e-mail: frank@diku.dk

PC-Bibliotek:

Peter Rasmussen
Roskildevænge 46-2tv.
4000 Roskilde.
(skriftligt)
Giro 7 499 140

Bulletin Board:

Telf. 3160 5319
Åbent hele døgnet
300 - 14400 bits/sec
V32bis, V42bis, MNP5
8bit 1 stop, ej paritet
SysOp: Vagn Nielsen

Disketteredaktør:

Redaktør: Viggo Jørgensen, FensmarksAlle 6, 3520 Farum, 42 95 32 01

M U G microcomputer-user-group

...en ikke-kommerciel forening for brugere af mikro-datamater, vore biblioteker understøtter IBM-PC og dermed kompatible mikro-datamater, samt CP/M.
Foreningen drives på frivillig basis og er rettet mod dem, der ønsker at få mere ud af deres computer end blot muligheden for at køre standard programmer.

Foreningen søger at støtte medlemmerne i brugen af deres computer ved arrangement af:

1. Medlemsmøder, hvor man kan mødes og snakke sammen, udveksle ideer, hente inspiration og få hjælp med problemer vedr. computere.

2. Fællesindkøb, hvorved vi kan opnå rabatter på komponenter, tidsskrifter, bøger, software, hardware etc.

3. Foredrag hvor folk, der ved mere end gennemsnittet om et emne, kommer og fortæller, så vi alle kan få udbytte af det.

4. Udsendelse af et aperiodisk nyhedsbrev, som udkommer på diskette i standard IBM format, med nyheder, tips, anmeldelser af bøger, soft- og hardware, kataloger fra foreningens software bibliotek samt diverse programmmer / shareware programmer.

Udgivelse af medlemsblad/hefte (almindeligvis 4-6 gange årligt.) med stof af forskellig art. Her kan medl. bringe artikler, små-nyt, spørgsmål, gratis (private) annoncer, osv.

Et bulletin board er til rådighed for medlemmerne, således at disse via modems kan udveksle meddelelser, programmer og få informationer, der stadig er "ovnvarme".

Foreningen hjemtager public domain/shareware og mod en lille kopiafgift stiller dette til rådighed for foreningens medlemmer. Kopiafgift (pt. 20 kr./volume) skal dække omkostninger og distribution samt udgøre grundlag for biblioteks-udbygning.

Public domain programmer er progr., der som navnet siger, ikke er omfattet af copyright og derfor kan distribueres frit. Det omfatter bl.a. programmeringssprog, tekstbehandling, regneark, database-programmer - endv. mange spil og værktøjer for blot at nævne et udsnit.

Kontingentet er 225 kr. årligt og gælder 1 år fra indmeldelsesdatoen.

Indmeldelse kan ske ved indbetaling af kontingentet (225 kr.) på girokonto:

5 68 6512 MUG Denmark, Esbern Snaresgade 6, 1725 København V.

Yderligere oplysning kan fås hos formanden eller kassereren på telf.:

4230 6913 & 3123 9236 samt BBS 3160 5319

41330
Frank Damgaard
Kastebjergvej 26A
2750 Ballerup

Eftersendes ikke ved
vedvarende adresseforandring
men tilbagesendes med oplysning
om den nye adresse

M.U.G. - / Lars Gråbæk - Esbern Snares Gade 6 - 1725 København V.

MicroComputer User Group

Indkaldelse til Generalforsamling

Der indkaldes til ordinær Generalforsamling i M.U.G
onsdag 27 okt 1993 kl. 19.00 i Vesterbros Kulturhus,
Lyrskovsgade 4, København V. (Lokale 2+3)

Dagsorden:

1. Valg af dirigent og referent.
2. Formandens beretning.
3. Kassererens beretning.
4. Bibliotekarernes beretning.
5. Indkomne forslag.
6. Valg af formand.
7. Valg af bestyrelsesmedlemmer.
8. Valg af 2 revisorer.
9. Eventuelt.

I dagens anledning trækkes lod om div. objekter doneret af forskellige sponsorer. Hvert medl. får et lod ved ankomst.

Kom og vær med!